Analysis of Kanye West’s discography

Introduction

Kanye West is one of the most influential artists of the 21st century. He is known for his often shocking antics, clever and funny lyrics and stellar music production. As he, as an artist, is close to my heart, I decided to analyse his lyrics from all his solo studio albums and two of his collaboration projects – Kids See Ghosts and Watch The Throne. The aim of this work is to get more introspective into his lyricism and get a deeper insight into how a text similarity model will connect his works.

Dataset

Kanye’s lyrics are freely available on the internet. In order for them to be viable as a dataset for a text analytics task, I used genius.com API to extract all songs from his studio albums. The final product is a JSON file containing name of the song, name of the album the song was is featured on and the lyrics. This format gave me freedom to analyse individual songs and whole albums alike. As the dataset is not really suitable for a classification task, after a consultation with my teacher, I was allowed to omit it.

Many of Kanye’s songs have a substantial number of features from other artists and the verses they contribute with are not Kanye’s, however for the sake of simplicity, I decided to include every line of lyrics that is listed on genius.

In order to be able to get valid result, the dataset required some level of preprocessing right away. First, when downloading the lyrics, some artifacts from the website needed to be removed. This was done using regex and is further described in the attached jupyter notebook *dataset_download.ipynb.* Furthermore, only songs with lyrics were considered. I also decided to omit skits. While skits have a unique personality and add to the narrative of the album, they do not act as a standalone piece of art and don’t show Kanye’s lyricism quite as well as his “normal” songs do.

Another challenge I faced was the presence of adlibs. Adlib is a part of the song, where a rapper or singer improvises. It mostly consists of semi-random sounds i.e. “yeah”, “uh”, “yo”, “bam”. They do not have any meaning besides hyping up the listener or filling up space, so I decided to remove them as they could negatively affect the results. List of removed adlibs can be found in the code.

Clustering

code available in analysis.ipynb in section Clustering

Goal of this task was to identify clusters of songs that belong together. For this I used KMeans clustering with TF-IDF score as a metric. This causes that songs that are lyrically close will be clustered together. This approach however does not guarantee a semantic similarity, as the word meaning is not considered.

Preprocessing for this task included lemmatization, removing of stopwords (adlibs included), expanding contractions and tokenization.

I decided to go with 5 clusters as I was not able to get KElbow score working with textual data. The yielded results are 5 clusters, each with different most common keywords. The cluster’s respective sizes are 38, 79, 45, 9, 5. After extracting the most important keywords by TF-IDF score from each cluster, we can guess the common theme of the songs.

	Most important keywords	**Assumed theme**
Cluster 0	hallelujah,need,light,like,donda,pray,new,want,see,thank,every,let,keep,life,know,go,make,god,jesus,lord	Spiritual, mentions of God and his mother Donda
Cluster 1	shit,think,never,man,would,back,could,take,make,want,tell,let,nigga,cannot,girl,see,say,know,go,like	Angry songs, raw expression of feelings
Cluster 2	would,fuckin,one,say,go,right,bad,make,want,black,gon,love,need,hand,bitch,like,know,shit,fuck,nigga	Songs encompassing stereotypical rap music themes
Cluster 3	thing,breathe,finish,nothing,alive,baby,turn,devil,go,lord,gon,love,lie,sun,na,know,alright,god,okay,wave	Songs centered around love and struggles of life
Cluster 4	roll,fade,na,yes,many,never,feel,could,heaven,take,inside,scar,soon,leave,know,deep,far,amaze,moon,lift	Upbeat optimistic songs, possibly spiritual

Figure 1: Table of clusters, their most important keywords and assumptions of themes

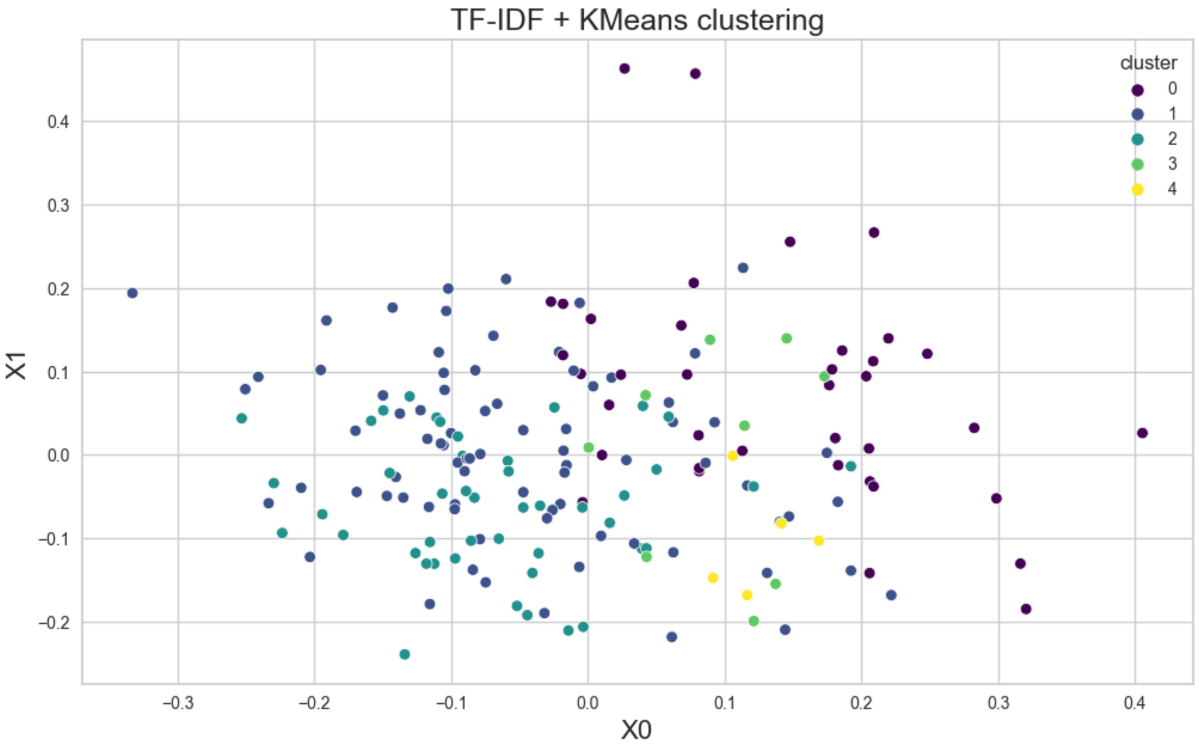


Figure 2: KMeans clusters plotted with reduced dimensionality using PCA

In *Figure 1* we can't really distinguish any reasonable patterns, as the dimensionality reduction from PCA is too great. Code for generating *Figure 1* was sourced from [1] along with the PCA and is not my original work.

The label assignment itself seemed promising and - based on manual review of the songs - decently accurate – this is to be considered a sidenote or a fun fact as it is not backed by any proof other than my knowledge of the songs.

Topic Modelling

code available in analysis.ipynb in section Topic modelling

This task aims to provide a deeper insight into important topics in Kanye's lyrics. For this task, all songs across Kanye's discography were used in order to distinguish general topics of his songs. After creating a corpus and dictionary, I passed these into LDA model.

Visualization of the model proved challenging, but in the end yielded a nice interactive table with the help of pyLDAvis library. After reading through the documentation, I decided to set the relevance metric λ = 0.6, as that yields the best results (I decided to trust the authors of the library in this case). [2]

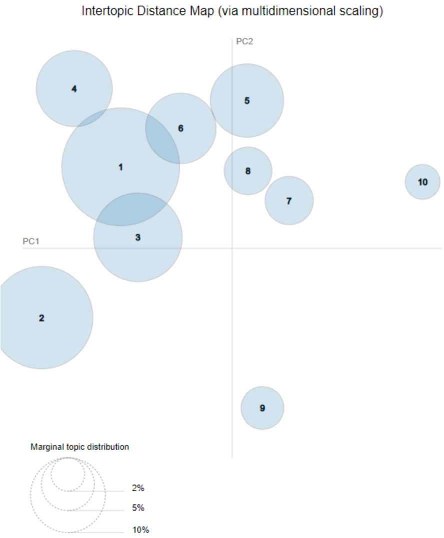


Figure 3: Visualization of 10 important topics across Kanye's lyrics

I will further analyse only a subset of topics that I consider interesting.

Topic 1

This topic is the most broad, encompassing 24.8 % of all tokens. While the topic is huge, it is hard to deduce any specifics. The presence of words like "nigga", "god", "love", "baby", I can assume that this topic to be a generalization of Kanye's lyrics and is a "big tent" topic for his songs, as those terms are among the more frequent terms lyric-wise. From the graph of intertopic distances we can see, that **topics 3, 4, 5, 6, 7, 8** are in proximity of this topic, which would support my assumption that this is a broadly generalized topic.

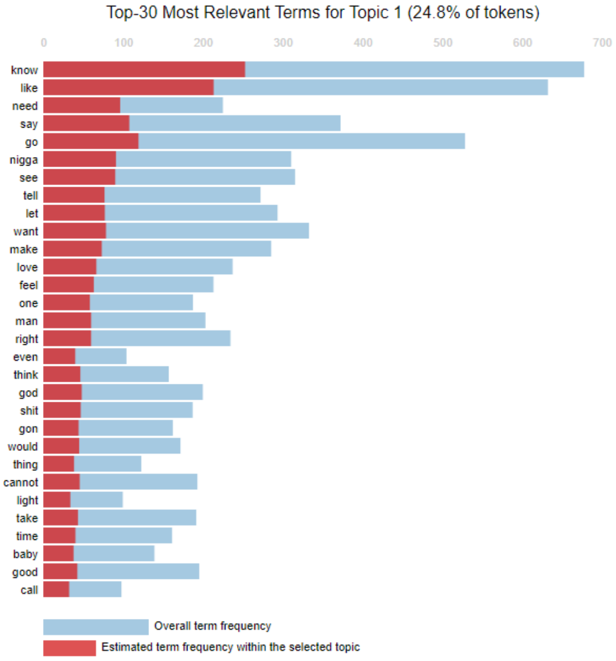
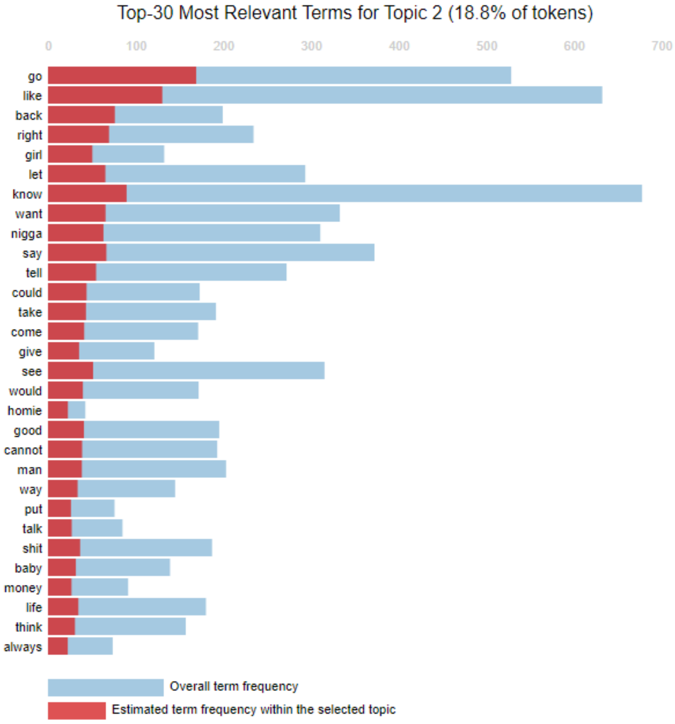


Figure 4: Most relevant terms for Topic 1

Topic 2

Second most important topic is Topic 2. On the intertopic distance plot it lies in proximity to 2, however lies a bit further than the "cluster" of **topics 1, 3, 4, 5, 6, 7, 8**. This topic is easier to interpret. From nouns like "nigga", "homie", "money", "girl" and verbs like "go", "let", "want" I can assume that this topic is talking about Kanye's rapper lifestyle, which can often be lavish, extravagant and shocking for normal people.



1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et. al (2012)
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2014)

Figure 5: Most relevant terms for Topic 2

Topic 9

This topic is interesting in a different way than the previous two in a way, that it is a complete outlier. Upon further inspection, we can see, that this topic is about Kanye’s spiritual side with words such as “love”, “lord”, “jesus”, “god” being among the most relevant.

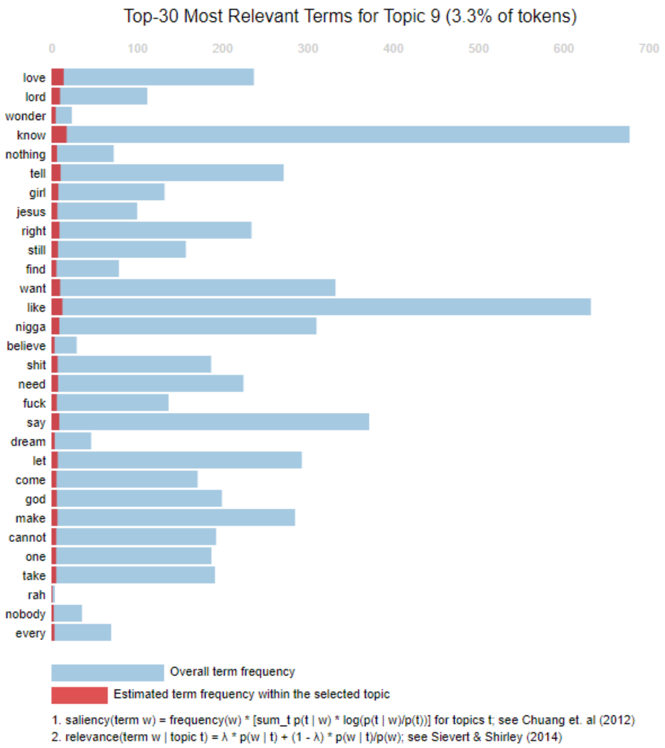


Figure 6: Most relevant terms for Topic 9

Other topics can be analogically analysed using the interactive tool in attached jupyter notebook in the section Topic modelling.

While topic modelling yielded some interesting results, interpretation is much harder than in other tasks. Easiest to interpret were the outliers with clear and distinct terms - such as the spiritual one.

Mutual Similarity

code available in analysis.ipynb in section Mutual Similarity

Many songs share a theme, message or are otherwise interconnected by their meaning. Analysis of mutual similarity can be a useful tool to uncovering these connections. While clustering provided insight into similarity based purely on words used, similarity analysis based on word embeddings can provide interesting insight into similarity based on meaning.

For this task, I decided to compare songs on two individual albums (or songs from one album if the method is provided with the same name twice) and calculating their cosine similarities. The model used was “all-mpnet-base-v2” as that was the model that was reported in documentation of sentence-transformer library as overall the best performing. Experiments with other models could be done.

For this task, unpreprocessed texts were used. The only change I made was removing adlibs as I believed that it could negatively skew the results based on words that bear no significant meaning.

As I considered 12 albums from Kanye's discography, generating all of them would be tedious, slow and hard to interpret (the total amount of results and plots would be 144), I decided to only select Kanye's most influential albums or albums I personally like or find interesting in other ways.

I used heatmaps for visualisation of the mutual cosine score of songs as it provides comprehensive yet easy to understand interpretation. I also decided to track most and least similar songs from each album comparison.

My Beautiful Dark Twisted Fantasy & Graduation

Comparison of arguably the most important hip-hop albums of 21st century yielded interesting, yet unsurprising results.

Most similar songs: All of the Lights, Flashing Lights

Least similar songs: So Appalled, Good Night

All of the Lights and Flashing lights being the most semantically similar songs on these albums is not surprising as they both use "lights" in similar context and both portray Kanye's toxic relationship with fame.

So Appalled and Good Night can't be more distant - while one is about life being too short to worry about things, other heads-on battles critics and popular culture as a whole.

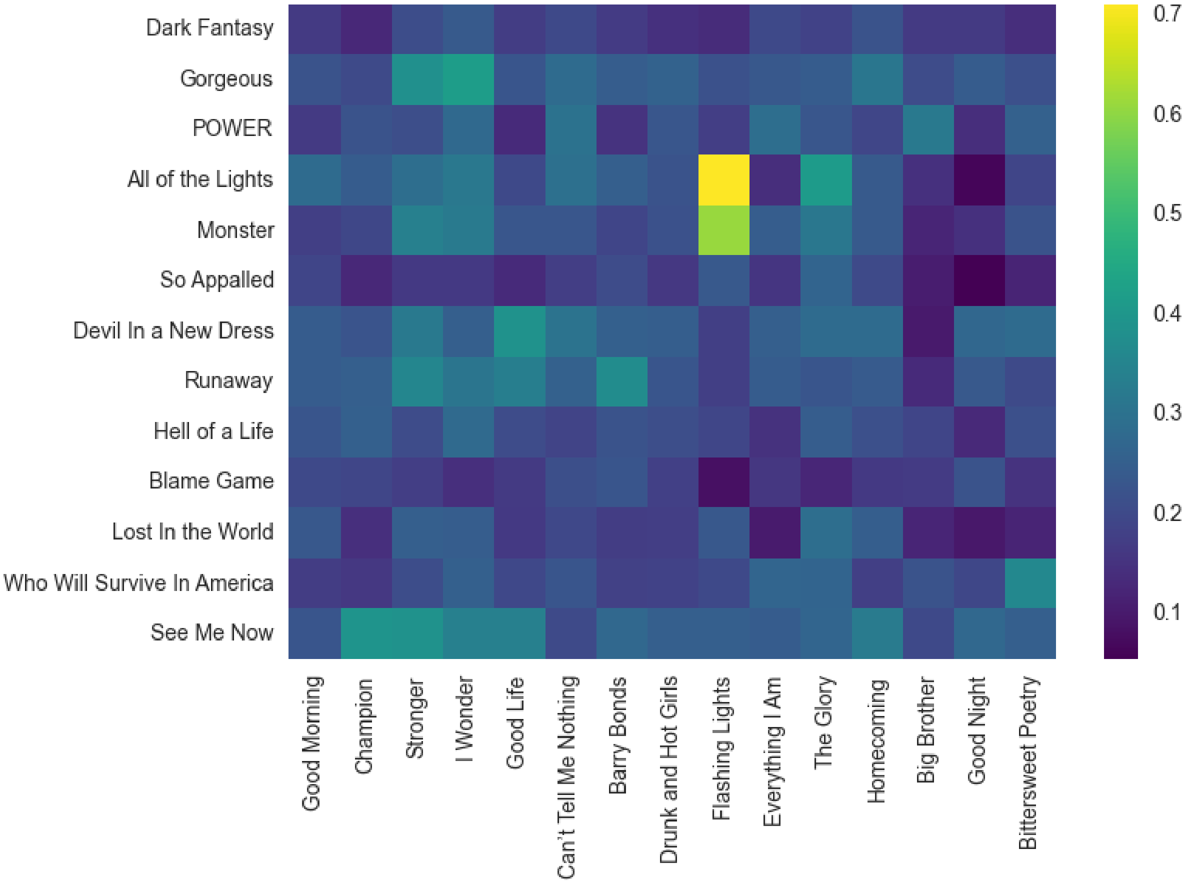


Figure 7: Heatmap of semantic similarities between MBDTF and Graduation

Kids See Ghosts & Ye

These albums both provide a deep dive into Kanye's mental and expose his vulnerable side. They delve deep into his mental issues, his family and personal struggles.

Most similar songs: Wouldn't Leave, Feel the Love

Least similar songs: Yikes, Reborn

Feel the Love and Wouldn't leave surprised me as the most similar duo. While both songs feature positive vibes and mention love, Feel the Love seems a bit too aggressive and braggadocious to be this similar.

Yikes and Reborn as the least similar make perfect sense as Yikes is an aggressive trap banger about Kanye's mental issues with some quirky lyrics, Reborn is a mellow feel-good song about moving forward from past struggles.

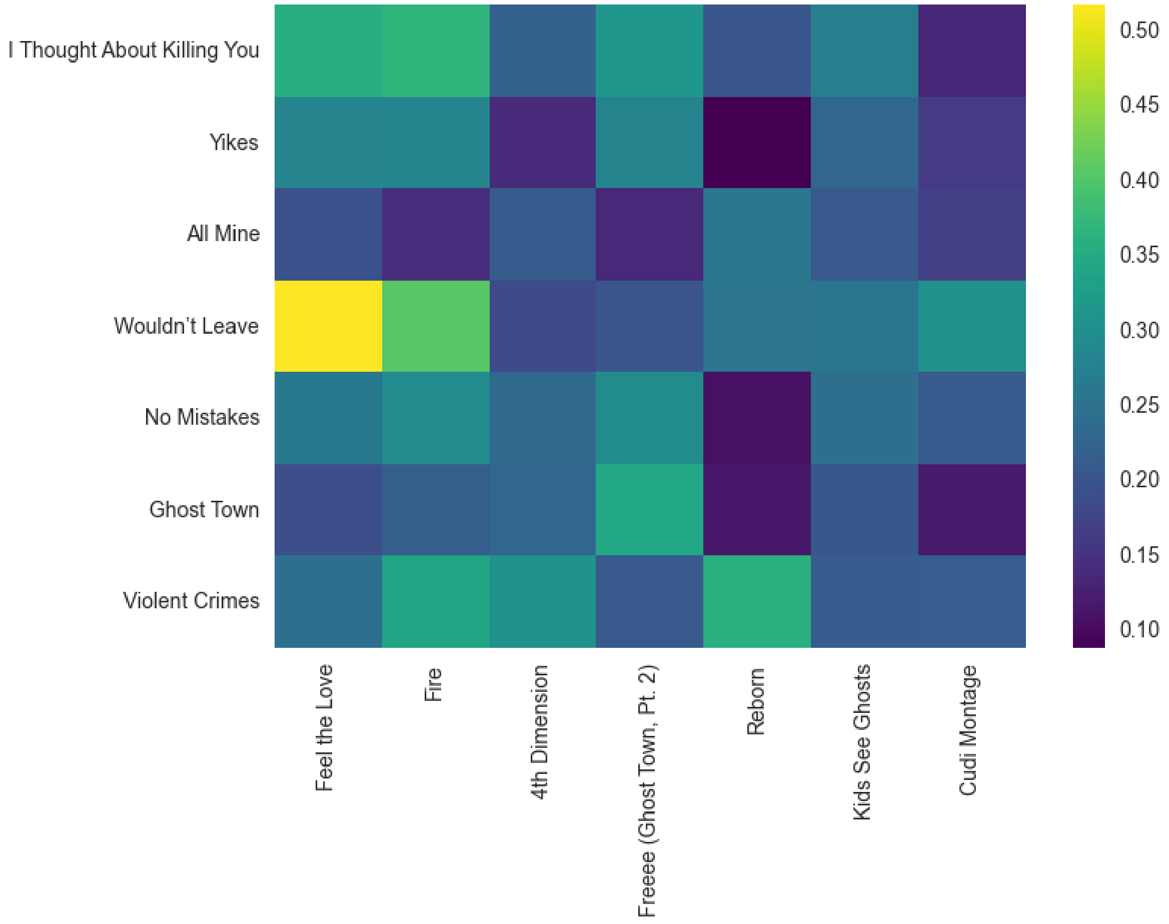


Figure 8: Heatmap of semantic similarities between Ye and Kids See Ghosts

The Life of Pablo & The Life of Pablo

As the last comparison, I decided to compare songs within The Life of Pablo, which is regarded as one of Kanye's best of 2010's.

Most similar songs: No More Parties in LA, Saint Pablo

Least similar songs: Famous, No More Parties in LA

No More Parties in LA and Saint Pablo both deal with celebrity lifestyle, although both a bit different. While the former is a critique of fake celebrity lifestyle, Saint Pablo is a deep personal story of Kanye's struggle with debt and downsides of being a celebrity.

Famous is a song that portrays how it is being famous without much of a critique that is present on No More Parties in LA.

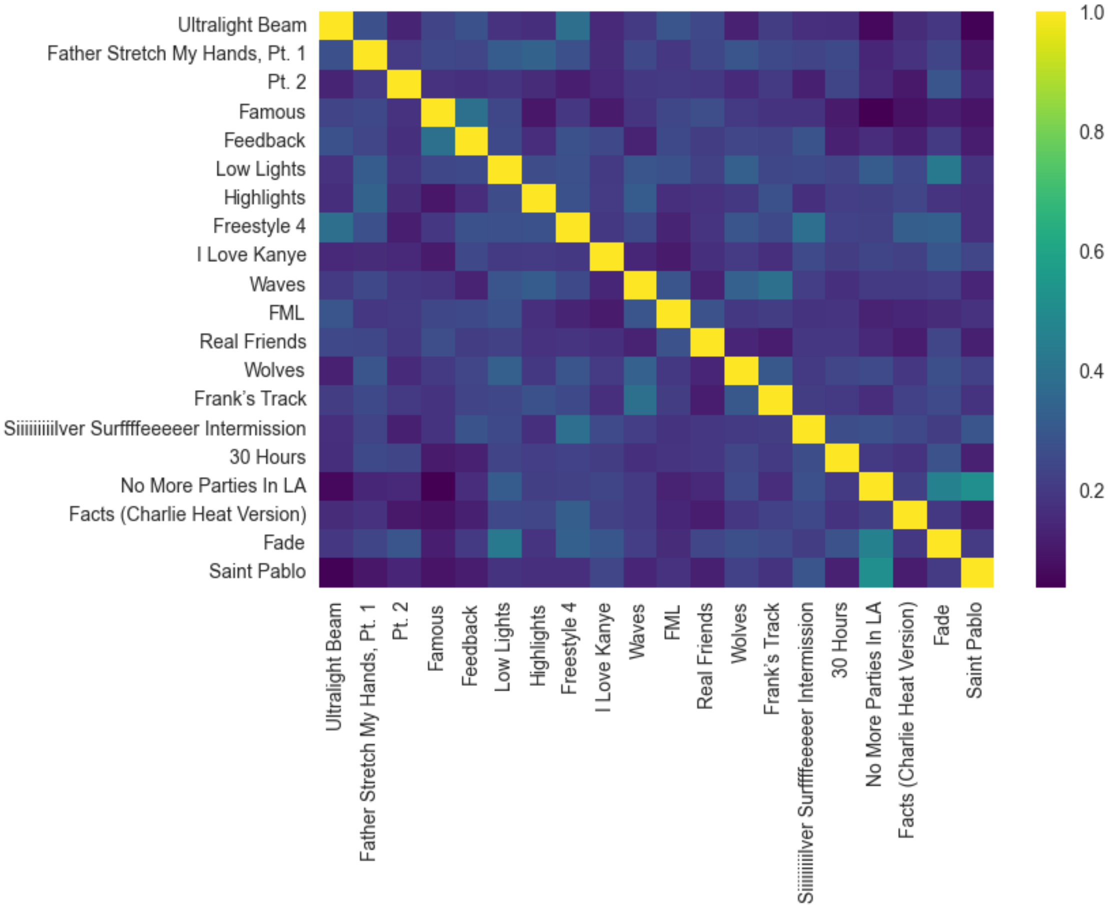


Figure 9: Heatmap of semantic similarities between songs on The Life of Pablo

Results from this analysis seemed most interesting to me, although I acknowledge some shortcomings. The texts are generally quite big and oftentimes hard to interpret. Kanye's lyrics are often cryptic and one song can touch on multiple themes at once. I am pleasantly surprised how well the model did.

Collocation analysis

code available in analysis.ipynb in section Mutual Similarity

Aim of collocation analysis is to identify common "neighbouring" words in documents. As per the assignment provided, I went with the most popular words based on TF-IDF score for each album. The words come lemmatized and cleared of stopwords including adlibs.

This task required tokenizing and vectorizing the album texts. I decided to treat all text from single album as one document. This could lead to negatively affecting the results by creating bigrams from ending of one and beginning of other documents. However, I decide to ignore this due to a condition of at least 5 occurrences of a bigram in order to be considered.

After extracting top 5 TF-IDF terms from each album, I could easily find their most common neighbours BigramCollocationFinder from nltk library.

My Beautiful Dark Twisted Fantasy

Below is provided a table containing the most important words based on TF-IDF score and the most common bigrams respectively.

Term	**Bigram**
ridiculous	(fuckin, ridiculous), (ridiculous, fuckin)
concert	(hand, concert), (concert, need)
toast	(let, toast)
slowly	(feel, slowly), (slowly, drift)
motherfuckin	(motherfuckin, monster), (know, motherfuckin)

Figure 10: Table containing most the frequent bigrams for most important terms in MBDTF

The most important terms are often influenced by oversaturated presence in one song i.e. "ridiculous" is ridiculously repeated in So Appalled and "motherfuckin" is oftentimes repeated on Monster.

These bigrams give a bit of insight into how these words are used from NLP perspective, however lemmatizing is not really useful for meaningful human understanding.

Ye

Below is provided a table containing the most important words based on TF-IDF score and the most common bigrams respectively.

Term	**Bigram**
someday	None
genie	(genie, bottle)
menacin	(could, menacing), (menacing, frightenin)
scare	(sometimes, scare)
mistake	(make, mistake), (mistake, girl)

Figure 11: Table containing the most frequent bigrams for most important terms in Ye

As for someone who listened to Ye extensively, it is clearly identifiable which songs these bigrams come from. I don't consider this output as interesting as previous tasks as it presents nothing new and is hard for humans to decipher.

The results from collocation analysis provided some limited insight into neighbouring words of the most important terms, but yielded no interesting nor surprising results.

Sources and Materials

[1] https://medium.com/mlearning-ai/text-clustering-with-tf-idf-in-python-c94cd26a31e7

[2] https://aclanthology.org/W14-3110.pdf