

# SONG INFORMATION AND LYRICS

PRI - 3RD MILESTONE

Afonso Caiado, up201806789

Diogo Martins, up201806280


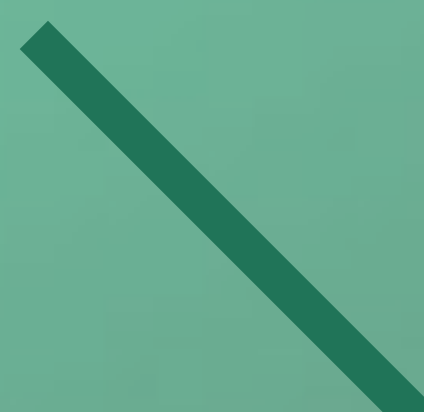
José Mações, up201806622



# Improvements

After the first two milestones, we proceeded to analyze our project and try to understand where we could upgrade some of its aspects.

This way, we changed some parts relative to the first and second milestones with the goal to improve our information system.

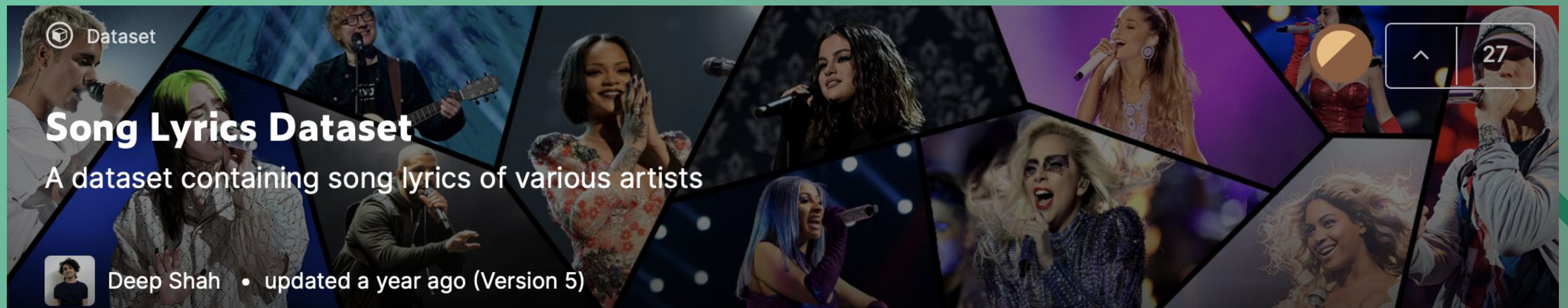


# Dataset

We started with a dataset comprised of multiple artists' songs with the respective lyrics and decided we wanted to have **each songs' genre**.

We joined other datasets with songs and their genre, but the overlap between our base dataset and the new datasets was **minimal**, which meant **we had not much data**.

We felt like losing that much data just to have a new genre column wasn't the best way to go about our proceedings. We felt that **having more information about each song would greatly improve our search system**.

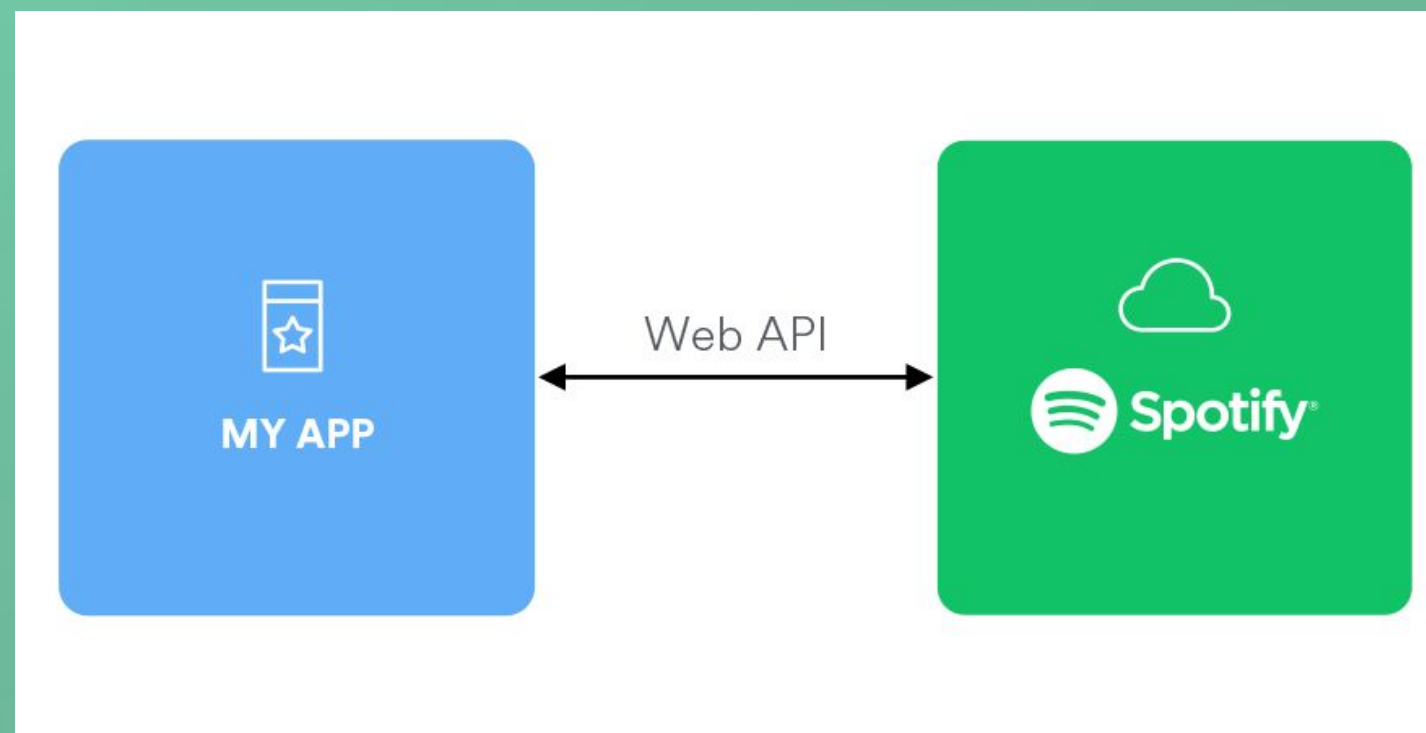











# Dataset

We kept the original dataset with songs and lyrics, and compiled a Spotify playlist with every song from each artist in the dataset. We then used **Spotify's API** to collect information about each song, **not only the genre** but also interesting metrics that could possibly be used in the future, more concretely **Duration, Popularity, Beats Per Minute, Danceability, Energy** and **Valence**.

This improvement lead to an increase in the dimension of our final dataset, which has 1323 songs, **more than double than the initial dataset**.



#	TITLE	ALBUM	DATE ADDED	
1	 <b>Lust For Life</b> E Drake	So Far Gone	10 days ago	2:56
2	 <b>Houstatlantavegas</b> E Drake	So Far Gone	10 days ago	4:51
3	 <b>Successful (feat. Trey Songz &amp; Lil Wayne)</b> E Drake, Trey Songz, Lil Wayne	So Far Gone	10 days ago	6:15
4	 <b>Let's Call It Off (feat. Peter Bjorn and John)</b> E Drake, Peter Bjorn and John	So Far Gone	10 days ago	3:54
5	 <b>November 18th</b> E Drake	So Far Gone	10 days ago	3:08
6	 <b>Ignant Shit (feat. Lil Wayne)</b> E Drake, Lil Wayne	So Far Gone	10 days ago	5:02
7	 <b>A Night Off (feat. Lloyd)</b> E Drake, Lloyd	So Far Gone	10 days ago	3:14

# Collection Composition

## Document

Only one, named Song, that has **"Title"**, **"Artist"**, **"Album"**, **"Lyrics"**, **"Top Genre"**, **"Date"** fields, the **"id"** one added to aid the evaluation process, and finally the ones recently added: **"Duration"**, **"Popularity"**, **"Beats Per Minute"**, **"Danceability"**, **"Energy"** and **"Valence"**.

## Indexing

Columns for indexing: **"Title"**, **"Artist"**, **"Album"**, **"Lyrics"**, **"Top Genre"**, **"Date"**, **"Duration"**, **"Popularity"**, **"Beats Per Minute"**, **"Danceability"**, **"Energy"** and **"Valence"**.

The **"Lyrics"** column contains large strings that allow text search, the indexing of it improved the performance of our system regarding the search.

# Schema

When we first created our schema, **it was very simple**, with basic filters and tokenizers.

This became clear when the evaluation was done, as there was no difference between searching with and without schema. We also realised we have a big text field, the *Lyrics* field, over which most of the search will be, **has room to be better analyzed**.

We explored Solr's documentation to find any filter we thought made sense in our case, and ended up adding a few to our schema.



# Schema

- **Stopword Filter:** we used Solr's **ManagedStopFilter** to remove the words "song" and "songs" from the query. This will allow the system to ignore those words and just search for the other words.
- **KStem Filter:** The KStem filter turns, for example, the words "jumping" and "jumped" into "jump". It generalizes some words. **KStemFilterFactory** from *Solr* was used.
- **EdgeNGram Filter:** This filter was already in use in the *Title* field and we decided to expand it to the *Lyrics*, to generate smaller tokens of each word, broadening the search possibilities. We used *Solr's* **EdgeNGramFilterFactory**, with a minimum gram size of 3 characters and a max gram size of 7.

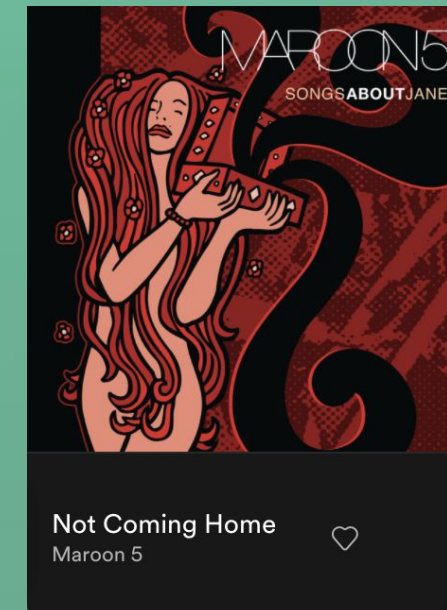


# Search Examples

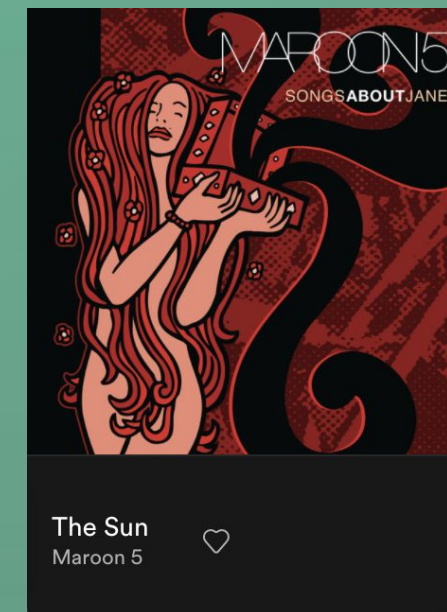
```
{
  "responseHeader": {
    "status": 0,
    "QTime": 26,
    "params": {
      "q": "sad songs",
      "defType": "edismax",
      "ps": "3",
      "indent": "true",
      "qf": "Lyric Title Album Top_Genre",
      "pf": "Lyric",
      "q.op": "OR",
      "lowercaseOperators": "true",
      "_": "1642694562567"
    }
  }
}
```

```
"response": {
  "numFound": 73,
  "start": 0,
  "numFoundExact": true,
  "docs": [
    {
      "Artist": "Maroon 5",
      "Title": "Not Coming Home",
      "Album": "Songs About Jane",
      "Date": "2002-06-25T00:00:00Z",
      "Lyric": "when you refuse me you confuse me what makes you think",
      "Duration": [261.226],
      "Popularity": [44],
      "BPM": [120.04],
      "Danceability": [0.667],
      "Energy": [0.921],
      "Valence": [0.378],
      "id": "708",
      "Top_Genre": ["pop"],
      "_version_": 1722490033532829696
    },
    {
      "Artist": "Maroon 5",
      "Title": "The Sun",
      "Album": "Songs About Jane",
      "Date": "2002-06-25T00:00:00Z",
      "Lyric": "after school walking home fresh dirt under my fingernails",
      "Duration": [251.693],
      "Popularity": [48],
      "BPM": [79.989],
      "Danceability": [0.532],
      "Energy": [0.73],
      "Valence": [0.558],

```



Not Coming Home  
Maroon 5



The Sun  
Maroon 5

These songs have  
in fact  
melancholic lyrics

Does it make you sad to  
find yourself alone?

Does it make you mad to  
find that I have grown?

And mama, I've been  
cryin'

'Cause things ain't how  
they used to be

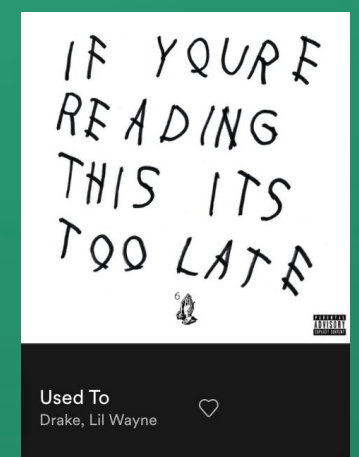
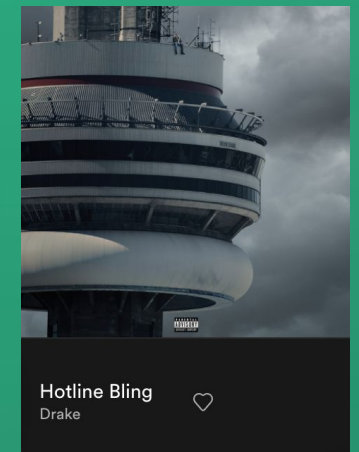


# Search Examples

As we can see, the lyrics match

```
{
  "responseHeader": {
    "status": 0,
    "QTime": 21,
    "params": {
      "q": "you use to call me on my cell phone",
      "defType": "edismax",
      "ps": "3",
      "indent": "true",
      "qf": "Lyric Title Album Top_Genre",
      "pf": "Lyric",
      "q.op": "OR",
      "lowercaseOperators": "true",
      "_": "1642694562567"
    }
  },
```

```
"response": { "numFound": 1308, "start": 0, "numFoundExact": true, "docs": [
  {
    "Artist": "Drake",
    "Title": "Hotline Bling",
    "Album": "Views",
    "Date": "2015-07-25T00:00:00Z",
    "Lyric": "you used to call me on my you used to you used to yeah you used to call me on my cell phone",
    "Duration": [267.06600000000003],
    "Popularity": [80],
    "BPM": [134.966],
    "Danceability": [0.891],
    "Energy": [0.628],
    "Valence": [0.552],
    "id": "55",
    "Top_Genre": ["canadian hip hop"],
    "_version_": 1722490030129152000
  },
  {
    "Artist": "Drake",
    "Title": "Used To",
    "Album": "If You're Reading This It's Too Late",
    "Date": "2015-02-13T00:00:00Z",
```



# Evaluation

For the evaluation of our improvements, we **repeated the queries we did previously** (**Sad Songs**, **Songs to Dance in the Shower**, **Nostalgic Songs**) in order to obtain a true comparison of our search system before and after improvements. This meant **we repeated the three search systems** we had initially for each query.

Furthermore, calculated the same metrics, Precision at 10 (P) and Average Precision (AP)

## Sad Songs:

System 1 (No Schema)  
 $P = 5/10 = 0.5$   
 $AP = 0.564727$

System 2 (Schema)  
 $P = 5/10 = 0.5$   
 $AP = 0.564727$

System 3 (Schema with Boost)  
 $P = 6/10 = 0.6$   
 $AP = 0.788183$

System 1 (No Schema)  
*Results* : 1010101101  
 $P = 6/10 = 0.6$   
 $AP = 0.613183$

System 2 (Schema)  
*Results* : 1010101010  
 $P = 5/10 = 0.5$   
 $AP = 0.599295$

System 3 (Schema with Boost)  
*Results* : 0110011111  
 $P = 7/10 = 0.7$   
 $AP = 0.492196$

Previously

Now

# Evaluation

## Songs to Dance in the Shower:

System 1 (No Schema)  
 $P = 5/10 = 0.5$   
 $AP = 0.372928$

System 2 (Schema)  
 $P = 5/10 = 0.5$   
 $AP = 0.372928$

System 3 (Schema with Boost)  
 $P = 7/10 = 0.7$   
 $AP = 0.727381$

Previously

System 1 (No Schema)  
*Results* : 1111001001  
 $P = 6/10 = 0.6$   
 $AP = 0.817945$

System 2 (Schema)  
*Results* : 1111001001  
 $P = 6/10 = 0.6$   
 $AP = 0.817945$

System 3 (Schema with Boost)  
*Results* : 1011111100  
 $P = 7/10 = 0.7$   
 $AP = 0.784436$

Now

## Nostalgic Songs:

System 1 (No Schema)  
 $P = 8/10 = 0.8$   
 $AP = 0.973765$

System 2 (Schema)  
 $P = 8/10 = 0.8$   
 $AP = 0.973765$

System 3 (Schema with Boost)  
 $P = 7/10 = 0.7$   
 $AP = 0.488183$

Previously

System 1 (No Schema)  
*Results* : 1101010101  
 $P = 6/10 = 0.6$   
 $AP = 0.715035$

System 2 (Schema)  
*Results* : 1101111010  
 $P = 7/10 = 0.7$   
 $AP = 0.826102$

System 3 (Schema with Boost)  
*Results* : 0111110111  
 $P = 6/10 = 0.8$   
 $AP = 0.643563$

Now



# Conclusion

In this last phase of our project, we improved some aspects such as the dataset and the schema, with the intention to improve our search system.

The process of *Evaluation* allowed us to compare the quality of our work with the modified aspects to the work we had at the end of the second milestone. By analyzing the results, we could see that the improvements did in fact contribute to upgrade the search system.

Nevertheless, an aspect of our work that we wish it would be better was the dataset, since the number of songs is not enormous, even though it's reasonable and we could get even more songs at the end.

To conclude, this project allowed to improve and establish good knowledge basis regarding search systems and information retrieval. Looking back at our work, we feel that we fulfilled the goals established in the subject of PRI.