

Lyrical Analysis Project

The Effect of Lyrical Construction on Song Popularity

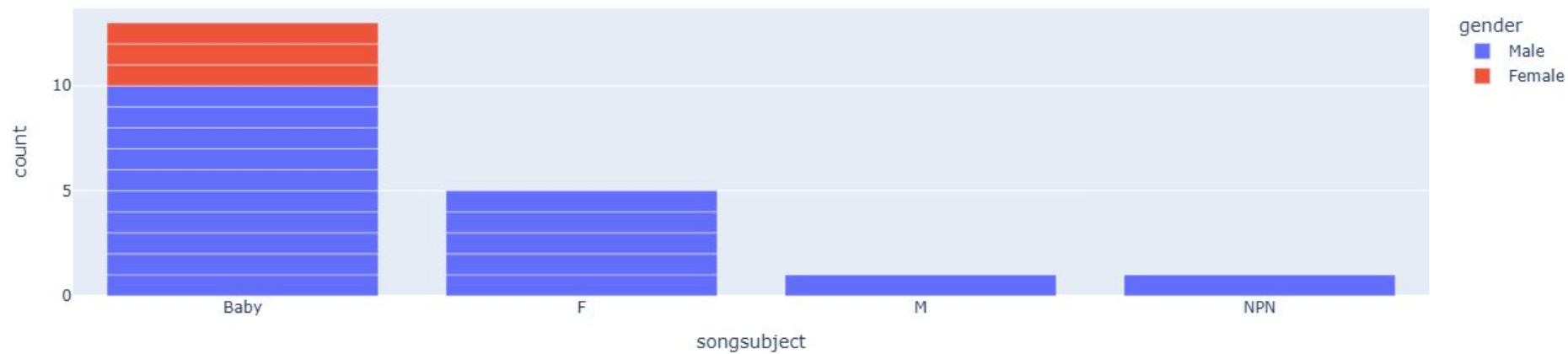
Understanding Our Data

- Pulling from Billboard Top 100 Over last ~70 years

GROUPS

- Songs in Number 1 Slot
- Songs in Number 50 Slot
- Songs in Number 100 Slot

Exploring Data Patterns



Exploring Data Patterns

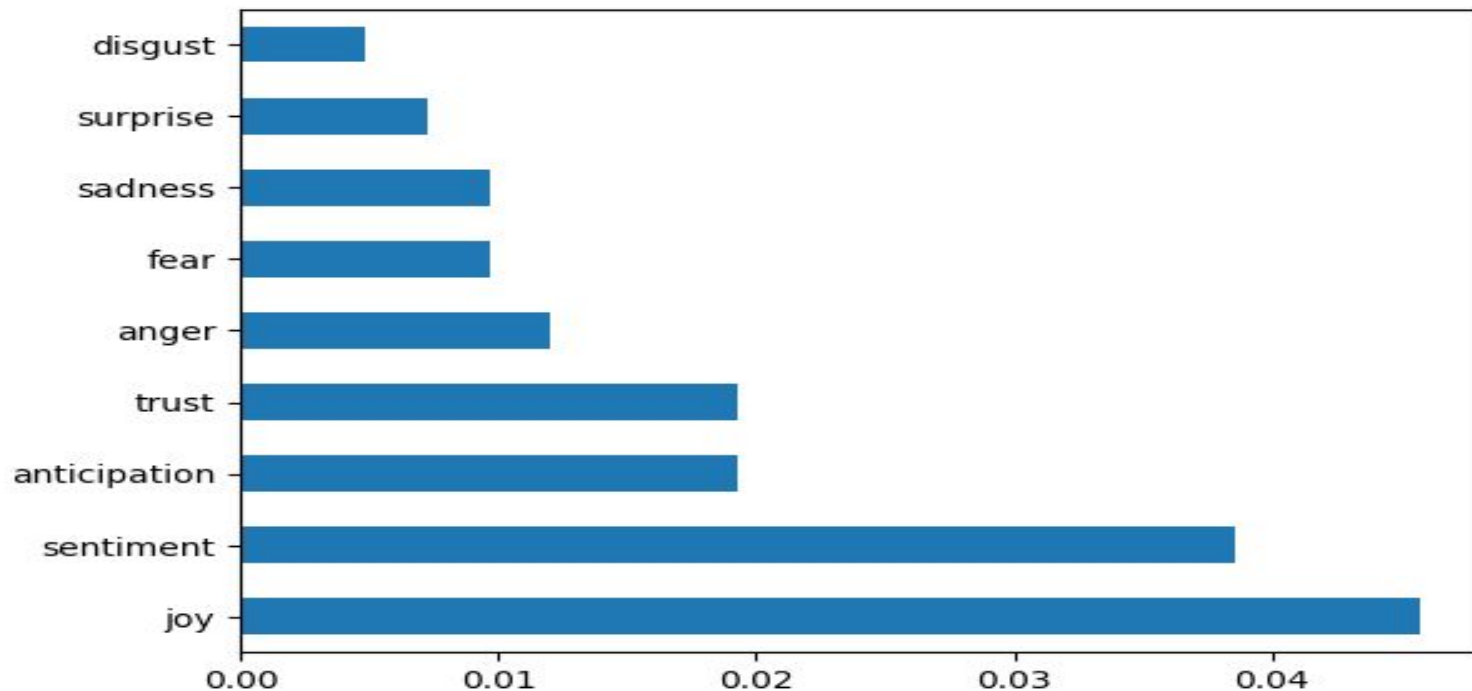
HighstRank: the highest ranking a given song ever reaches

Year: year of song release

Here, we see that on average, songs are ranked around 41 at the highest. The average year in the dataset is 1994. Meaning a lot of our content may hover around 1990-early 2000s

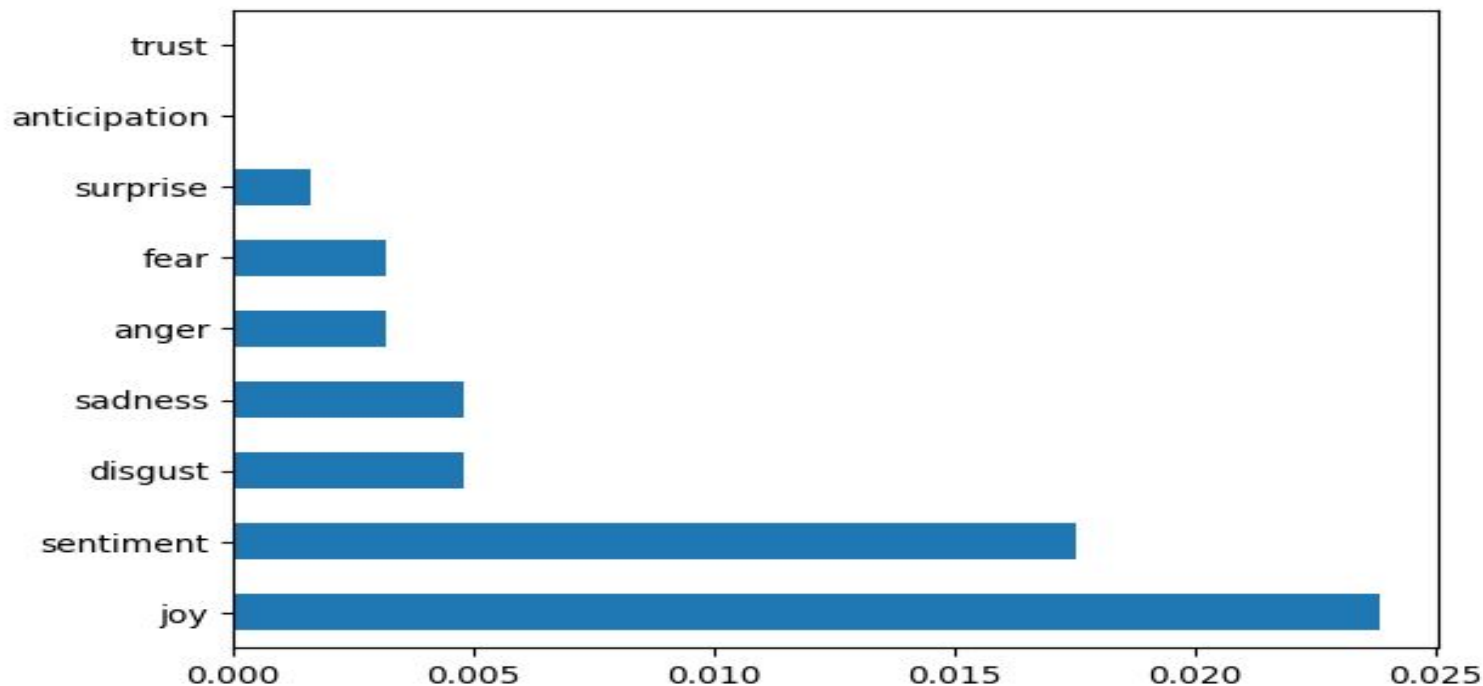
	highstRank	year
mean	41.484750	1994.991863
std	30.801748	18.637030
min	1.000000	1958.000000
25%	12.000000	1978.000000
50%	39.000000	2000.000000
75%	67.000000	2012.000000
max	100.000000	2019.000000

What Lyrical Sentiments Do Number 1 Songs Encapsulate?



Let Me Love You by Mario

What Lyrical Sentiments Do Number 1 Songs Encapsulate?



Only Girl in the World by Rihanna

Text Analysis

As we transform each songs lyrics into bite size pieces, new analysis becomes possible!

- Frequency Stats
- GPT Models
- Text Generation

Text Generation

1 STOP

2 A MOHAIR SUIT

3 'YEAH I DID.'

4 'I LOVE THE WAY LOVE'S SUPPOSED TO BE

5 SUPPOSED TO BE IN LOVE IN EAST L.A.'

6 MAMAS COME ON'

7 BY CARLOS SANTANA

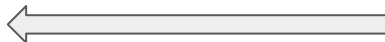
8 AND I'M THINKIN' 'BOUT STICKIN' TO THE BEANPOLE DAMES IN THE WORLD LIKE I'M THE ONLY
ONE WHO KNOWS YOUR HEART 'ONLY GIRL IN THE WORLD

This text is generated from models called bi & trigram models.

The text is generated from the highest order model and strings the words together to resemble the original content

It doesn't hold the same contextual properties as a GPT, but shows us general trends across the data and supports creativity!

Here's a section that blends multiple artists into one sentence



GPTS; Large Language Models

We can utilize pre-trained GPTs to answer lyrical questions and finish sentences with context

```
model.eval()
```

```
prompt = "love is"
```

```
inputs = tokenizer(prompt, return_tensors='pt')
```

```
outputs = model.generate(inputs['input_ids'], max_length=50, do_sample=True, top_k=40)
```

```
print(tokenizer.decode(outputs[0]))
```

The highlights section above is the prompt. This will contextualize the GPT