

---

# Swiftie Lyric Chatbot & Song Recommendation System

*Final Project*

---

Aaliya Merchant  
MSDS 453: Natural Language Processing  
December 8th, 2024

# Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Research Design and Modeling Method</b>	<b>2</b>
2.1	Data Wrangling and Preprocessing . . . . .	2
2.2	Feature Engineering . . . . .	3
2.3	Model Development and Methods . . . . .	3
2.4	Experiments and Hyperparameter Tuning . . . . .	4
2.5	Model Comparison and Justification . . . . .	4
<b>3</b>	<b>Exploratory Data Analysis (EDA)</b>	<b>4</b>
3.1	Album Statistics . . . . .	4
3.1.1	Word Count and Song Length . . . . .	5
3.1.2	Unique Words and Lexical Richness . . . . .	5
3.1.3	Average Song Length . . . . .	5
3.1.4	Lexical Richness . . . . .	5
3.1.5	Song Length and Lexical Richness Range . . . . .	5
3.1.6	Evolution Over Time . . . . .	5
3.2	Sentiment Analysis . . . . .	6
3.3	Top Songs by Sentiment and Lexical Richness . . . . .	6
3.4	Top 10 Most Frequent Entities . . . . .	6
3.5	Visualizations and Insights . . . . .	6
3.6	Exploratory Data Analysis Key Findings . . . . .	6
<b>4</b>	<b>Results</b>	<b>7</b>
4.1	Sentiment Distribution of Songs . . . . .	7
4.2	Song Recommendation Outputs . . . . .	7
4.2.1	Example 1: Input with Filters . . . . .	7
4.2.2	Example 2: Handling Typographical Errors . . . . .	8
4.2.3	Example 3: Overly Narrow Filters . . . . .	8
4.2.4	Example 4: Adjusted Filters for Broader Results . . . . .	8
4.2.5	Interpretations and Insights . . . . .	9
4.3	Visual Results: Sentiment Distribution . . . . .	9
<b>5</b>	<b>Conclusion</b>	<b>9</b>
5.1	Directions for Future Work . . . . .	9
<b>A</b>	<b>Appendix</b>	<b>10</b>
A.1	Tables . . . . .	10
A.2	Visuals . . . . .	12
A.3	Advanced Swiftie Lyric Bot Interaction . . . . .	15

# 1 Introduction

Artificial intelligence is increasingly leveraged to personalize user experiences, particularly in entertainment and creative industries. This project aims to develop an advanced chatbot that offers personalized song recommendations based on the lyrics of Taylor Swift. By combining Natural Language Processing (NLP) techniques with a rich lyrical dataset, the chatbot provides an engaging and customized interaction for fans and music enthusiasts.

The dataset for this project is sourced from the publicly available repository Taylor Swift Lyrics Dataset. It includes lyrics from Taylor Swift's complete discography of 11 original albums, excluding Taylor's Versions. This dataset is preprocessed to ensure semantic richness and thematic consistency, forming the foundation for building a chatbot capable of analyzing and responding based on lyrical content.

The objective of the project is to create a chatbot that delivers personalized song recommendations by analyzing user input and aligning it with the sentiment, themes, and lyrical structure of Taylor Swift's songs. Key functionalities of the chatbot include:

- **Lyric Similarity Analysis:** Using TF-IDF (Term Frequency-Inverse Document Frequency) to compare the similarity between user input and song lyrics.
- **Sentiment Analysis:** Identifying the emotional tone of user input to recommend songs matching the sentiment.
- **Entity Recognition:** Extracting thematic keywords and entities to enhance song recommendations.
- **Customizable Filters:** Allowing users to refine recommendations by album, sentiment, or theme.

The project employs tools such as SpaCy for preprocessing, Scikit-Learn for TF-IDF analysis, and sentiment analysis models to interpret user input. By integrating these techniques, the chatbot provides a tailored experience that not only reflects the artistry of Taylor Swift's lyrics but also caters to individual preferences. This innovative approach demonstrates how NLP techniques can be applied to creative datasets, offering insights into personalized content delivery and the potential of AI in enhancing fan engagement.

## 2 Research Design and Modeling Method

The goal of this project was to develop an advanced chatbot for personalized song recommendations based on Taylor Swift's lyrics. To achieve this, natural language processing (NLP) techniques were employed for tasks such as lyric similarity analysis, sentiment analysis, and thematic understanding. This section describes the modeling methods, experiments conducted, and the hyperparameter tuning process.

### 2.1 Data Wrangling and Preprocessing

The dataset used in this project consists of the lyrics from Taylor Swift's complete discography, excluding Taylor's versions. The dataset was preprocessed to remove noise and ensure that the lyrics are in a suitable format for analysis.

- **Data Cleaning:** The raw lyrics were cleaned to remove unnecessary parts of the dataset, such as album information and labels like "[Chorus]". This was achieved by applying the `clean_lyrics()` function, which ensured that the focus remained on the textual content of the lyrics.
- **Tokenization and Lemmatization:** To prepare the text for analysis, tokenization and lemmatization were performed using the `spaCy` library. Tokens were lemmatized to their root forms, ensuring that variations of a word (e.g., "run" vs. "running") were treated as the same entity. The `nlp()` function from `spaCy` was used for tokenization, and the lemma for each token was extracted to reduce dimensionality.
- **Stopword Removal:** Common but non-informative words (e.g., "the," "and," "in") were removed during preprocessing, using `spaCy`'s built-in stopwords list.

These preprocessing steps were critical for preparing the text data for subsequent feature extraction and model development.

## 2.2 Feature Engineering

Feature engineering involved converting the lyrics into numerical representations that the model could process efficiently. The two main techniques used for this purpose were TF-IDF (Term Frequency-Inverse Document Frequency) and sentiment analysis.

- **TF-IDF Vectorization:** TF-IDF was chosen to represent the lyrics, as it accounts for the importance of terms in relation to the entire dataset. The `TfidfVectorizer` from `sklearn` was used to perform this transformation. The vectorization parameters were set as follows:

```
vectorizer = TfidfVectorizer(max_df=0.85, min_df=0.01, ngram_range=(1, 2))
tfidf_matrix = vectorizer.fit_transform(cleaned_lyrics)
```

Here, `max_df` was set to 0.85 to exclude terms that appear too frequently across all lyrics, which are likely uninformative. The `min_df` was set to 0.01 to include terms that appear in at least 1% of the lyrics. The `ngram_range` was set to (1, 2) to consider both unigrams and bigrams, capturing both individual words and multi-word expressions that may hold significant meaning in the lyrics.

- **Sentiment Analysis:** To match user queries with songs of a similar emotional tone, sentiment analysis was performed using the VADER sentiment analysis tool, which is well-suited for short text inputs like song lyrics. The sentiment score for each song was calculated using the following code:

```
sentiment_score = sentiment_analyzer.polarity_scores(lyrics)
sentiment = 'positive' if sentiment_score['compound'] > 0 else 'negative'
```

The sentiment analysis classified the emotional tone of each lyric as either “positive” or “negative,” providing a key filter for song recommendations.

## 2.3 Model Development and Methods

The core functionality of the chatbot was designed to recommend songs based on user input, taking into account lyric similarity and sentiment alignment. The following techniques were applied to match lyrics to user queries.

- **Cosine Similarity for Lyric Matching:** To recommend songs based on the similarity of lyrics, cosine similarity was used to compare the TF-IDF vectors of the user’s query and the song lyrics. This was implemented with the `cosine_similarity()` function from `sklearn.metrics.pairwise`, as shown below:

```
cosine_similarities = cosine_similarity(user_query_vector, tfidf_matrix)
```

The resulting similarity scores were used to rank songs in terms of relevance to the user’s query.

- **Sentiment Alignment for Emotional Consistency:** To ensure that the recommendations aligned with the emotional tone of the user input, sentiment alignment was applied. If the user query expressed a positive sentiment, the chatbot would recommend songs with a positive sentiment. This was implemented in the following code:

```
filtered_songs = filter_by_sentiment(sentiment, song_sentiments)
```

The function `filter_by_sentiment()` filtered songs based on sentiment, ensuring that the chatbot’s recommendations were emotionally consistent with the user’s request.

- **Thematic Matching via Named Entity Recognition (NER):** Custom NER labels were added to capture key themes in the lyrics (e.g., “love,” “heartbreak,” “friendship”). Using the `spaCy` library, entities were extracted from the lyrics to facilitate thematic filtering, as shown below:

```
doc = nlp(song_lyrics)
entities = [(ent.text, ent.label_) for ent in doc.ents]
```

This method allowed the chatbot to recommend songs based on thematic elements such as love or loss.

## 2.4 Experiments and Hyperparameter Tuning

In this project, several experiments were conducted to fine-tune the model and improve its performance. The main focus of the experiments was on the following aspects:

- **TF-IDF Parameters:** Different values for `max_df` and `min_df` were tested to see how they impacted the quality of song recommendations. A balance was sought between excluding overly common words (via `max_df`) and including sufficient unique terms (via `min_df`). The best-performing configuration was `max_df=0.85` and `min_df=0.01`, which provided a good trade-off between commonality and relevance.
- **Sentiment Analysis Model:** The effectiveness of `VADER` for sentiment analysis was compared with other models, such as `TextBlob` and `BERT`. However, due to the short and informal nature of song lyrics, `VADER` was chosen for its speed and accuracy on such data.

## 2.5 Model Comparison and Justification

Throughout the project, various models and methods were compared to assess their effectiveness for generating personalized song recommendations. Initially, simple methods like cosine similarity and sentiment classification were implemented. These were later combined with more sophisticated techniques, such as NER and thematic matching, to improve the accuracy and relevance of the chatbot’s recommendations.

The choice of methods was guided by the need to balance computational efficiency with the richness of the recommendations. The use of bigrams in the TF-IDF vectorization, the sentiment alignment, and the integration of NER were all driven by the goal of ensuring that the chatbot could understand and respond to both the emotional tone and the thematic content of the user’s query.

# 3 Exploratory Data Analysis (EDA)

This analysis aims to uncover meaningful patterns and trends in Taylor Swift’s lyrical content across her extensive discography, which will inform the development of a song recommendation chatbot. By examining various metrics, including lyric length, sentiment analysis, lexical richness, and entity recognition, we seek to understand both the evolution of Taylor Swift’s songwriting and the thematic elements within her lyrics. This deeper understanding will allow the chatbot to offer personalized, context-sensitive song recommendations based on both the emotional and thematic tones of her music.

## 3.1 Album Statistics

Table 1 summarizes key metrics for each of Taylor Swift’s albums, such as word count, unique words, average song length, and lexical richness. These metrics provide a quantitative overview of the evolution of both lyrical complexity and the production of her songs across her career. Taylor Swift’s early albums, such as *Taylor Swift* and *Fearless*, exhibit shorter average song lengths and fewer unique words, reflecting a simpler lyrical style. In contrast, later albums like *Speak Now*, *Red*, and *Reputation* demonstrate noticeable increases in both word count and lexical diversity, marking a shift toward more detailed and complex songwriting.

A significant outlier is *The Tortured Poets Department*, which has a much larger word count (5288 words) and a higher song count (31) compared to the earlier albums. This indicates a shift toward more expansive lyricism, with a deeper focus on thematic development across a larger volume of tracks.

### 3.1.1 Word Count and Song Length

One of the most apparent trends across Taylor Swift's albums is the increase in lyric length and song count. Early albums like *Taylor Swift* (1953 words) and *Fearless* (1955 words) feature relatively short lyrics, which is consistent with the country-pop style of these records. However, albums like *Speak Now* (2617 words) and *Red* (3042 words) show a clear evolution toward longer songs with more intricate lyrics. *The Tortured Poets Department*, with 5288 words and 31 tracks, stands out as having the most extensive lyrics by far, marking a significant departure from the shorter, more straightforward songs of earlier albums.

### 3.1.2 Unique Words and Lexical Richness

The analysis of unique words across albums provides insight into the increasing lyrical complexity in Swift's songwriting. Albums such as *Speak Now* and *Reputation* demonstrate a sharp increase in the number of unique words, reflecting a more diverse and expansive vocabulary. On the other hand, albums like *1989* and *Red* have fewer unique words, which can be attributed to the more accessible and repetitive lyrical style associated with pop music. *The Tortured Poets Department* again leads with the highest unique word count (1642), reflecting the album's more intricate lyricism and richer vocabulary.

### 3.1.3 Average Song Length

Song length, measured by the average number of words per song, has evolved over time. Swift's early albums, such as *Taylor Swift* (130.2 words per song) and *Fearless* (150.38 words per song), feature relatively short tracks. However, albums like *Red* (160.11 words per song) and *Reputation* (210.56 words per song) exhibit longer tracks with more detailed content. Despite *The Tortured Poets Department* featuring the longest total word count, the average song length (170.58 words) remains comparable to that of other albums, suggesting a balanced distribution of longer songs across the tracklist.

### 3.1.4 Lexical Richness

Lexical richness, or the ratio of unique words to total words, varies significantly across Taylor Swift's albums. *The Tortured Poets Department* boasts the highest lexical richness (0.6313), reflecting its expansive and varied vocabulary. Other albums like *Folklore* (0.5995) and *Evermore* (0.556) also show high lexical richness, indicating a return to more complex lyricism reminiscent of her earlier works. Conversely, albums like *1989* (0.3451) and *Red* (0.4763) demonstrate lower lexical richness, signaling a shift toward simpler and more repetitive lyrics typical of mainstream pop.

### 3.1.5 Song Length and Lexical Richness Range

The variation in song lengths and lexical richness across albums provides additional insight into the diversity of Taylor Swift's songwriting style. For instance, *1989* features a wide range of song lengths and lexical richness scores, indicating a blend of both simple and complex lyrical content. In contrast, albums like *Speak Now* and *Midnights* exhibit narrower ranges in these metrics, suggesting a more consistent songwriting style across the tracklist.

### 3.1.6 Evolution Over Time

The evolution of Taylor Swift's songwriting is clearly evident in the data. Her earlier albums, such as *Fearless* and *Speak Now*, feature intricate storytelling with higher lexical richness, whereas later albums like *1989* and *Reputation* adopt a more straightforward, pop-oriented style. However, albums like *Folklore* and *Evermore* represent a return to more introspective and reflective lyricism, demonstrating Swift's ability to balance commercial appeal with deeper thematic exploration. This evolution reflects her growing versatility as an artist and her ability to adapt her songwriting to fit changing musical landscapes.

### 3.2 Sentiment Analysis

Sentiment analysis (Table 2) reveals the emotional tone of Taylor Swift's lyrics across albums. Sentiment polarity measures whether lyrics are generally positive or negative, while subjectivity indicates how personal or emotional the lyrics are. Albums like *Red* and *Taylor Swift* have high sentiment polarity scores, reflecting a generally positive emotional tone, while albums like *1989* and *The Tortured Poets Department* show more negative sentiment, suggesting a more somber or introspective mood.

Subjectivity is highest in *Taylor Swift* (0.6245), indicating a deeply personal and reflective tone in many of the songs, while albums like *Lover* (0.5328) have lower subjectivity, suggesting a more neutral or objective tone. This variability in sentiment and subjectivity across albums allows for the development of a recommendation system that can tailor song suggestions based on a user's emotional state or mood.

### 3.3 Top Songs by Sentiment and Lexical Richness

The top songs by sentiment polarity (Table 3) show the emotional extremes in Taylor Swift's catalog, with songs like "Welcome to New York" and "London Boy" representing some of her most positive and celebratory works. In contrast, the top songs by lexical richness (Table 4) highlight tracks like "The Manuscript" from *The Tortured Poets Department*, which display a high level of linguistic complexity and introspection.

These contrasting patterns between sentiment polarity and lexical richness underscore the dual nature of Taylor Swift's music: emotionally uplifting and intellectually stimulating. The chatbot can use these differences to provide users with more tailored song recommendations that match both their mood and their desired level of lyrical complexity.

### 3.4 Top 10 Most Frequent Entities

Named Entity Recognition (NER) helps identify the most frequently mentioned entities in Taylor Swift's lyrics, such as "first," "one," "yesterday," and "night." These entities frequently appear in songs centered around themes of relationships, time, and personal reflection. Understanding the frequency and context of these entities will allow the chatbot to recommend songs that align with specific themes of interest, such as love, nostalgia, or personal growth. For example, a user expressing interest in themes of personal reflection might receive recommendations for songs from albums like *Red* or *Fearless*, which explore such concepts.

### 3.5 Visualizations and Insights

Graphs and co-occurrence matrices have been generated to identify patterns in the relationships between frequently occurring entities across albums. These visualizations provide deeper insight into how entities like "first," "one," and "yesterday" co-occur in various songs. Such patterns will help refine the chatbot's recommendation engine by identifying keywords that resonate with specific emotional or thematic preferences.

For instance, if a user is interested in themes of romantic beginnings, the chatbot could suggest songs where "first" and "one" frequently co-occur. Likewise, songs that contain the entities "yesterday" and "night" might be recommended to users seeking nostalgic or reflective tracks.

### 3.6 Exploratory Data Analysis Key Findings

The exploratory data analysis (EDA) provides valuable insights into Taylor Swift's songwriting across her career. By analyzing metrics such as album statistics, sentiment analysis, lexical richness, and named entity recognition, we can inform the design of a recommendation system for the chatbot. The EDA reveals the evolution of Taylor Swift's music from simpler, more personal early works to more complex, reflective lyrics in her later albums. These findings will guide the chatbot in making personalized, context-aware song recommendations based on a user's emotional tone, lyrical complexity preferences, and thematic interests.

This analysis forms the foundation for developing a chatbot that is capable of responding dynamically to user queries by leveraging insights from Taylor Swift's lyrical content. By understanding patterns in sentiment, thematic elements, and linguistic complexity, the chatbot can provide song suggestions that align with users' moods, preferences, and specific prompts. For example, a user expressing feelings of nostalgia

or heartbreak could receive recommendations tailored to these emotions, while another seeking lyrically rich and introspective tracks might be guided to songs from albums like *\*Folklore\** or *\*Evermore\**.

The EDA also supports the chatbot's ability to engage users in meaningful conversations by highlighting key themes, entities, and emotional tones in Swift's music. By connecting these elements with user inputs, the chatbot becomes not just a tool for song recommendations but also a companion that resonates with the depth and relatability of Taylor Swift's discography. Ultimately, the insights gained from this analysis empower the chatbot to deliver an enriching and highly personalized user experience.

## 4 Results

This section presents the outcomes of the advanced song recommendation system based on Taylor Swift's lyrics. The analysis includes sentiment classification, theme filtering, and album preferences, as well as the system's ability to handle user inputs dynamically. For further details, please refer to the GitHub repository dedicated to this assignment, which can be accessed at the following URL: <https://github.com/amerchant23/MSDS-453-Final-Project>.

### 4.1 Sentiment Distribution of Songs

Before presenting specific recommendations, the sentiment distribution of Taylor Swift's catalog was analyzed using polarity values. This categorization provided insights into the emotional tone of the lyrics, a key component of the recommendation system. The sentiment distribution, as illustrated in Figure 4, is as follows:

- **Positive Sentiment:** Approximately 35% of the songs carried a positive sentiment, reflecting uplifting and empowering themes often present in Taylor Swift's lyrics.
- **Negative Sentiment:** A smaller subset, 25%, displayed negative sentiment, commonly exploring themes of heartbreak and emotional challenges.
- **Neutral Sentiment:** The remaining 40% of songs were neutral, balancing upbeat and somber tones and often focusing on narrative storytelling.

These findings informed the sentiment filter, enabling the system to recommend songs that align with the user's preferred emotional tone. For instance, users interested in positive or introspective lyrics could receive recommendations tailored to those sentiments.

### 4.2 Song Recommendation Outputs

The recommendation system was tested with various user queries, incorporating sentiment, album, and thematic filters. Several examples showcasing the system's performance are detailed below:

#### 4.2.1 Example 1: Input with Filters

**User Input:** *Love Story*

**Filters Applied:** Album: *Fearless*, Sentiment: Positive, Themes: "love," "fairytale."

**Recommendations:**

1. *The Best Day* (Similarity: 0.10)
2. *White Horse* (Similarity: 0.09)
3. *The Way I Loved You* (Similarity: 0.09)
4. *Fifteen* (Similarity: 0.06)

The system successfully matched the user's preferences, providing recommendations aligned with sentiment and thematic filters within the specified album.



### 4.2.2 Example 2: Handling Typographical Errors

It is important to note that the song titles in the dataset are normalized to ensure consistency in formatting. During the preprocessing stage, certain characters (such as spaces and special characters) were removed, and camel case was split into separate words. This normalization ensures that titles are processed uniformly, but can occasionally result in variations in the song names.

For instance, the song "Guilty as Sin" appears in the dataset as "Guiltyas Sin". This is due to the way the data was read and normalized, where spaces between words were removed. The absence of spaces and the merging of words, such as "Guiltyas Sin", is a result of the automated normalization process intended to standardize song titles across the dataset.

Despite these normalization inconsistencies, the chatbot effectively handles user input through fuzzy matching. When users input a song title, such as "Guilty as Sin", which might not match the exact normalized title in the dataset, fuzzy matching is used to detect and correct minor discrepancies in spelling or spacing. This ensures that users can still find the song they are looking for, even if the title is entered differently than it appears in the dataset. For example, when the chatbot does not find "Guilty as Sin", it suggests "Guiltyas Sin", which is the normalized form in the dataset.

**User Input:** *Guilty as Sin*

**System Response:** "Could not find *Guilty as Sin*. Did you mean *Guiltyas Sin*?"

**User Input:** *yes*

**System Response:** "Found your song: *Guiltyas Sin*"

**Filters Applied:** Sentiment: Negative

**Recommendations:**

1. *Than Kyoua Imee* (Similarity: 0.08)

By using fuzzy matching, the chatbot improves the user experience, allowing for greater flexibility and ensuring that recommendations are still relevant, even when the user's input contains slight spelling or formatting variations.

### 4.2.3 Example 3: Overly Narrow Filters

**User Input:** *Midnight Rain*

**Filters Applied:** Album: *Reputation*, Sentiment: Neutral

**System Response:** "No recommendations found with the specified filters. Try adjusting or relaxing the filters for broader results."

The system handled overly restrictive filters by suggesting broader search parameters, thus enhancing user interaction.

### 4.2.4 Example 4: Adjusted Filters for Broader Results

**User Input:** *Midnight Rain*

**Filters Applied:** Album: *Fearless*

**Recommendations:**

1. *Forever and Always* (Similarity: 0.09)
2. *The Way I Loved You* (Similarity: 0.06)
3. *Change* (Similarity: 0.05)

Relaxing the filters resulted in broader recommendations, demonstrating the adaptability of the system.

#### 4.2.5 Interpretations and Insights

- **Sentiment Analysis:** The sentiment filter effectively tailored recommendations to the user's emotional preferences, enhancing the relevance of the results.
- **Theme-Based Filtering:** Theme keywords allowed for deeper customization, enabling users to receive recommendations aligned with narrative themes such as "love" or "fairytale."
- **User Experience:** The system displayed robustness in handling user inputs, including typographical errors, and ensured flexibility through fuzzy matching techniques.

### 4.3 Visual Results: Sentiment Distribution

The sentiment analysis results were visualized in a pie chart (Figure 4), illustrating the proportions of positive, negative, and neutral songs in Taylor Swift's discography. This visualization highlights the emotional diversity of the catalog and supports sentiment-based recommendations.

## 5 Conclusion

This project demonstrates the potential of natural language processing techniques in creating a personalized chatbot for song recommendations based on Taylor Swift's lyrics. By leveraging advanced methodologies such as TF-IDF vectorization, sentiment analysis, and thematic entity recognition, the chatbot offers users an engaging and context-sensitive experience. Key insights from exploratory data analysis, including the evolution of lyrical complexity, sentiment trends, and thematic diversity, were instrumental in designing a model that aligns with the unique characteristics of Taylor Swift's music.

The integration of cosine similarity for lyric matching and sentiment alignment for emotional consistency ensures that recommendations are not only accurate but also meaningful. Furthermore, the inclusion of customizable filters empowers users to refine their recommendations, enhancing the interactivity and personalization of the chatbot.

Through the application of machine learning tools like SpaCy, Scikit-Learn, and VADER, this project bridges the gap between technology and creativity, showcasing the versatility of NLP in the entertainment domain. Beyond its technical contributions, the chatbot provides a deeper appreciation of Taylor Swift's artistry by enabling fans to explore her music in a novel and intuitive way.

In conclusion, this project highlights the transformative role of artificial intelligence in personalizing user experiences, particularly in creative industries. The successful implementation of this chatbot sets a foundation for future innovations in music recommendation systems, illustrating how data-driven approaches can deepen fan engagement and redefine content interaction.

### 5.1 Directions for Future Work

Future developments of the Taylor Swift song recommendation chatbot could focus on incorporating more advanced machine learning models, such as deep learning techniques for sentiment analysis and lyric generation. Enhancements to the thematic matching system, such as using more sophisticated named entity recognition (NER) models or exploring topic modeling techniques, could provide even more personalized recommendations. Additionally, integrating user feedback into the recommendation system would allow the chatbot to continuously improve its accuracy over time. Expanding the dataset to include Taylor Swift's "Taylor's Version" albums and incorporating real-time music trends would also enhance the chatbot's relevance and engagement.

# A Appendix

## A.1 Tables

Album	Total Word Count	Total Unique Words	Avg. Song Length (words)	Avg. Lexical Richness	Song Count
Taylor Swift	1953	490	130.2	0.5149	15
Fearless	1955	572	150.38	0.5501	13
Speak Now	2617	752	186.93	0.5126	14
Red	3042	730	160.11	0.4763	19
1989	2733	531	210.23	0.3451	13
Reputation	3369	840	210.56	0.4291	16
Lover	3163	819	175.72	0.4871	18
Folklore	2249	843	140.56	0.5995	16
Evermore	2390	838	159.33	0.5560	15
Midnights	2024	675	155.69	0.5147	13
The Tortured Poets Department	5288	1642	170.58	0.6313	31

Table 1: Summary Statistics of Taylor Swift Albums

Album	Average Polarity	Average Subjectivity
Taylor Swift	0.81154	0.624533
Fearless	0.733638	0.601923
Speak Now	0.302007	0.618786
Red	0.831058	0.592947
1989	0.156092	0.545923
Reputation	0.775319	0.591000
Lover	0.450711	0.532833
Folklore	0.270962	0.619812
Evermore	0.271387	0.563533
Midnights	0.532585	0.597077
The Tortured Poets Department	0.11439	0.60371

Table 2: Sentiment Analysis Output by Album

Table 3: Top 10 Songs with the Highest Sentiment Polarity

Song	Sentiment Polarity	Album
Welcome to New York	0.9991	1989
London Boy	0.9991	Lover
Out of the Woods	0.9989	1989
Dress	0.9985	Reputation
This Is Why We Can't Have Nice Things	0.9982	Reputation
Stay Beautiful	0.9981	Taylor Swift
Fearless	0.9980	Fearless
Everything Has Changed	0.9979	Red
Call It What You Want	0.9978	Reputation
Hey Stephen	0.9977	Fearless

Table 4: Top 10 Songs with the Highest Lexical Richness

<b>Song</b>	<b>Lexical Richness</b>	<b>Album</b>
The Manuscript	0.8673	The Tortured Poets Department
The Smallest Man Who Ever Lived	0.7943	The Tortured Poets Department
How Did It End	0.7699	The Tortured Poets Department
Loml	0.7596	The Tortured Poets Department
Fresh Out The Slammer	0.7589	The Tortured Poets Department
Champagne Problems	0.7569	Evermore
The Black Dog	0.7569	The Tortured Poets Department
You're On Your Own Kid	0.7552	Midnights
The Best Day	0.7548	Fearless
Invisible String	0.7518	Folklore

## A.2 Visuals

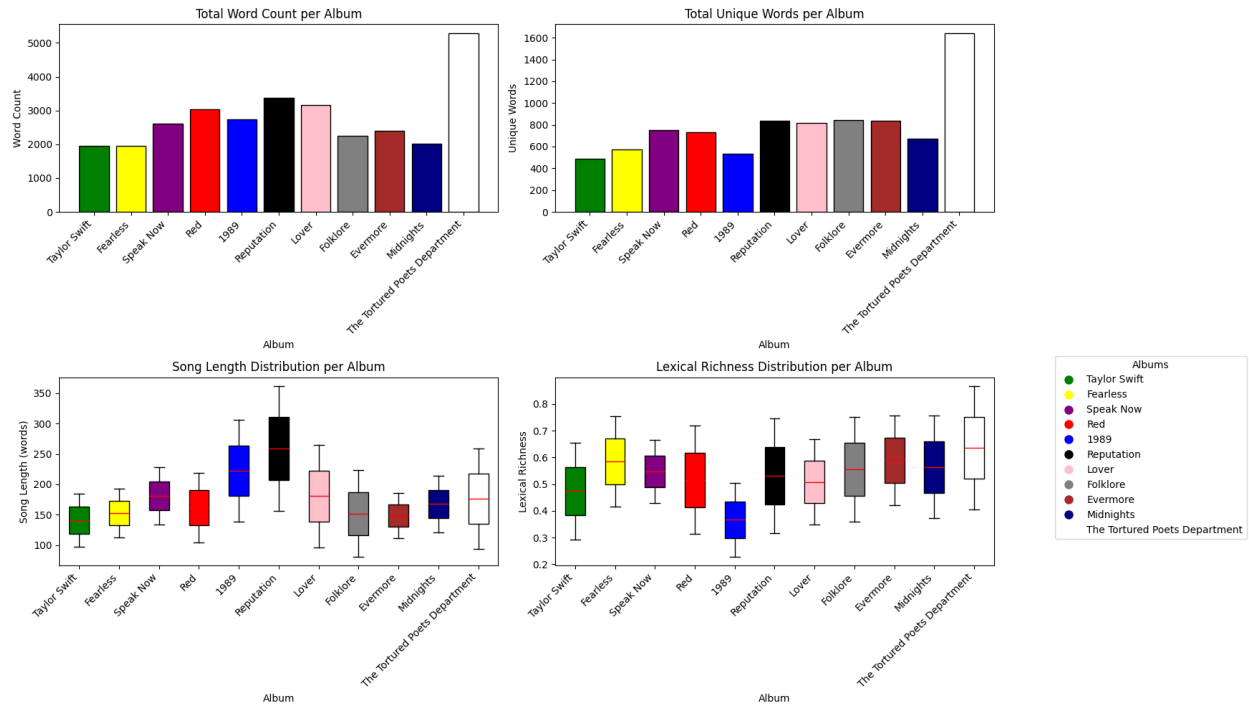


Figure 1: Album Statistics

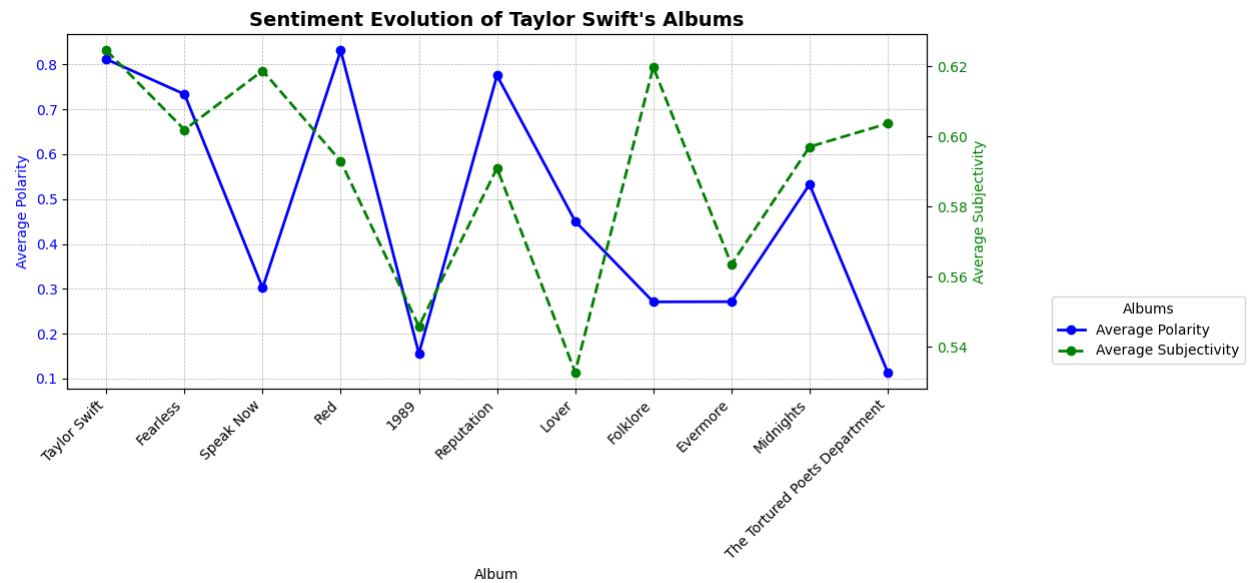


Figure 2: Sentiment Evolution of Taylor Swift's Albums

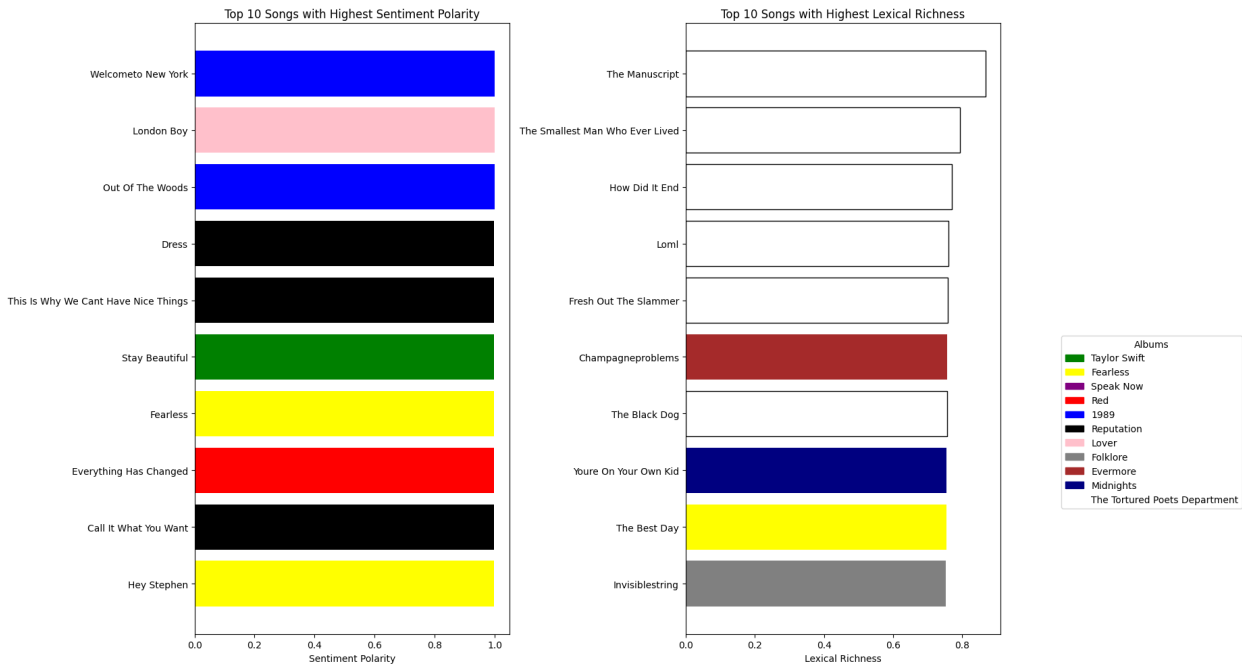


Figure 3: Top 10 Songs

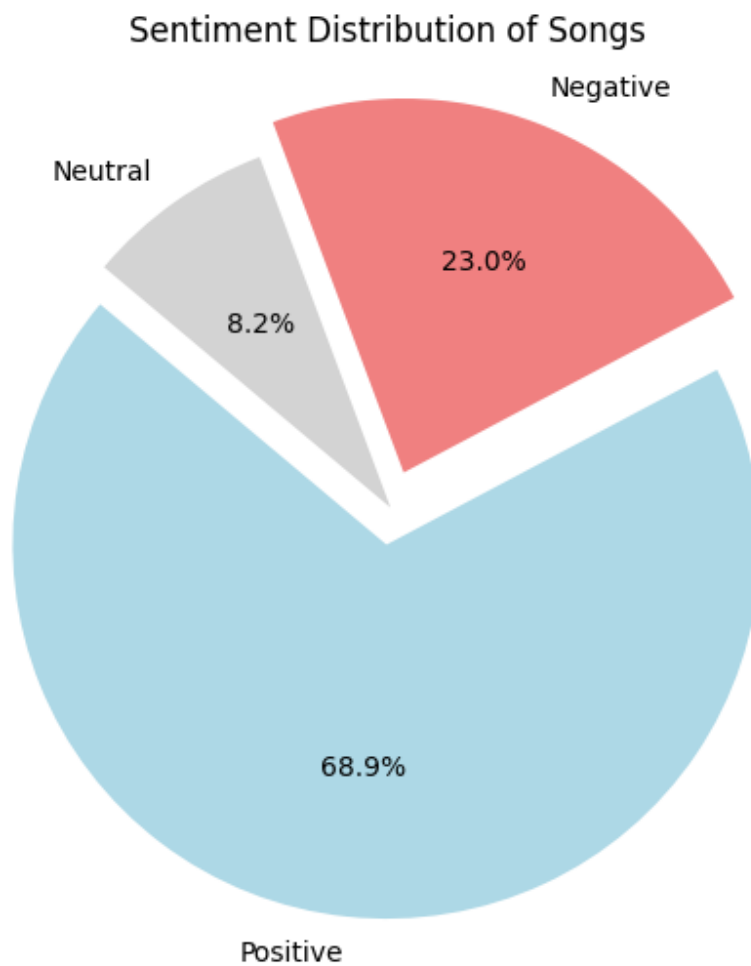


Figure 4: Sentiment Distribution of Songs

### A.3 Advanced Swiftie Lyric Bot Interaction

#### Welcome to the Advanced Swiftie Lyric Bot!

Looking for personalized song recommendations? Let's get started!

Type the name of a Taylor Swift song, and I'll recommend others.

Add filters like album, sentiment, or themes for a tailored experience.

Type 'exit' anytime to leave.

**Your song choice (or type 'exit'):** Love Story

Found your song: Love Story

Do you want a single song recommendation or a playlist? (Type 'single', 'playlist', or 'exit'): playlist

Want to filter by album? Type the album name or press Enter to skip (or type 'exit'): Fearless

Sentiment preference (positive/negative/neutral)? Press Enter to skip or type 'exit': positive

Key themes or keywords to match? Separate by commas or press Enter to skip (or type 'exit'): love, fairytale

Normalized input: Love Story

Theme keywords: ['love', 'fairytale']

Here are your top recommendations for 'Love Story':

1. The Best Day (Similarity: 0.10)
2. White Horse (Similarity: 0.09)
3. The Way I Loved You (Similarity: 0.09)
4. Fifteen (Similarity: 0.06)

**Your song choice (or type 'exit'):** Guilty as sin

Couldn't find 'Guilty as sin'. Did you mean 'Guilty as Sin'? (yes/no or type 'exit'): yes

Found your song: Guilty as Sin

Do you want a single song recommendation or a playlist? (Type 'single', 'playlist', or 'exit'): single

Want to filter by album? Type the album name or press Enter to skip (or type 'exit'):

Sentiment preference (positive/negative/neutral)? Press Enter to skip or type 'exit': negative

Key themes or keywords to match? Separate by commas or press Enter to skip (or type 'exit'):

Normalized input: Guilty as Sin

Here are your top recommendations for 'Guilty as Sin':

1. Than Youa Imee (Similarity: 0.08)

**Your song choice (or type 'exit'):** Peter

Found your song: Peter

Do you want a single song recommendation or a playlist? (Type 'single', 'playlist', or 'exit'): playlist

Want to filter by album? Type the album name or press Enter to skip (or type 'exit'):

Sentiment preference (positive/negative/neutral)? Press Enter to skip or type 'exit':

Key themes or keywords to match? Separate by commas or press Enter to skip (or type 'exit'): person

Normalized input: Peter

Theme keywords: ['person']

No recommendations found with the specified filters. This could be because:

1. **Sentiment:** The selected sentiment might not match any songs.
  2. **Combination of filters:** The combination of album, sentiment, or theme filters may be too narrow.
- Try adjusting the filters or removing some of them for broader results!

**Your song choice (or type 'exit'):** Peter

Found your song: Peter

Do you want a single song recommendation or a playlist? (Type 'single', 'playlist', or 'exit'): playlist

Want to filter by album? Type the album name or press Enter to skip (or type 'exit'):

Sentiment preference (positive/negative/neutral)? Press Enter to skip or type 'exit':

Key themes or keywords to match? Separate by commas or press Enter to skip (or type 'exit'): name

Normalized input: Peter

Theme keywords: ['name']

Here are your top recommendations for 'Peter':



1. Blank Space (Similarity: 0.15)
2. Ready for It (Similarity: 0.13)
3. Better Than Revenge (Similarity: 0.12)
4. Everything Has Changed (Similarity: 0.12)
5. Hey Stephen (Similarity: 0.07)
6. Than Kyoua Imee (Similarity: 0.06)
7. Guilt yas Sin (Similarity: 0.05)
8. All Too Well (Similarity: 0.04)
9. Who's Afraid of Little Old Me (Similarity: 0.04)
10. The Last Great American Dynasty (Similarity: 0.04)

**Your song choice (or type 'exit'):** midnight

Couldn't find 'midnight'. Did you mean 'Midnight Rain'? (yes/no or type 'exit'): yes

Found your song: Midnight Rain

Do you want a single song recommendation or a playlist? (Type 'single', 'playlist', or 'exit'): single

Want to filter by album? Type the album name or press Enter to skip (or type 'exit'): Reputation

Sentiment preference (positive/negative/neutral)? Press Enter to skip or type 'exit': neutral

Key themes or keywords to match? Separate by commas or press Enter to skip (or type 'exit'):

Normalized input: Midnight Rain

No recommendations found with the specified filters. This could be because:

1. **Sentiment:** The selected sentiment might not match any songs.
2. **Combination of filters:** The combination of album, sentiment, or theme filters may be too narrow. Try adjusting the filters or removing some of them for broader results!

**Your song choice (or type 'exit'):** midnight rain

Couldn't find 'midnight rain'. Did you mean 'Midnight Rain'? (yes/no or type 'exit'): yes

Found your song: Midnight Rain

Do you want a single song recommendation or a playlist? (Type 'single', 'playlist', or 'exit'): playlist

Want to filter by album? Type the album name or press Enter to skip (or type 'exit'): Fearless

Sentiment preference (positive/negative/neutral)? Press Enter to skip or type 'exit':

Key themes or keywords to match? Separate by commas or press Enter to skip (or type 'exit'):

Normalized input: Midnight Rain

Here are your top recommendations for 'Midnight Rain':

1. Forever Always (Similarity: 0.09)
2. The Way I Loved You (Similarity: 0.06)
3. Change (Similarity: 0.05)
4. Hey Stephen (Similarity: 0.04)
5. Breathe (Similarity: 0.04)
6. White Horse (Similarity: 0.04)
7. The Best Day (Similarity: 0.03)
8. Tell Me Why (Similarity: 0.02)
9. You Belong With Me (Similarity: 0.02)
10. Love Story (Similarity: 0.01)

**Your song choice (or type 'exit'):** exit

Thanks for chatting! 'You've got no reason to be afraid, you're on your own, kid