*Jing Cheng*

*Final Project in Computational Literacy*

# Themes in song lyrics across decades

## Introduction

When trying to choose a topic for this final project and aligning with my personal interest. I've always been fascinated by music literature, the way that written lyrics can convey emotions and represent current societal issues. So I wanted to choose a topic that focuses on the analysis of songs. Music is a multisensory experiences, but considering the difficulty of analyzing visual music video or auditory melody, I've decided to focus on song lyrics for my final project.

The goal is to identify the recurring themes and evaluate for differences or similarity in the number one hit song across decades. Though I am doing my bachelor's in computer science, I will admit that coding is not my strongest suit and therefore did not want to overcomplicate it, So my idea is to use topic modelling tools to identify the themes and see whether there is a correlation between the lyrics in different centuries.

- Research Questions

*"What recurring themes dominate hit songs from the 70s, 80s, and 90s and do they reflect societal values or emotions?"*

## Data Collection and preprocessing

To narrow the scope of analysis, I chose the hit songs from 1970s, 80s and 90s respectively. Due to problems such as lack of available archived data and charts, as I originally plan to extract data from Billboard. Therefore, I chose three widely acclaimed songs from each decades, according to Forbes best songs of the century article.

Forbes best songs in 70s/80s/90s have indicated the top song from each decade:

- "*Bohemian Rhapsody" by Queen (1975)*
- *"Purple Rain" by Prince (1984)*
- *"I Will Always Love You" by Whitney Houston (1992)*

The next step, I extracted the lyrics from Genius, an open-source platform, in txt. file. While the dataset is small, I think they are considered representative of their respective decades in terms of impact, as these are all hit songs that the general public can agree on.

Then proceeding to preprocessing of the raw data, I've taken measures to remove common filler and stopwords from the lyrics, lowercase the text and remove punctuations and special characters. This is done through the help of code provided from GeeksforGeeks. In the future, I would like to include more refined and custom stopword list to include words, such as "ohh, woo", that are commonly presented in lyrics.

In the code shown below, we first import stopword list from *nltk* library, then proceed to breakdown texts into individual words. Then filtering out the words from stopword list, we get a list of words in "x" format. Next step is to lowercase these words and then to remove punctuations. Last step is to join the word into one string (i.e., text format again).

```
I'll always love you"""

stop_words = set(stopwords.words('english'))

word_tokens = word_tokenize(example_sent)

filtered_sentence = [w for w in word_tokens if not w.lower() in stop_words]

filtered_sentence = []

for w in word_tokens:
    if w not in stop_words:
        filtered_sentence.append(w)

print(word_tokens)
print(filtered_sentence)

import string

filtered_sentence = [w.lower() for w in word_tokens if w.isalpha() and w.lower() not in stop_words]

print("Filtered Words:")
print(filtered_sentence)

cleaned_text = " ".join(filtered_sentence)

print("Cleaned Text:")
print(cleaned_text)
```

(The code showcases the example of when cleaning the lyrics for “I will always love you”. Part of the lyrics has been cropped out when screenshotting the VS code page)

The processed and cleaned text were copied from terminal to a folder all under txt.file. This allow us to proceed to the next step—identifying the themes in the processed texts respectively.

## Topic Modelling

Latent Dirichlet Allocation (LDA) tool is employed to analyze for the themes in each song’s lyrics. This approach allow us to identify clusters of words that frequently occur together, which are then interpreted as themes. This step is also done in python, through the use of *sklearn* library.

I’ve considered inputting the raw lyrics data into this model and tried to pre-run it, also the formatted data but without the removal of stopword. However it is difficult to analyze those text, it was hard for the model to give a precise prediction as well.

The cleaned text of each songs is first vectorized, then the lyrics were stored in document that are loaded later. The LDA model is implemented and trained the data. The topic was limited to one for each song and then printed and displayed in the terminal as follows:

*Analyzing themes for each song:*

*Song: Bohemian Rhapsody.txt*

*Topic 1:*

*me, go, you, let, no*

*Song: purple rain.txt*

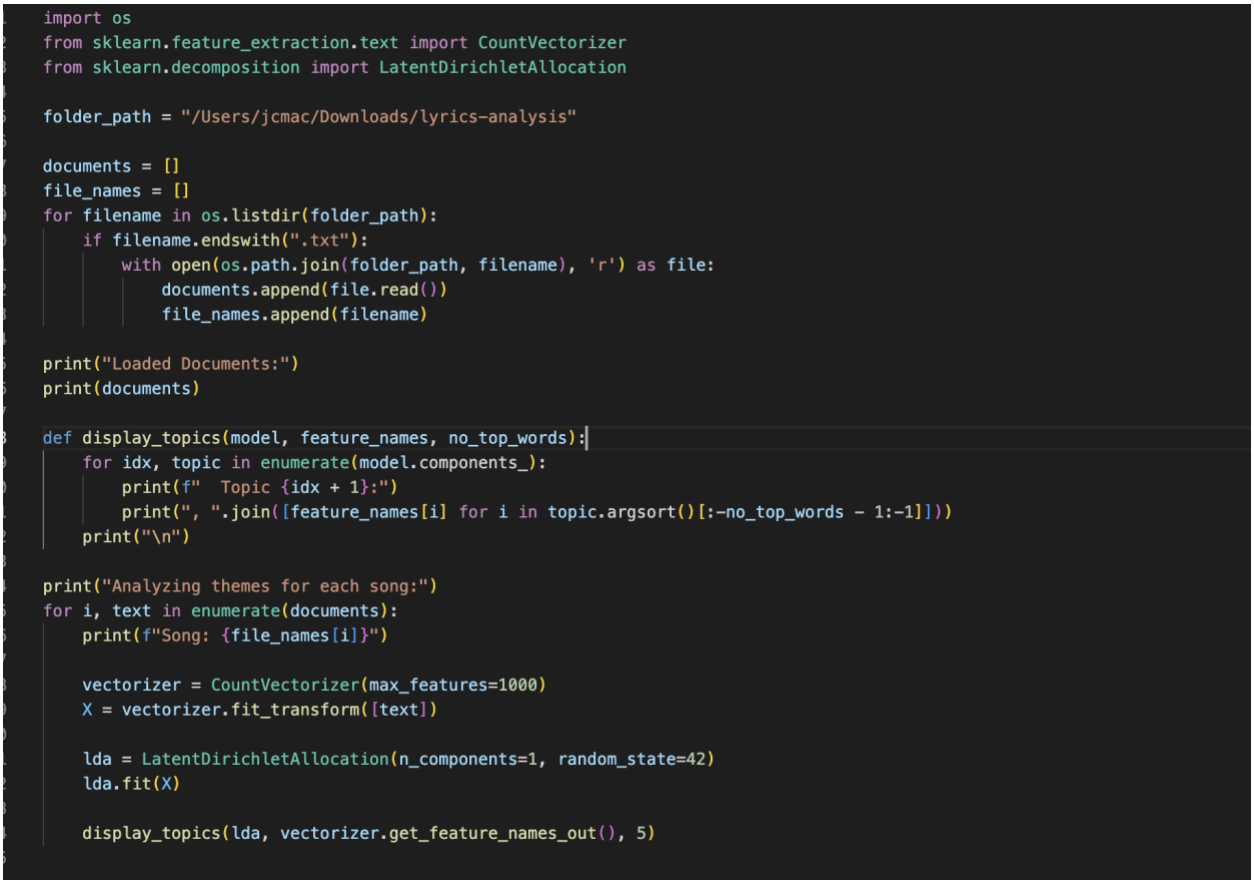
*Topic 1:*

*purple, rain, you, to, only*

*Song: I will always love you.txt*

*Topic 1:*

*you, love, always, will, and*



```
import os
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.decomposition import LatentDirichletAllocation

folder_path = "/Users/jcmac/Downloads/lyrics-analysis"

documents = []
file_names = []
for filename in os.listdir(folder_path):
    if filename.endswith(".txt"):
        with open(os.path.join(folder_path, filename), 'r') as file:
            documents.append(file.read())
            file_names.append(filename)

print("Loaded Documents:")
print(documents)

def display_topics(model, feature_names, no_top_words):
    for idx, topic in enumerate(model.components_):
        print(f"  Topic {idx + 1}:")
        print(", ".join([feature_names[i] for i in topic.argsort()[:-no_top_words - 1:-1]]))
    print("\n")

print("Analyzing themes for each song:")
for i, text in enumerate(documents):
    print(f"Song: {file_names[i]}")

    vectorizer = CountVectorizer(max_features=1000)
    X = vectorizer.fit_transform([text])

    lda = LatentDirichletAllocation(n_components=1, random_state=42)
    lda.fit(X)

    display_topics(lda, vectorizer.get_feature_names_out(), 5)
```

The theme directly produced from python in my opinion are rather vague and inaccurate, however it can be manually categorized into more comprehensible and understandable vocab as follows, though these are up to personal interpretation so the accuracy cannot be promised:

*Song: Bohemian Rhapsody*

*Theme: Self conflict*

*Song: Purple Rain*

*Theme: Longing*

*Song: I will always love you*

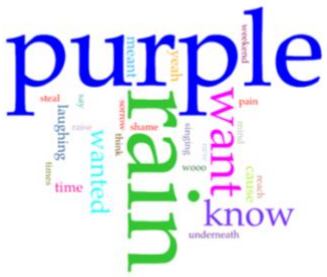
*Theme: Commitment and Love*

## Visualization

I tried to first visualize the themes in python, by using the code labelled as tm2.py in my folder, the wordcloud to generate image. However the results is unsatisfactory as it only display the identified theme and still has the same accuracy issue as mentioned above.

So I sought to another tool, the web-based Voyant tool, where the cleaned lyrics are input. And it automatically showcases the most common words and generate image for it. This tools offer a clearly visualization, however lacking in accuracy in depicting the actual theme due to its functionality nature.


*(Bohemian rhapsody)*




*(Purple rain)*


*(I will always love you)*

## Analysis and Improvements

In both visualization and themes identified in words, the research question posed at the start of this project—whether recurring themes in song lyrics reflect societal values and emotions across decades—was partially addressed but still can be further explored.

The analysis of the lyrics identified the recurring themes in chosen songs. Though the extracted themes broadly align with the overall emotional tone of the songs. For example, *"I Will Always Love You"* is a love ballad, and its dominant theme of "commitment and love". Similarly, *"Bohemian Rhapsody"* reflects existential struggle, which aligns with words like "me," "go," and "let." These themes hint at the societal values tied to the pursue of love and existential reflection.

However, the results also highlighted several limitations:

First being vagueness, the themes, while directionally correct, lack depth and are too generic. For instance, "*Purple Rain*" is dominated by the word “purple,” which adds little value to the semantics analysis. Though it overlaps at times as well in the example of *"I Will Always Love You"*. The small dataset also cannot make a definitive and logical and cohesive claim about the societal values across decades.

The nature of song lyrics structure causes certain words to disproportionately dominate the results. For example, words used in chores as it repeated significantly more compared to other parts of lyrics, such as bridges. This affects the overall judgement and can negatively impact the results generated from the model as well.

The limitation of a dataset, consisting of only three songs restricts the generalizability of insights and learnability of LDA model as well. However this was also due to the lack of open-source data as many songs/lyrics were copyrighted and it made downloading difficult. This does pose a problem with answering the initial research question.

## Improvements

A broader and more representative dataset, for example with multiple songs from each decade, could provide a better insight. As mentioned in data preprocessing part, the choice of stopwords list could maybe be improved for futures as well. It might be better to write a more detailed and custom code to achieve that. Given my personal limitation with programming, I also got a lot of help from stackoverflow and could implement that in future coding as well. The visualization could be done differently in the future, with the use of other tools in python. And with correctly identified themes, it could be possible to incorpate a semantics analysis as well, which I've decided not to implement in my project due to the lack of better understanding.