Song Recommendation System Based on Caption

By

Lee Hui Xing

FACULTY OF COMPUTING AND INFORMATION TECHNOLOGY

TUNKU ABDUL RAHMAN UNIVERSITY OF MANAGEMENT AND TECHNOLOGY
KUALA LUMPUR

ACADEMIC YEAR
2024/25

Song Recommendation System Based on Caption

By

Lee Hui Xing

Supervisor: Ms Nik Nur Adlin Binti Nik Qausbee

A project report submitted to the
Faculty of Computing and Information Technology
in partial fulfillment of the requirement for the
Bachelor of Computer Science (Honours)

Department of Mathematical and Data Science
Faculty of Computing and Information Technology
Tunku Abdul Rahman University of Management and Technology
Kuala Lumpur

Declaration

The project submitted herewith is a result of my own efforts in totality and in every aspect of the project works. All information that has been obtained from other sources had been fully acknowledged. I understand that any plagiarism, cheating or collusion or any sorts constitutes a breach of TAR University rules and regulations and would be subjected to disciplinary actions.

Lee Hui Xing

Bachelor of Computer Science (Honours) in Data Science

ID: 22WMR13582

Abstract

Social media platforms have become an integral part of daily communication, often serving as a place for users to express their emotions through captions. However, despite advancements in personalization technology, there is still a gap in systems that recommend songs to match the sentiment and context of user-generated captions. This project addresses this gap by designing a recommendation system that aligns social media captions with songs, thereby enhancing user engagement and emotional resonance.

This project integrates techniques from natural language processing (NLP) and information retrieval to achieve its objectives. It incorporates sentiment analysis using VADER to classify the emotional tone of captions and contextual analysis leveraging topic modeling techniques like Latent Dirichlet Allocation (LDA) to extract relevant themes and keywords. The Spotify API is used to retrieve song metadata, such as genre, energy, and acousticness, which serve as input features for the recommendation engine. Integration of TF-IDF is utilized to measure the semantic alignment between captions and song features.

The system is modular, comprising components for data collection, preprocessing, sentiment analysis, contextual analysis, and song recommendation generation. Sample datasets are thoroughly tested, including captions and song data from public platforms such as Kaggle. The testing evaluates the system’s ability to recommend related song genres and songs that effectively capture the sentiment and context of captions.

The results indicate the system's effectiveness in creating relevant genre recommendations, although it has difficulties when dealing with complicated sarcasm or extremely nuanced language. This project demonstrates how integrating NLP techniques with music metadata analysis can bridge the gap between text-based and audio-based personalization, providing an innovative solution to improve user experiences on social media platforms. Future work aims to refine contextual knowledge and expand the recommendation scope across multiple platforms.

Acknowledgement

I would like to express my heartfelt gratitude to everyone who supported and guided me throughout the course of this project.

First, I would like to thank my Final Year Project supervisor, Ms Nik Nur Adlin Binti Nik Qausbee, for her invaluable guidance, encouragement, and constructive feedback. Her knowledge and guidance have played a crucial role in influencing the development and implementation of this project.

Special thanks to my family for their unwavering support and encouragement, especially during challenging moments. Their belief in me kept me motivated and focused throughout this project.

I would also like to thank my friends for their supportations, invaluable suggestions and for being a source of inspiration and positivity throughout this project.

This project would not have been possible without the collective efforts and support of all these individuals and resources. To all, I extend my deepest gratitude.

Table of Contents

Chapter 1

Introduction

1 Introduction

Songs are essential in improving social media posts as they make them more emotionally impactful and immersive. By matching the right song with a caption, users can enhance their message, evoke desired emotions, and create a deeper connection with their audience. Despite the importance of music on social media, the process of finding the ideal song to pair with a caption is still a tiresome and time-consuming task. Users often have to rely on their own knowledge or search through trending songs, which can be limiting and frustrating. This is where a more efficient and intelligent solution is needed, which would make the process of choosing songs easy and intuitive.

By automatically recommending songs based on the caption of an English social media post, the Song Recommendation System closes this gap. The system analyses the emotional tone and contextual content of the captions using Natural Language Processing and sentiment analysis techniques to choose songs that fit the themes or mood expressed. For instance, an enthusiastic, high-energy song could be appropriate for a celebratory post, while a peaceful, low-energy tune might be appropriate for a caption about a leisurely evening. This guarantees that the suggested song expresses the user's desired feeling, increasing the impact and engagement of their social media posts.

This system improves the overall social media experience while saving users time by simplifying the song selection process. By introducing users to songs they might not have discovered otherwise, it broadens their musical horizons and eliminates the need for tedious manual searches. Additionally, the system's personalized and intuitive approach fosters creativity, enabling users to express themselves more effectively through the combination of music and text.

Furthermore, this project highlights the potential of context-aware recommendation systems, which go beyond traditional methods by combining real-time data and emotional insights. Unlike traditional music recommendation systems, which rely on user listening history or genre preferences, this system incorporates the content and sentiment of user-generated language. By addressing the gap between technology and human creativity, it provides an innovative way to improve self-expression on social media platforms, establishing the way for future sentiment-based recommendation applications in other fields.

1.1 Objectives

1.1.1 Sentiment Analysis

Sentiment Analysis is a key technique in Natural Language Processing, which will be used to determine the emotional tone of the text is positive, negative, or neutral. The objective of this phase includes text preprocessing (clean the text, stop word removal, punctuations, and unnecessary characters removal) and sentiment classification using machine learning models (logistic regression, random forest, SVM, neural network).

1.1.2 Contextual Analysis of Captions

Identifying the key words, topics and themes provides further context for this recommendation system. The objective of this phase includes keyword extraction (TF-IDF) and topic modelling (LDA) to improve the accuracy of personalized song recommendation by adding a deeper understanding of the caption.

1.2 Background

1.2.1 Problem Statement

In today's social media-driven world, social media platforms like Instagram play a significant role in personal expression, allowing users to share stories, moments, and thoughts through visual contents along with captions. Music is a powerful tool that enhances the posts, adding emotional depth and helping users to convey their feelings, evoke emotions from the viewer and highlights important moments. However, the current process of selecting songs to match with the social media posts is manual, requiring users to search and choose songs by themselves without any intelligent suggestions based on their captions. This leads to a disconnect as users must decide the songs on their own instead of having suggestions that reflect the context or emotional tone of their posts. Users may not have an exact idea to choose for a suitable song, they can only choose from songs that they know or trending songs, as the current feature only recommends popular songs of the moment. Users have to browse and preview the songs first if there is no targeted song, despite the platforms classifying the songs into genres, mood and themes. This process can be time-consuming, as users must sift through various options, manually searching for a song that fits the tone of their post, often leading to frustration and delayed content creation. Besides, personalized suggestions based on listening history are provided by music streaming services like Spotify and YouTube, but they do not take into consideration the context of real-time content creation, like social media captions. Therefore, a song recommendation system based on caption could addresses the gap and provide a more seamless user experience.

1.3 Advantages and Contributions

Users can receive song genre and related song suggestions instantly as they type their captions, allowing for more creative content creation and saves time. This immediate feedback ensures that users can quickly find songs that match the mood and context of their posts, enhancing the overall user experience. Additionally, this recommendation system promotes wider music exploration by exposing users to new and relevant songs they may not have considered, which is in contrast to conventional recommendation systems that rely on past listening history or genre preferences. This encourages listening to and discovering a wider variety of music.

1.4 Project Plan

1.4.1 Project Scope

In this project, I am going to develop a Song Recommendation System based on captions that includes two significant processes which are sentiment analysis and contextual analysis of captions. Before performing the analysis, data collection is to be carried out. Then, the data is pre-processed to be used in recommendation algorithms. Machine learning models are then applied to match and recommend songs. After that, the algorithms will be evaluated using the performance metrics.

1.4.2 Milestone

Task	Start date	End date	Duration (weeks)
Project I			
Proposal	4/9/2024	30/9/2024	4
Introduction	1/10/2024	19/10/2024	3
Literature review	20/10/2024	4/11/2024	2
Methodology	5/11/2024	26/11/2024	3
Project II			
Data Collection	8/12/2024	22/12/2024	2
Data Preprocessing	24/12/2024	30/12/2024	1
Sentiment Analysis	3/1/2025	18/1/2025	2
Contextual Analysis	19/1/2025	2/2/2025	2
System Implementation	10/2/2025	4/3/2025	3
System testing	5/3/2025	12/3/2025	1
Discussions and Conclusion	20/3/2025	1/4/2025	2

1.5 Chapter Summary and Evaluation

In this chapter, we have known that the recommendation system aims to match songs with the emotional tone and themes of user-generated text using Natural Language Processing techniques and sentiment analysis. To improve the accuracy of recommendation, the key objectives including sentiment and contextual analysis are outlined. From the background, the motivations of this system are to provide real-time personalized music suggestions, enhance user experience by saving time and promoting broader music discovery. The project plan included a milestone which listed the duration taken for each stage of this project.

Chapter 2

Literature Review

2 Literature Review

In this chapter, key concepts and theories related to this work will be discussed. In addition, previous related studies will be explored.

2.1 Concept and Theory

2.1.1 Recommendation Systems

A recommendation system is a computer-based technique that is intelligent and predicts based on the adoption and utilization of users, enabling them to select items from a vast inventory of online content (Singh et al., 2021). There are three recommendation systems approaches: Content-based Filtering, Collaborative Filtering and Hybrid approach. In content-based recommender systems, items are grouped into profiles based on their features. When a user rates an item positively, similar items from that profile are combined to create a user profile, which aggregates all positively rated item profiles. Collaborative filtering relies on measuring similarity between users, identifying a ‘neighbourhood’ of users with preferences similar to the target user. Items favoured by this neighbourhood are then recommended to the target user. The effectiveness of collaborative filtering depends on accurately identifying this neighbourhood. A hybrid technique combines two or more recommender methods to overcome the limitations of individual approaches. This can involve integrating results from separate techniques, using content-based filtering within a collaborative framework, or applying collaborative filtering within a content-based approach. Such hybrid methods typically improve both performance and accuracy in recommender systems (Roy & Dutta, 2022).

2.1.2 Sentiment Analysis

Sentiment analysis, as known as opinion mining, is a task of Natural Language Processing (NLP) that involves examining people’s viewpoints and emotions towards subjects, like products or events to classify whether the text expresses positive or negative sentiments (Liu, B., 2022). Most of the literature typically categorizes sentiment analysis methods into three main types: machine learning approaches, lexicon-based approaches, and hybrid approaches. The most commonly used approach is machine learning, which utilizes algorithms and linguistic features for sentiment classification. In contrast, the lexicon-based approach relies on a sentiment lexicon—a collection of words and phrases that typically convey positive or negative sentiments. Meanwhile, hybrid approaches enhance sentiment analysis performance by combining machine learning with lexicon-based techniques (Birjali et al., 2021).

Sentiment analysis has a relevance in song recommendation systems by enabling a deeper alignment of music choices with listener’s emotional states. By using sentiment analysis, it allows for an understanding of whether song lyrics carry positive, neutral, or negative emotions, so that recommendation systems can create more accurate, mood-congruent playlists (Sarin et al., 2022).

2.2 Related Works

2.2.1 Music Recommendation Systems

Numerous previous works have explored various approaches to music recommendation to improve recommendation accuracy and personalization.

A research by (Fessahaye et al., 2019) stated that they developed an algorithm named Tunes Recommendation System (T-RECSYS) by using a combination of content-based and collaborative filtering to create a precise recommendation system with real-time prediction by feeding a deep learning classification model. Through training, a deep neural network that receives this information learns to identify patterns in a person's listening history and eventually suggests songs that it believes the user would like. They concluded that the system achieved high precision scores, showing that this approach can generate personalized, precise recommendations similar to popular systems like Spotify's Discover Weekly. T-RECSYS demonstrated potential for real-time updates and scalability to other domains, making it applicable to various platforms, including video and e-commerce recommendations (Fessahaye et al., 2019).

An emotional music recommender system that leverages user personality traits, moods, and emotions, offering a more tailored listening experience by embedding personality and behavioural insights within a content-based framework was introduced. This system used mood detection from recent listening history and integrated it into the recommendation process, enhancing the relevance of suggested songs in real-time. Their findings also underscored the value of incorporating psychological insights into recommendation frameworks to better meet user needs in real-time. The system’s overall mood detection accuracy was strong (Moscato et al., 2021).

A music recommender system using a Convolutional Recurrent Neural Network (CRNN) model to enhance genre-based recommendations was presented. By combining convolutional and recurrent layers, the CRNN model captures both the audio frequency features and temporal patterns within music, achieving improved genre classification accuracy compared to standard CNN models. This research concluded that users prefer recommendations that consider genre, indicating that genre-based systems like CRNNs offer enhanced user satisfaction in music recommendations (Adiyansjah et al., 2019).

Moreover, an interactive music recommendation system that integrates mood-based filtering with audio content-based recommendations named MoodPlay was introduced. It utilizes a hybrid cascading recommendation approach that first pre-filters artists based on mood similarity and then refines the results using audio similarity. It uses the Geneva Emotional Music Scale (GEMS) to categorize music by mood and improves user involvement with features such as mood-based trail recommendations and configurable mood influence. The study concluded that incorporating mood information improves user satisfaction and recommendation accuracy by effectively facilitate music discovery and addressing the cognitive challenges in traditional recommendation systems (Andjelkovic et al., 2019).

Authors	**Title**	**Findings**
Ferdos Fessahaye, Luis Perez, Tiffany Zhan, Raymond Zhang, Calais Fossier, Robyn Markarian, Carter Chiu, Justin Zhan, Laxmi Gewali, Paul Oh	T-RECSYS: A Novel Music Recommendation System Using Deep Learning	Developed T-RECSYS using a combination of content-based and collaborative filtering with a deep neural network for personalized recommendations.
Vincenzo Moscato, Antonio Picariello, Giancarlo Sperli	An Emotional Recommender System for Music	Leveraged user personality traits, moods, and emotions in a content-based framework with mood detection from recent listening history for real-time recommendations.
Adiyansjah, Alexander A S Gunawan, Derwin Suhartono	Music recommender system based on genre using convolutional recurrent neural networks	Utilized a Convolutional Recurrent Neural Network (CRNN) combining convolutional and recurrent layers to capture audio frequency features and temporal patterns for genre-based recommendations.
Ivana Andjelkovic, Denis Parra, John O’Donovan	Moodplay: Interactive music recommendation based on Artists’ mood similarity	Uses a hybrid cascading recommendation approach combining mood-based filtering (using Geneva Emotional Music Scale) and

		audio content-based recommendations for personalized music suggestions.

Table 2.1: Literature review summary table of Music Recommendation Systems

Based on the findings on music recommendation systems, most existing music recommendation systems focus on content-based and collaborative filtering techniques, often incorporating deep learning models for genre analysis, or use mood-based filtering tools for personalised recommendations. In contrast to these approaches, my proposed system integrates sentiment analysis and Natural Language Processing to match social media captions directly with suitable songs. This focuses on textual sentiment and caption context as the primary input, which differs from the audio-based or user history-driven frameworks seen in previous studies, making it a novel approach to enhance user experience on social media platforms.

2.2.2 Sentiment-Based Recommendations

Sentiment analysis has numerous applications in enhancing recommendation systems, helping deliver more personalized and relevant suggestions based on users' emotions.

Referring to (Osman et al., 2019), they proposed a contextual sentiment-based recommender system to provide recommendations in the electronic products domain. The sentiment-based model combined contextual information from textual reviews with ratings to address the issues faced in traditional systems such as data sparsity and domain insensitivity, which reduce recommendation accuracy. They concluded that their model effectively improves recommendation accuracy in the electronic products domain with better accuracy in predictions (Osman et al., 2019).

Besides, sentiment-based recommender system was applied in the food and beverage field. (Asani et al., 2021) introduced a context-aware restaurant recommender system that utilizes sentiment analysis to derive users’ food preferences from their comments, allowing for more personalized recommendations. By clustering similar food items and analysing sentiment scores from reviews, the system effectively aligns restaurant suggestions with individual user tastes. As a result, they concluded that their sentiment-based recommender system achieved high accuracy, with a precision of 92.8% in providing relevant restaurant suggestions (Asani et al., 2021).

Contextual awareness and sentiment analysis have been integrated into tourism recommendation systems over time, improving their ability to offer tailored suggestions. Referring to (Abbasi-Moud et al., 2021), they introduced a context-aware tourism recommendation system that utilizes semantic clustering and sentiment analysis to extract user preferences from textual reviews. The system could identify user preferences more accurately by clustering similar terms and applying sentiment analysis. In addition, it further adapts recommendations based on time, location, and weather conditions, ensuring that suggestions are relevant to the user's current context. This research concluded that their system showed superior accuracy, especially when generating top recommendations for nearby attractions (Abbasi-Moud et al., 2021).

According to (Kumar et al., 2020), sentiment analysis can also be utilized for movie recommendation. They use the data from microblogging platforms like Twitter, which contain rich, real-time public opinions about movies. The study integrates techniques such as data collection through APIs, sentiment classification (positive, neutral, negative) and collaborative filtering, which combines user preferences with general trends. Challenges associated with analysing unstructured and noisy text data from microblogs are also reviewed. The results demonstrate that incorporating sentiment analysis into movie recommendation systems improves the accuracy and relevance of recommendations compared to traditional methods.

A Credibility, Interest and Sentiment Enhanced Recommendation (CISER) model is introduced in the research by (Hu et al., 2020), which integrates user profiling, reviewer credibility analysis and fine-grained sentiment analysis for robust product recommendations. The model uses reviewer credibility scores, sentiment polarity, and user interest patterns derived from product reviews to recommend items effectively. Techniques such as fastText for fine-grained sentiment scoring and heuristic-driven user preference mining are critical components. The system assigns reviewers trust, knowledge, and influence scores to improve suggestion accuracy while reducing the impact of fraudulent or biased reviews. The CISER model significantly improves recommendation accuracy, with a mean average precision (MAP) of 93% for single product suggestions. The results suggest that combining sentiment analysis, reviewer reputation, and user interest patterns improves recommendation performance over standard methods.

Authors	**Title**	**Findings**
N. A. Osman, S. A. M. Noah, M. Darwich	Contextual sentiment based recommender system to provide recommendation in the electronic products domain	Combined contextual information from textual reviews with ratings in a sentiment-based model to address issues like data sparsity and domain

		insensitivity for improved recommendation accuracy.
Elham Asani, Hamed Vahdat-Nejad, Javad Sadri	Restaurant recommender system based on sentiment analysis	Applied sentiment analysis to derive food preferences from user comments, clustered similar food items, and analyzed sentiment scores for personalized restaurant recommendations.
Zahra Abbasi-Moud, Hamed Vahdat-Nejad, Javad Sadri	Tourism recommendation system based on semantic clustering and sentiment analysis	Used semantic clustering and sentiment analysis to extract user preferences from textual reviews. Adapted recommendations based on time, location, and weather conditions for tailored suggestions.
Sudhanshu Kumar, Kanjar De, Partha Pratim Roy	Movie Recommendation System Using Sentiment Analysis From Microblogging Data	A hybrid recommendation system combining content-based filtering (CBF), collaborative filtering (CF), and sentiment analysis of tweets using the VADER sentiment analysis tool.
Shigang Hu, Akshi Kumar, Fadi Al-Turjman, Shivam Gupta, Simran Seth, Shubham	Reviewer Credibility and Sentiment Analysis Based User Profile Modelling for Online Product Recommendation	Developed CISER model using fastText for fine-grained sentiment scoring and heuristic-driven user preference mining. Assigned credibility scores to reviewers based on trust, expertise, and influence.

Table 2.2: Literature review summary table of Sentiment-based Recommendations

Existing studies have explored various approaches to recommendation systems, such as leveraging sentiment analysis, contextual modelling, clustering user preferences, and integrating credibility scoring for personalized suggestions. These methods primarily address issues like data sparsity, domain insensitivity and tailoring recommendations based on specific user traits or preferences. However, my project focuses on analysing social media captions to select songs based on the matched song genres that match the emotional and contextual nuances of the text, providing a unique application and approach within the area of sentiment-based recommendation systems.

2.3 Chapter Summary and Evaluation

In this chapter, the key concepts and theories of recommendation systems and sentiment analysis are discussed to have a further understanding on the research topic. Moreover, some papers related to music recommendation systems and sentiment-based recommendations are studied and discussed to gain insights into various applications of music recommendation systems and recommendation systems that utilized sentiment analysis.

Chapter 3

Methodology and Requirements Analysis

3 Methodology and Requirements Analysis

The methodologies and requirements needed for developing the recommendation system are outlined in this chapter.

3.1 Requirement Analysis

3.1.1 System Architecture

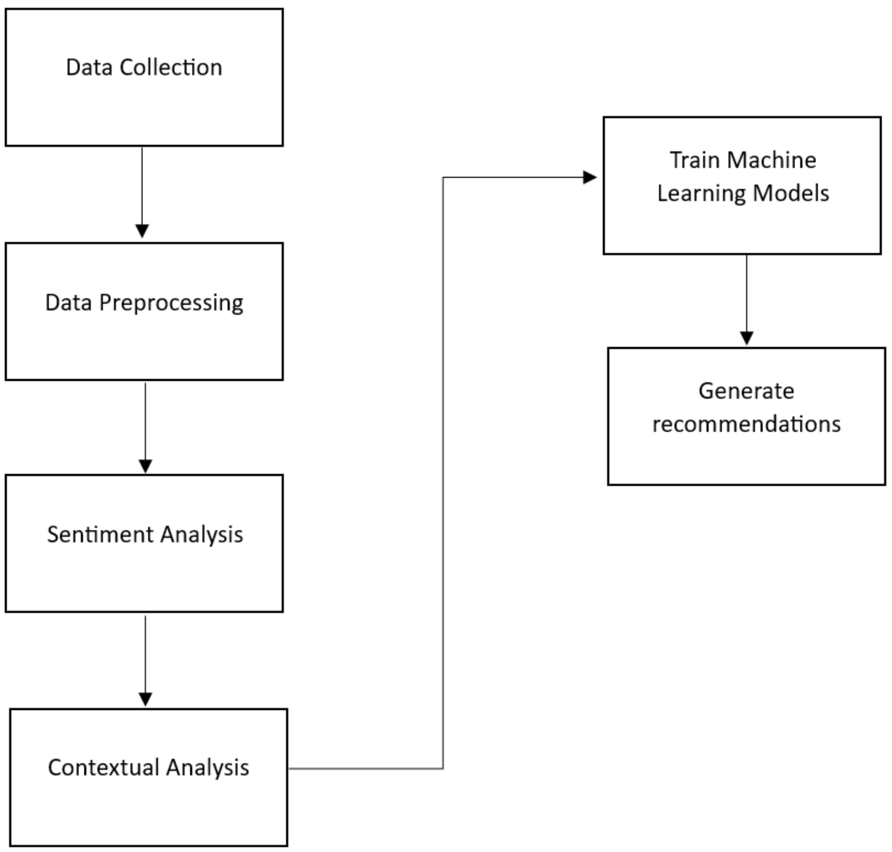

Figure 1: System Architecture of Song Recommendation System Based on Caption

The system architecture begins with data collection, where user-generated social media captions serve as input through the system interface. Next, the text data is then passed to the data preprocessing stage, where it undergoes cleaning and standardization. This involves removing stop words, special characters, emojis, and punctuations to ensure the text is free of noise and ready for analysis. Once the text is pre-processed, it proceeds to sentiment analysis, where the emotional tone of the captions is analysed and classified into categories such as happy, sad, angry, fear, excited and more. Techniques like VADER and TF-IDF are applied. Following this, the contextual analysis phase applies topic modelling methods, such as Latent Dirichlet Allocation (LDA), to uncover themes or topics from the captions, providing deeper insights into the contextual meaning behind the user input. By combining the results from sentiment analysis and contextual analysis, the system compares the processed captions with the vectorized song features. To recommend song genres that align with both the emotional tone and context of the captions, machine learning models such as Logistic Regression, Random Forest, and Neural Networks will be trained and utilized.

3.1.2 Data Collection

The data collection process involves gathering information from two primary sources: song data, which was initially intended to be collected using the Spotify API but is now sourced from a ready dataset on Kaggle due to Spotify API restrictions on scraping audio features (Spotify, 2024), and the social media captions, which are also obtained from a sentiment analysis dataset on Kaggle since direct scraping from platforms like Instagram may have restrictions. The song data included features like genre, acousticness, energy, and artists, providing a rich set of attributes necessary for matching songs with social media captions. The user-generated captions are to serve as input text for the sentiment analysis and contextual analysis processes. These captions reflect users' expressions, moods, and themes, making them an integral part of the recommendation system. The dataset includes a variety of caption samples, ensuring the text input is diverse and representative of actual social media usage.

[4]:

	artist	song	duration_ms	explicit	year	popularity	danceability	energy	key	loudness	mode	speechiness	acousticness	instrumentalness	liveness	valence
0	Britney Spears	Oops!...I Did It Again	211160	False	2000	77	0.751	0.834	1	-5.444	0	0.0437	0.3000	0.000018	0.3550	0.894
1	blink-182	All The Small Things	167066	False	1999	79	0.434	0.897	0	-4.918	1	0.0488	0.0103	0.000000	0.6120	0.684
2	Faith Hill	Breathe	250546	False	1999	66	0.529	0.496	7	-9.007	1	0.0290	0.1730	0.000000	0.2510	0.278
3	Bon Jovi	It's My Life	224493	False	2000	78	0.551	0.913	0	-4.063	0	0.0466	0.0263	0.000013	0.3470	0.544
4	*NSYNC	Bye Bye Bye	200560	False	2000	65	0.614	0.928	8	-4.806	0	0.0516	0.0408	0.001040	0.0845	0.879

Figure 2: Overview of Song Dataset

[6]:

	Unnamed: 0	text	label
0	0	i just feel really helpless and heavy hearted	4
1	1	ive enjoyed being able to slouch about relax a...	0
2	2	i gave up my internship with the dmrg and am f...	4
3	3	i dont know i feel so lost	0
4	4	i am a kindergarten teacher and i am thoroughl...	4

Figure 3: Overview of Captions Dataset

3.1.3 Data Preprocessing

1. Tokenization

   This step involves breaking down the text into individual words (tokens) to analyse them separately using spaces and punctuations as delimiters. For example, the original input text “I am happy” will be [‘I’, ‘am’, ‘happy’] after it is tokenized.

2. Stopword Removal

   This process is to remove common words such as ‘of’, ‘the’, ‘and’ that do not carry significant meaning using predefined stopword lists like NLTK or SpaCy.

3. Lemmatization

   Lemmatization is to reduce words to their root form to ensure that similar words are treated the same. For example, the word ‘played’ will become ‘play’ after it is lemmatized.

4. Normalization

   This step includes converting text to lowercase, removing numbers, removing white space, and removing punctuations to ensure the consistency and uniformity.

3.1.4 Sentiment Analysis

1. VADER (Valence Aware Dictionary and sEntiment Reasoner)

   VADER is a rule-based sentiment analysis tool with lexicon. It combines a sentiment lexicon with a list of lexical properties that are often classified as either positive or negative based on their semantic orientation. VADER is particularly well-suited for analysing social media texts due to its ability to handle informal language, such as abbreviations, emoticons, slang, and punctuation. For example, it can recognize expressions like “amazing!!!” or “soooo happy” as having strong positive sentiment due to the use of exclamation marks, repeated letters, or elongated words. VADER is ideal for social media texts, as it not only indicates the positivity and negativity score, but also the degree of positivity or negativity of a sentiment. VADER examines a text to determine whether any of the words are found in the VADER lexicon. It can use the polarity_scores() function to determine the polarity indices. The metric values of the negative, positive, and compound for a given statement will be returned as a result. The compound score is a normalized value between -1 (most negative) to +1 (most positive) (Bonta et al., 2019). The typical threshold values are listed as following:

   Positive sentiment: Compound score is 0.05 or higher

   Neutral sentiment: Compound score is greater than -0.05 but less than 0.05

Negative sentiment: Compound score is -0.05 or lower

2. TF-IDF (Term Frequency-Inverse Document Frequency)

TF-IDF is a well-known technique for determining a word's significance in a document. It is used to identify important keywords in captions by measuring the relevance of a word in a document relative to its frequency across a set of documents. TF-IDF algorithm gives a word a weight according to how often it appears in a document (Mishra et al., 2019). Words that frequently occur in a document but are less common across other documents are assigned higher weights, indicating their importance.

The TF-IDF algorithm works as follows:

Term Frequency (TF) measures how often a word (t) appears in a document:

$$TF\ (t) = \frac{(\text{Number of times term } t \text{ appears in a document})}{(\text{Total number of terms in the document})}$$

Inverse Document Frequency (IDF) measures the importance of a word:

$$IDF\ (t) = \frac{(\text{Total number of document})}{(\text{Number of documents containing the term } t)}$$

TF-IDF Score is obtained by the formula:

$$TF - IDF = \mathrm{TF}(t) * \mathrm{IDF}(t)$$

TF-IDF can be applied to captions to identify key words or phrases that reflect the main themes or emotions. For instance, in a caption like "Relaxing evening with a beautiful sunset", the algorithm might assign higher weights to words like "relaxing" or "sunset" while ignoring common words like "with" or "a".

3.1.5 Contextual Analysis

Topic Modelling

Topic modelling is a technique in information retrieval that automatically uncovers and organizes the underlying themes in a collection of text documents. It automatically groups words that frequently co-occur, uncovering latent structures in the data without requiring prior labels or classifications. By condensing large volumes of text into interpretable topics, it provides a systematic way to analyse unstructured data, making it an essential tool in many natural language processing (NLP) applications (Abdelrazek et al., 2023). Hence, topic modelling is used to extract themes from social media captions, helping to understand the context beyond simple sentiment analysis. Social media captions often convey complex ideas, emotions, or situations, such as "celebrations," "nature," or "relationships." Therefore, topic modelling, specifically techniques like Latent Dirichlet Allocation (LDA) organizes captions into interpretable topics based on word usage. For example, captions containing “beach,” “sunset,” and “ocean” may be categorized under a “travel” topic, whereas captions using terms like “party,” “fun,” and “friends” may belong to a “celebration” topic.

1. Latent Dirichlet Allocation (LDA)

   LDA is a commonly used technique for topic modelling. LDA is useful for summarizing, clustering, linking, or processing large datasets, as it produces a list of topics with associated weights for each document. The Dirichlet distribution is used to generate the per-document topic distributions, which are then employed in the LDA generative process to assign words within a document to various topics. In LDA, documents are observable entities, while topics, per-document topic distributions, and the topic assignments of individual words within documents are hidden structures. LDA processes individual documents along with several parameters to generate a model comprising weights that can be normalized into probabilities. These probabilities are of two types: (a) the probability of a specific document generating a particular topic and (b) the probability of a specific topic generating certain words from the vocabulary. Documents labelled with a list of topics (type a probabilities) are often further processed to generate type b probabilities, which correspond to the occurrence of specific words. In simple terms, the LDA algorithm relies on the distribution of words within a document, identifying whether words belong to the same topic while accounting for multiple topics within the document (Negara et al., 2019). Therefore, LDA aims to assign each caption a topic probability distribution, indicating the likelihood of the caption belonging to each topic. For example, a caption like “Amazing sunset by the beach” might be strongly associated with the topic “nature” with an 80% probability. These identified topics can then be associated with relevant song features, such as genres or moods, to improve the contextual feature vector used to match captions with songs.

3.1.6 Modelling

Logistic Regression, Random Forest, and Neural Network (MLPClassifier) were trained and evaluated to identify the most suitable technique for genre prediction. Each model was evaluated using metrics like Precision, Recall, F1-score, and Hamming loss, allowing for a comprehensive comparison of their performance.

Metrics	Description	Formula
Hamming Loss	Metric used in multi-label classification to evaluate the	$\frac{1}{nL}\sum_{i=1}^{n}\sum_{j=1}^{L}\left[I(y_j^{(i)} \neq \hat{y}_j^{(i)}\right]$

	fraction of labels that are incorrectly predicted	
Precision	Measures how many of the predicted positive labels are actually correct	$\frac{TP}{TP + FP}$
Recall	Measures how many of the actual positive labels were correctly predicted	$\frac{TP}{TP + FN}$
F1-Score	The harmonic mean of Precision and Recall	$2 * \frac{Precision * Recall}{Precision + Recall}$

Table 3.1: Metrics Used in Model Evaluation

3.2 Non-functional Requirements

Requirement	**Description**	**How It Will Be Achieved**
1. Performance	The system should provide song recommendations in real-time to enhance user experience.	The system will be optimized for speed by using pre-trained models for sentiment analysis and efficient algorithms to match captions with songs. The goal is to return results within seconds per query. Testing and optimization will ensure low latency.
2. Usability	The system should be user-friendly and easy to use.	The system will have a simple interface integrated in Jupyter Notebook which allows user to input caption for recommendation.
3. Reliability	The system should provide accurate recommendations consistently without failures.	The system response speed and accuracy of recommendations will be checked and monitored regularly.

3.3 System Development Environment

3.3.1 Hardware Environment

The hardware specifications that I used for this project are listed as following:

- AMD Ryzen 7 5800H with Radeon Graphics CPU @ 3.2 GHz
- 16GB RAM
- NVIDIA GeForce GTX 1650 GPU

These chosen hardware specifications are well-suited for this project due to their capability to handle the computational demands of Natural Language Processing, Sentiment Analysis and Machine Learning algorithms required for this song recommendation system. The processor with multiple cores and threads is essential for processing large datasets, which significantly reduces the time needed for training and testing machine learning models. With 16GB of RAM, the project can handle data-intensive tasks like text preprocessing, model training and real-time song recommendation. Although the GTX 1650 is a mid-range GPU, it provides sufficient computational power for accelerating deep learning tasks.

3.3.2 Software Environment

The software required for this project are listed as following:

1. Anaconda

   Anaconda is a free software toolkit designed for research and science, offering integrated development environments (IDEs) for coding in Python or R language. It provides access to various environments that simplify code development and management. It includes a wide range of libraries and pre-built functions, which is easy for me to access essential tools for data analysis and machine learning (Rolon-Mérette et al., 2020).

2. Jupyter Notebook

   Jupyter Notebook is a popular platform for interactive literate programming, designed to simplify the documentation, sharing, and reproducibility of data analysis (Pimentel et al., 2019). It is a web-based IDE that runs in our browser, allowing code blocks to be executed separately for flexibility. It supports combining code, visualizations, equations, and text in a single document, making it ideal for creating organized, shareable, and visually appealing reports (Rolon-Mérette et al., 2020).

3.4 Chapter Summary and Evaluation

In this chapter, requirements like system architecture, methods used for data collection, techniques used for data preprocessing and algorithms used in this system are stated clearly. Non-functional requirements are stated to justify how they will be achieved in this system. Besides, the sources from which the data are obtained are detailed and the software and hardware requirements have also been outlined to ensure the system's optimal functionality.

Chapter 4

System Design

4 System Design

This chapter provides an overview of the user interface developed for the Caption-Based Song Recommender System. How users interact with the system to receive personalized song suggestions based on their input captions is also explained.

4.1 Song Recommendation System

4.1.1 PySimpleGUI

PySimpleGUI is a Python library that simplifies the process of creating graphical user interfaces (GUIs). I have selected PySimpleGUI to deploy a user interface for this system as it offers a simple yet effective way to create user-friendly graphical interfaces with minimal code. It allows rapid development and easy customization, making it ideal for deploying machine learning models in a lightweight and interactive desktop environment. Since this system only requires basic user interactions such as entering a caption, selecting a model, and displaying song recommendations, PySimpleGUI provides all the necessary components without the complexity of more advanced GUI frameworks.

4.1.2 User Interface Design

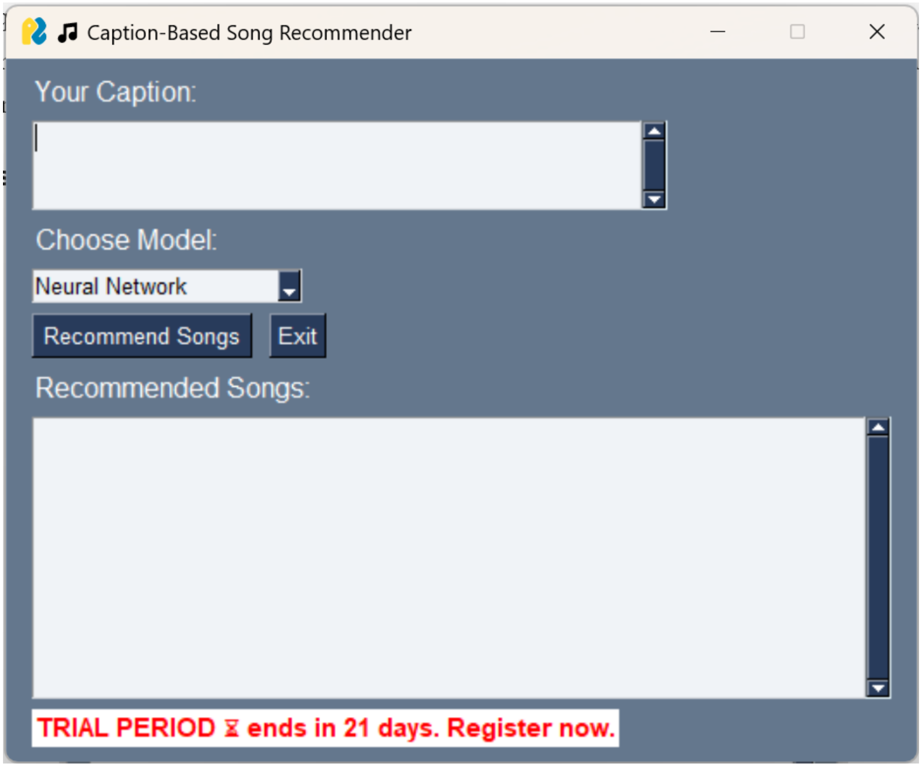

Figure 4: Screenshot of UI (1)

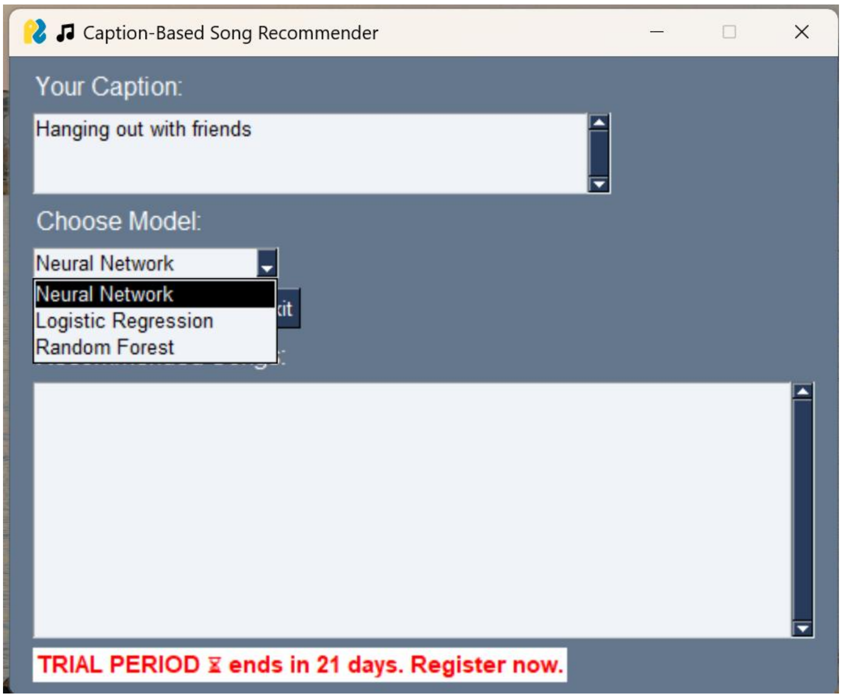

Figure 5: Screenshot of UI (2)

The user interface of the song recommendation system is designed using PySimpleGUI. User is required to input the caption in the first dialog box and then select any one of the models from the dropdown list for genre prediction as shown in Figure 5.

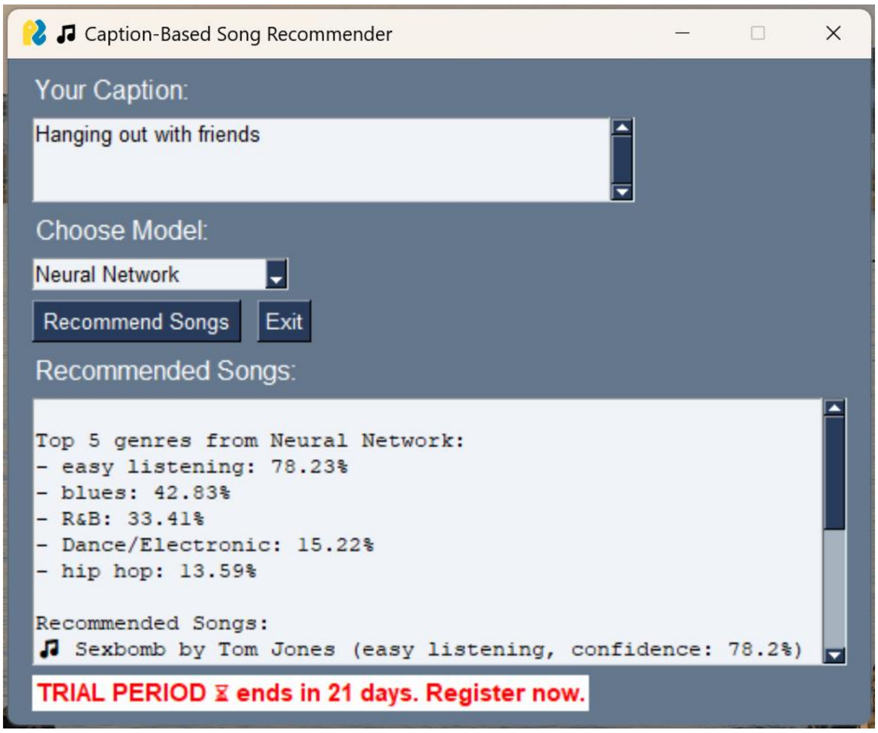

Figure 6: Screenshot of UI (3)

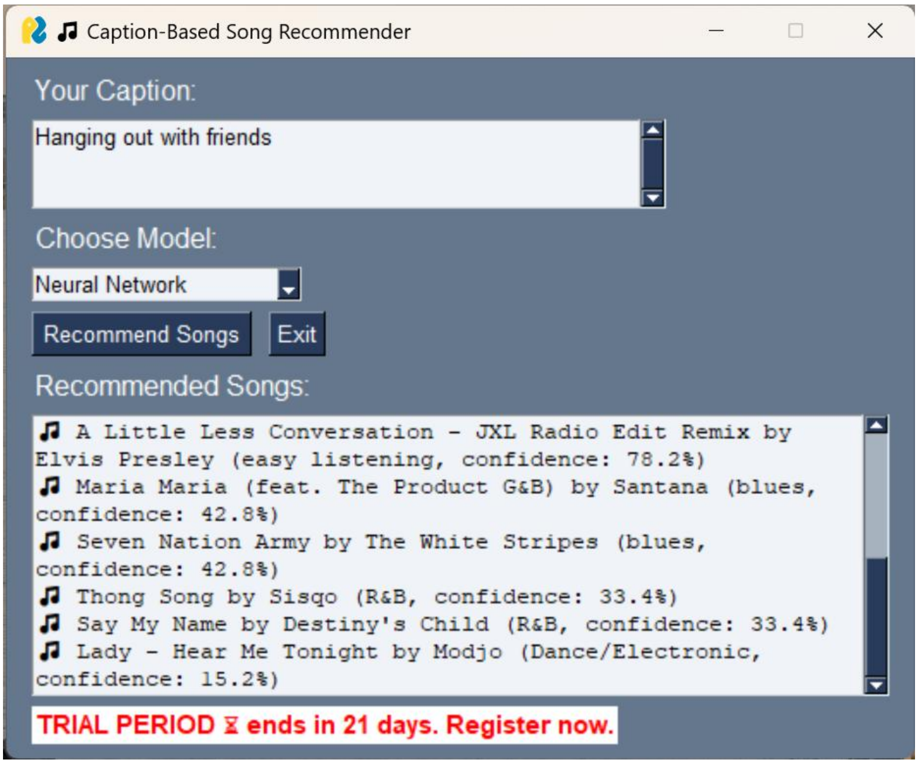

Figure 7: Screenshot of UI (4)

The recommended genres and songs will be displayed after user clicked the “Recommend Songs” button. It will show the top 5 genres that are relevant and their respective probabilities. When user clicks “Exit” button, the user interface window will be closed.

4.2 Chapter Summary and Evaluation

In this chapter, the design and implementation of the user interface for the song recommender system using PySimpleGUI is presented and demonstrated. Overall, the UI enhances the user experience by offering an accessible and interactive way to engage with the recommender system.

Chapter 5

Implementation and Testing

5 Implementation and Testing

5.1 Implementation

5.1.1 Text Preprocessing

As mentioned in 3.1.3 Data Preprocessing, text preprocessing includes tokenization, stop word removal and lemmatization. Before these steps, there are several normalization steps to be carried out such as dropping unwanted columns, converting texts to lowercase, handling emojis, handling English shortforms, removing numbers, white space and remove punctuations. First, from the earlier Figure 3 that shows a snippet of the caption's dataframe, there are originally 3 columns 'Unnamed: 0', 'text' and 'label'. The 'Unnamed: 0' is the index label, while the 'label' is the originated emotion label of the text. Since we planned to classify the sentiment into positive, negative and neutral categories, the original 'label' column is dropped for further manual sentiment analysis.

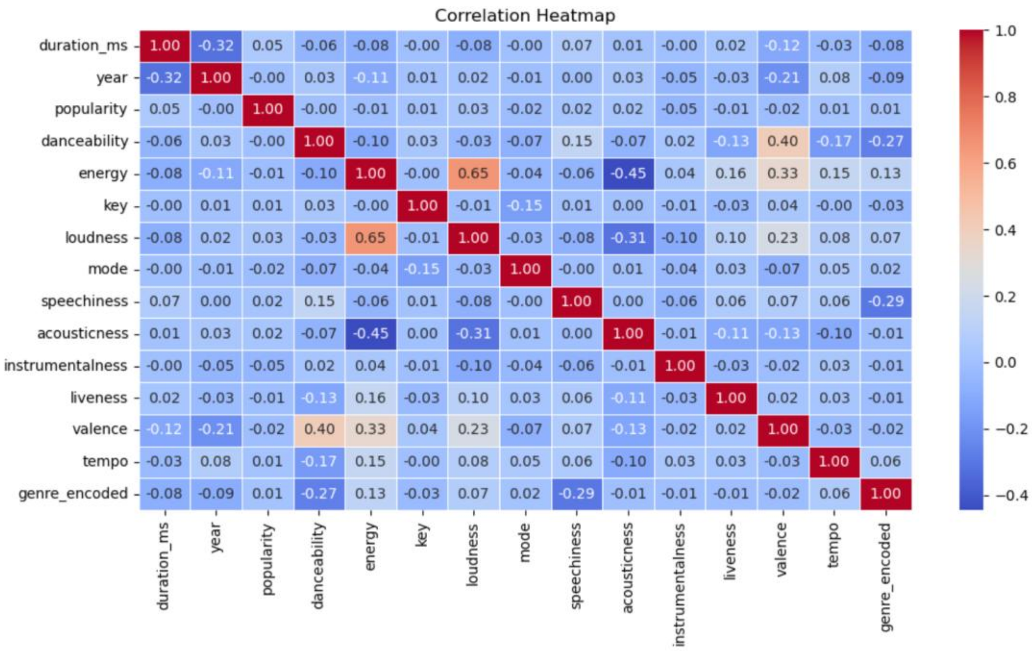

Figure 8: Correlation Heatmap of Audio Features

The correlation heatmap from Figure 8 visually represents the linear relationships between different audio features in the song dataset. The heatmap highlights key relationships between musical features. Notably, 'energy' and 'loudness' show a strong positive link: energetic songs tend to be louder. 'Danceability' and 'valence' also correlate positively, suggesting danceable tracks often have a more positive feel. Conversely, 'energy' and 'loudness' are negatively associated with 'acousticness', meaning energetic and loud songs are typically less acoustic. 'Speechiness' also shows a negative trend with 'acousticness'.

Beyond these strong connections, the heatmap displays a range of weaker correlations, both positive and negative, which could be explored further. Many other feature pairs exhibit minimal correlation, indicating they are largely independent of each other within this dataset. This overview of feature relationships is valuable for understanding the underlying structure of the music data.

Next, the text data is converted into lowercase. Lowercasing the text ensures consistency by treating words identically regardless of their capitalization. This is to reducing the vocabulary size and simplifying analysis. Emojis, while expressive, can introduce noise and complexity for many NLP models. Replacing them with their textual descriptions using the demoji library provides a way to retain the semantic information conveyed by the emoji in a text format that is more readily processed. For example, a smiley face emoji might be replaced with "[grinning face]," allowing the sentiment to be considered without the need for specialized emoji handling. Then, English short forms are expanded using the contractions library addresses the issue of variations in writing style. Contractions like “don't” are expanded to “do not,” and “it's” becomes “it is.” This standardization ensures that these semantically equivalent phrases are represented uniformly, which can improve the accuracy of downstream analyses.

Removing numbers is often beneficial when the numerical value itself is not the primary focus of the text analysis. For tasks like sentiment analysis or topic modelling, the presence of numbers might not contribute meaningful information and could even be distracting to the model. Therefore, numbers in texts are removed in this case. The whitespace removal step eliminates unnecessary spaces, tabs, and newlines from the text. This step helps in standardizing the text format and can prevent issues arising from inconsistent spacing, which might be interpreted as different tokens by some NLP tools. Similarly, punctuations are removed to help to focus on the essential words in the text as punctuation marks often serve grammatical purposes but might not carry significant semantic meaning for certain analyses. The results of all these steps are saved into a single, new column in the dataframe as shown in Figure 9 below.

[32]:

	text	Lowercased_Text	rm_emoji	english_shortform	remove_num	white_space_rm	remove_punctuation
36130	id say maybe made them feel foolish but that w...	id say maybe made them feel foolish but that w...	id say maybe made them feel foolish but that w...	id say maybe made them feel foolish but that w...	id say maybe made them feel foolish but that w...	id say maybe made them feel foolish but that w...	id say maybe made them feel foolish but that w...
138065	i joined the lds church i admit to feeling som...	i joined the lds church i admit to feeling som...	i joined the lds church i admit to feeling som...	eye joined the lds church eye admit to feeling...	eye joined the lds church eye admit to feeling...	eye joined the lds church eye admit to feeling...	eye joined the lds church eye admit to feeling...
146440	i must admit i didnt feel like hugging him not...	i must admit i didnt feel like hugging him not...	i must admit i didnt feel like hugging him not...	eye must admit eye didnt feel like hugging him...	eye must admit eye didnt feel like hugging him...	eye must admit eye didnt feel like hugging him...	eye must admit eye didnt feel like hugging him...
103337	i hate that i can still feel if any nerve is d...	i hate that i can still feel if any nerve is d...	i hate that i can still feel if any nerve is d...	eye hate that eye can still feel if any nerve ...	eye hate that eye can still feel if any nerve ...	eye hate that eye can still feel if any nerve ...	eye hate that eye can still feel if any nerve ...
315528	im actually feeling a little smug	im actually feeling a little smug	im actually feeling a little smug	im actually feeling a little smug	im actually feeling a little smug	im actually feeling a little smug	im actually feeling a little smug

Figure 9: Columns in df2 after normalization steps

The next step will be tokenization, which breaks down the text string into smaller units called ‘tokens’. This process is crucial because it provides structure to raw text, which is initially just a string of characters, and it enables the creation of features for machine learning models. From Figure 10, each sentence from the previous column has been split into individual words.

[35]:

	text	Lowercased_Text	rm_emoji	english_shortform	remove_num	white_space_rm	remove_punctuation	tokenized_text
36130	id say maybe made them feel foolish but that w...	id say maybe made them feel foolish but that w...	id say maybe made them feel foolish but that w...	id say maybe made them feel foolish but that w...	id say maybe made them feel foolish but that w...	id say maybe made them feel foolish but that w...	id say maybe made them feel foolish but that w...	[id, say, maybe, made, them, feel, foolish, bu...
138065	i joined the lds church i admit to feeling som...	i joined the lds church i admit to feeling som...	i joined the lds church i admit to feeling som...	eye joined the lds church eye admit to feeling...	eye joined the lds church eye admit to feeling...	eye joined the lds church eye admit to feeling...	eye joined the lds church eye admit to feeling...	[eye, joined, the, lds, church, eye, admit, to...
146440	i must admit i didnt feel like hugging him not...	i must admit i didnt feel like hugging him not...	i must admit i didnt feel like hugging him not...	eye must admit eye didnt feel like hugging him...	eye must admit eye didnt feel like hugging him...	eye must admit eye didnt feel like hugging him...	eye must admit eye didnt feel like hugging him...	[eye, must, admit, eye, didnt, feel, like, hug...
103337	i hate that i can still feel if any nerve is d...	i hate that i can still feel if any nerve is d...	i hate that i can still feel if any nerve is d...	eye hate that eye can still feel if any nerve ...	eye hate that eye can still feel if any nerve ...	eye hate that eye can still feel if any nerve ...	eye hate that eye can still feel if any nerve ...	[eye, hate, that, eye, can, still, feel, if, a...
315528	im actually feeling a little smug	im actually feeling a little smug	im actually feeling a little smug	im actually feeling a little smug	im actually feeling a little smug	im actually feeling a little smug	im actually feeling a little smug	[im, actually, feeling, a, little, smug]

Figure 10: Columns in df2 after tokenization

After the tokenization process, stop words in the sentences will be removed. Stop words are commonly occurring words in a language that often do not carry significant meaning or contribute much to the overall understanding of a text, especially in the context of tasks like text classification or information retrieval. By removing stop words, we can focus on the words that are more informative and carry more semantic weight. This can help the model to better identify the key features in the text that are relevant to the task at hand. Therefore, we can see

in the first row from Figure 11, the common words “them” and “but” are removed as shown in after_rm_stopwords column as compared to tokenized_text column.

[45]:

_Text	rm_emoji	english_shortform	remove_num	white_space_rm	remove_punctuation	tokenized_text	rm_eng_stopwords	after_rm_stopwords	lemmatized_english_text
aybe n feel t that w...	id say maybe made them feel foolish but that w...	id say maybe made them feel foolish but that w...	id say maybe made them feel foolish but that w...	id say maybe made them feel foolish but that w...	id say maybe made them feel foolish but that w...	[id, say, maybe, made, them, feel, foolish, bu...	[id, say, maybe, made, feel, foolish, would, r...	id say maybe made feel foolish would reeeeeeal...	id say maybe make feel foolish would reeeeeeal...
e lds nit to som...	i joined the lds church i admit to feeling som...	eye joined the lds church eye admit to feeling...	eye joined the lds church eye admit to feeling...	eye joined the lds church eye admit to feeling...	eye joined the lds church eye admit to feeling...	[eye, joined, the, lds, church, eye, admit, to...	[eye, joined, lds, church, eye, admit, feeling...	eye joined lds church eye admit feeling somewh...	eye join lds church eye admit feel somewhat as...
dmit i el like g him not...	i must admit i didnt feel like hugging him not...	eye must admit eye didnt feel like hugging him...	eye must admit eye didnt feel like hugging him...	eye must admit eye didnt feel like hugging him...	eye must admit eye didnt feel like hugging him...	[eye, must, admit, eye, didnt, feel, like, hug...	[eye, must, admit, eye, didnt, feel, like, hug...	eye must admit eye didnt feel like hugging eye...	eye must admit eye didnt feel like hug eye ang...
i can if any is d...	i hate that i can still feel if any nerve is d...	eye hate that eye can still feel if any nerve ...	eye hate that eye can still feel if any nerve ...	eye hate that eye can still feel if any nerve ...	eye hate that eye can still feel if any nerve ...	[eye, hate, that, eye, can, still, feel, if, a...	[eye, hate, eye, still, feel, nerve, damaged, ...	eye hate eye still feel nerve damaged badly en...	eye hate eye still feel nerve damage badly eno...
tually a little smug	im actually feeling a little smug	im actually feeling a little smug	im actually feeling a little smug	im actually feeling a little smug	im actually feeling a little smug	[im, actually, feeling, a, little, smug]	[im, actually, feeling, little, smug]	im actually feeling little smug	im actually feel little smug

Figure 11: Columns in df2 after stop word removal and lemmatization

The lemmatized_english_text column also shows that words have been reduced to their base or root form, known as the lemma. For instance, in the second row, “joined” is lemmatized to “join”, and “feeling” becomes “feel”. This process is called lemmatization, and it is applied to standardize words with similar meanings but different forms. Unlike stemming, which might simply remove suffixes, lemmatization considers the word's context and part of speech to arrive at the correct base form. This is important because it reduces redundancy in the data, grouping together inflected forms of words and therefore improving the accuracy and efficiency of downstream text analysis tasks.

5.1.2 Sentiment Analysis and Contextual Analysis

After the text are pre-processed and cleaned well, VADER is applied to the text to obtain the sentiment scores for the text. The scores are compound, positive, negative and neutral. The function takes the compound score as input and classifies the sentiment as positive, negative and neutral by following the threshold mentioned earlier in 3.1.4 VADER. The obtained sentiment labels are then stored in a new column called sentiment_label. Finally, the code displays the first few rows of the dataframe, showing the lemmatized_english_text along with the calculated compound, positive, neutral, and negative sentiment scores, and the final sentiment_label. For example, the first row's text, “id say maybe make feel foolish would reeeeeeal ...”, has a compound score of -0.2500, and is classified as “Negative”.

[48]:

_punctuation	tokenized_text	rm_eng_stopwords	after_rm_stopwords	lemmatized_english_text	sentiment_scores	compound	positive	neutral	negative	sentiment_label
maybe made el foolish but that w...	[id, say, maybe, made, them, feel, foolish, bu...	[id, say, maybe, made, feel, foolish, would, r...	id say maybe made feel foolish would reeeeeeal...	id say maybe make feel foolish would reeeeeeal...	{'neg': 0.206, 'neu': 0.686, 'pos': 0.108, 'co...	-0.2500	0.108	0.686	0.206	Negative
joined the lds eye admit to feeling...	[eye, joined, the, lds, church, eye, admit, to...	[eye, joined, lds, church, eye, admit, feeling...	eye joined lds church eye admit feeling somewh...	eye join lds church eye admit feel somewhat as...	{'neg': 0.132, 'neu': 0.519, 'pos': 0.349, 'co...	0.5563	0.349	0.519	0.132	Positive
ust admit eye didnt feel like iugging him...	[eye, must, admit, eye, didnt, feel, like, hug...	[eye, must, admit, eye, didnt, feel, like, hug...	eye must admit eye didnt feel like hugging eye...	eye must admit eye didnt feel like hug eye ang...	{'neg': 0.395, 'neu': 0.432, 'pos': 0.173, 'co...	-0.7693	0.173	0.432	0.395	Negative
that eye can f any nerve ...	[eye, hate, that, eye, can, still, feel, if, a...	[eye, hate, eye, still, feel, nerve, damaged, ...	eye hate eye still feel nerve damaged badly en...	eye hate eye still feel nerve damage badly eno...	{'neg': 0.589, 'neu': 0.411, 'pos': 0.0, 'comp...	-0.9559	0.000	0.411	0.589	Negative
ually feeling a little smug	[im, actually, feeling, a, little, smug]	[im, actually, feeling, little, smug]	im actually feeling little smug	im actually feel little smug	{'neg': 0.0, 'neu': 0.726, 'pos': 0.274, 'comp...	0.1298	0.274	0.726	0.000	Positive

Figure 12: Columns in df2 after applying VADER

Next, topic modelling using Latent Dirichlet Allocation (LDA) is performed on the two different datasets: captions and songs. For the captions (stored in df2), the pre-processed text is tokenized from the “lemmatized_english_text” column, creating a list of words for each caption. These tokens are then used to build a dictionary and a bag-of-words (BoW) representation. An LDA model is trained on this BoW corpus to discover 5 underlying topics, and each caption is assigned its dominant topic. A similar process is applied to the song genres (stored in df1). The genre strings are split into lists of individual genres, creates a dictionary and BoW representation, trains a separate LDA model, and assigns a dominant topic to each song based on its genre list. In essence, it is aimed to extract thematic structures from the caption texts and genre classifications using LDA.

The LDA model identifies five topics in the captions, each represented by weighted words. It also identifies five topics in song genres, heavily dominated by specific genres. For example, Topic 0 is associated with "R&B" and "pop" while Topic 1 is associated with "hip hop" and "pop." After that, the sentiment of songs is classified based on the valence score of each song. Valence score which is equal or higher than 0.6 will be labelled as "positive", 0.4 or lower as "negative", and "neutral" otherwise.

5.1.3 Multi-label Binarization

The "genre" column in df1_filtered dataframe is converted from a comma-separated string to a list of individual genres. This is essential for handling songs that belong to multiple genres. A MultiLabelBinarizer is then used to one-hot encode these genre lists, transforming them into a binary matrix where each column represents a unique genre, and each row indicates the presence or absence of that genre for a given song. This encoded data is converted into a new dataframe, genre_df. The original "genre" and the newly created "genre_list" columns are dropped from df1_filtered, and the one-hot encoded genre dataframe (genre_df) is concatenated to it, creating df1_final. To prepare for merging, the sentiment label columns in df2_filtered and df1_final are renamed to "sentiment_label_caption" and "sentiment_label_song" respectively. Finally, the code performs an inner merge of df2_filtered and df1_final based on these cleaned sentiment label columns, resulting in a new dataframe df_matched that combines data from captions and songs with matching sentiment.

[67]:
```
# Initialize MultiLabelBinarizer
mlb = MultiLabelBinarizer()

# Transform genre column
genre_encoded = mlb.fit_transform(df1_filtered["genre_list"])

# Convert back to DataFrame
genre_df = pd.DataFrame(genre_encoded, columns=mlb.classes_)

print(genre_df.head())
```

```
   Dance/Electronic  Folk/Acoustic  R&B  World/Traditional  blues  classical  \
0                 0              0    0                  0      0          0
1                 0              0    0                  0      0          0
2                 0              0    0                  0      0          0
3                 0              0    0                  0      0          0
4                 0              0    0                  0      0          0

   country  easy listening  hip hop  jazz  latin  metal  pop  rock  set()
0        0               0        0     0      0      0    1     0      0
1        0               0        0     0      0      0    1     1      0
2        1               0        0     0      0      0    1     0      0
3        0               0        0     0      0      1    0     1      0
4        0               0        0     0      0      0    1     0      0
```

Figure 13: Code Snippet of applying Multi Label Binarizer

10	able afford new sofa leather one cost sek eye ...	0	neutral	Like a Boy	Ciara	0	neutral	0	0	1	...
11	able afford new sofa leather one cost sek eye ...	0	neutral	Viva La Vida	Coldplay	4	neutral	0	0	0	...
12	able afford new sofa leather one cost sek eye ...	0	neutral	Sorry	Justin Bieber	3	neutral	0	0	0	...
13	able afford new sofa leather one cost sek eye ...	0	neutral	Love Yourself	Justin Bieber	3	neutral	0	0	0	...
14	able afford new sofa leather one cost sek eye ...	0	neutral	Yo (Excuse Me Miss)	Chris Brown	0	neutral	0	0	1	...
15	able afford new sofa leather one cost sek eye ...	0	neutral	Impossible	Shontelle	0	neutral	0	0	1	...

Figure 14: Columns in df_matched

5.1.4 Logistic Regression Model Evaluation

```
# Apply train-test split correctly
X_train, X_test, y_train, y_test = train_test_split(
    df_matched["lemmatized_english_text"],
    y,  # Use the multi-label binarized labels
    test_size=0.2,
    random_state=42,
)

# TF-IDF Transformation
tfidf_vectorizer = TfidfVectorizer(
    max_features=1000,
    min_df=5,
    max_df=0.5,
    ngram_range=(1, 2),
    sublinear_tf=True,
    stop_words='english',
    dtype=np.float32
)
```

Figure 15: Code Snippet of train-test split and TF-IDF

The data used for training and testing is 80% and 20% respectively. TF-IDF vectorization is applied to the training set's captions to convert the text into a numerical representation, capturing word importance; the same vectorizer is then used to transform the test set's captions.

```
LogisticRegression - Training Hamming Loss: 0.1327
LogisticRegression - Test Hamming Loss: 0.1333
LogisticRegression - Precision Score: 0.8355
LogisticRegression - Recall Score: 0.4529
LogisticRegression - F1 Score: 0.5874
```

Figure 16: Results of Logistic Regression model

Both the training and test Hamming Losses are around 0.13, indicating that, on average, about 13% of the genre labels for a song are incorrectly predicted. The precision score is 0.8355, meaning that when the model predicts a genre, it is correct about 83.55% of the time. The recall score is 0.4529, indicating that the model only correctly identifies about 45.29% of the actual genres a song belongs to. While the F1 score is 0.5874, which suggests a moderate balance between precision and recall. This means that when the model predicts a genre, it is often correct, but it misses a significant portion of the actual genres associated with a song.

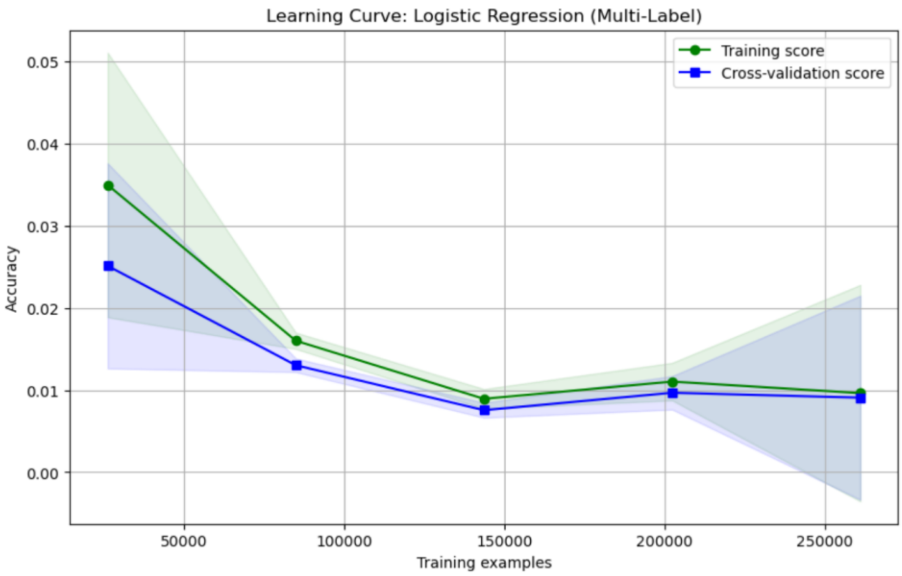

Figure 17: Learning Curve of Logistic Regression

The cross-validation score starts off significantly lower, indicating initial overfitting. However, as the training size increases, the gap between the training and cross-validation scores narrows, suggesting that the model begins to generalize better with more data. Despite this improvement, both the training and cross-validation scores remain quite low—hovering around 1% to 3%—highlighting potential issues. These low scores may be attributed to the complexity of the multi-label classification problem, data imbalance, or the simplicity of logistic regression, which might not be expressive enough for the task.

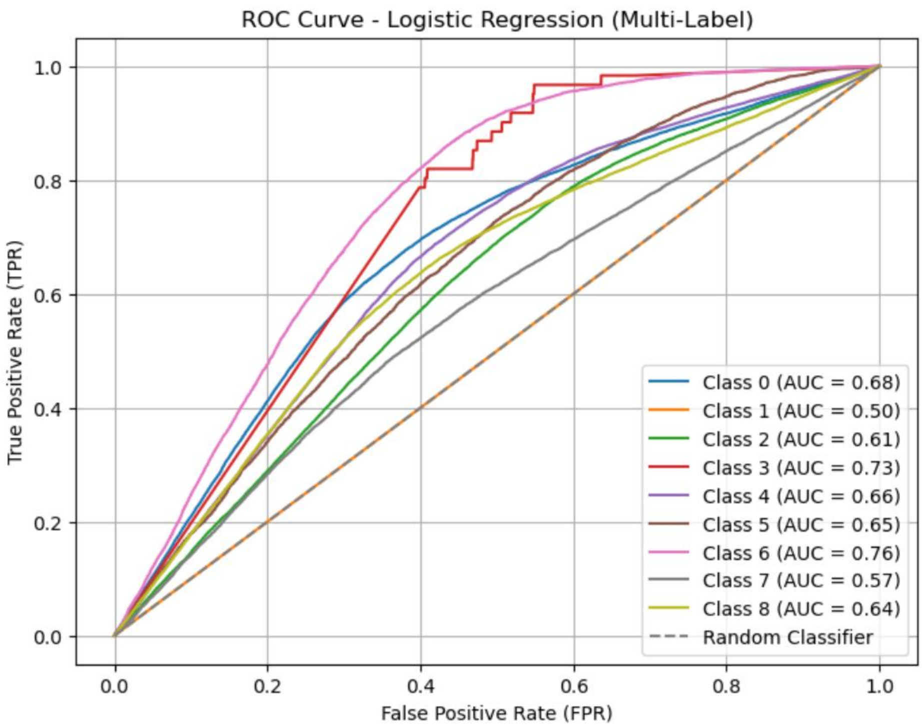

Figure 18: ROC Curve of Logistic Regression

From the plot, Class 6 has the highest AUC of 0.76, indicating relatively strong discriminative performance. Other well-performing classes include Class 3 (AUC = 0.73) and Class 0 (AUC = 0.68). On the other hand, Class 1 has an AUC of 0.50, suggesting the model performs no better than random for this class. Similarly, Class 7 (AUC = 0.57) and Class 2 (AUC = 0.61) reflect only modest ability to separate true positives from false positives. The ROC curves suggest that while Logistic Regression shows promise for some classes, its performance varies significantly across labels—likely due to class imbalance or differences in feature patterns for each class.

5.1.5 Random Forest Model Evaluation

```
RandomForest - Training Hamming Loss: 0.1310
RandomForest - Test Hamming Loss: 0.1497
RandomForest - Precision Score: 0.7261
RandomForest - Recall Score: 0.4586
RandomForest - F1 Score: 0.5622
```

Figure 19: Results of Random Forest model

Figure 14 shows a training Hamming Loss of 0.1310 and a test Hamming Loss of 0.1497, indicating a slight increase in error on unseen data. The model achieves a precision of 0.7261, meaning it is correct about 72.61% of the time when it predicts a genre. However, its recall is 0.4586, suggesting it only captures about 45.86% of the actual genres. The F1 score, balancing precision and recall, is 0.5622. Overall, the model demonstrates reasonable precision but lower recall, implying it is fairly accurate when it makes a prediction but misses a significant portion of the true genres.

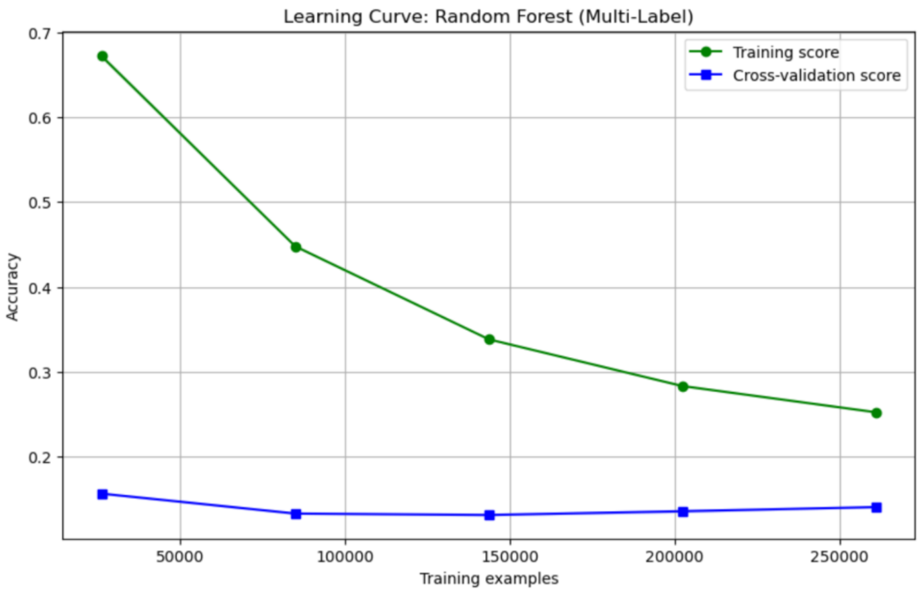

Figure 20: Learning Curve of Random Forest

The pattern shown in Figure 15 suggests a high bias problem, where the model is not complex enough to capture the underlying structure of the data. The flatness of the CV score, even as the training set grows, indicates that simply adding more data is not improving the model's ability to learn effectively. This could be due to several factors, such as suboptimal model complexity, lack of expressive features or unoptimized hyperparameters. Therefore, I have tried to adjust the hyperparameters for this model, but the results are remained the same. It may the data imbalance issue in the dataset.

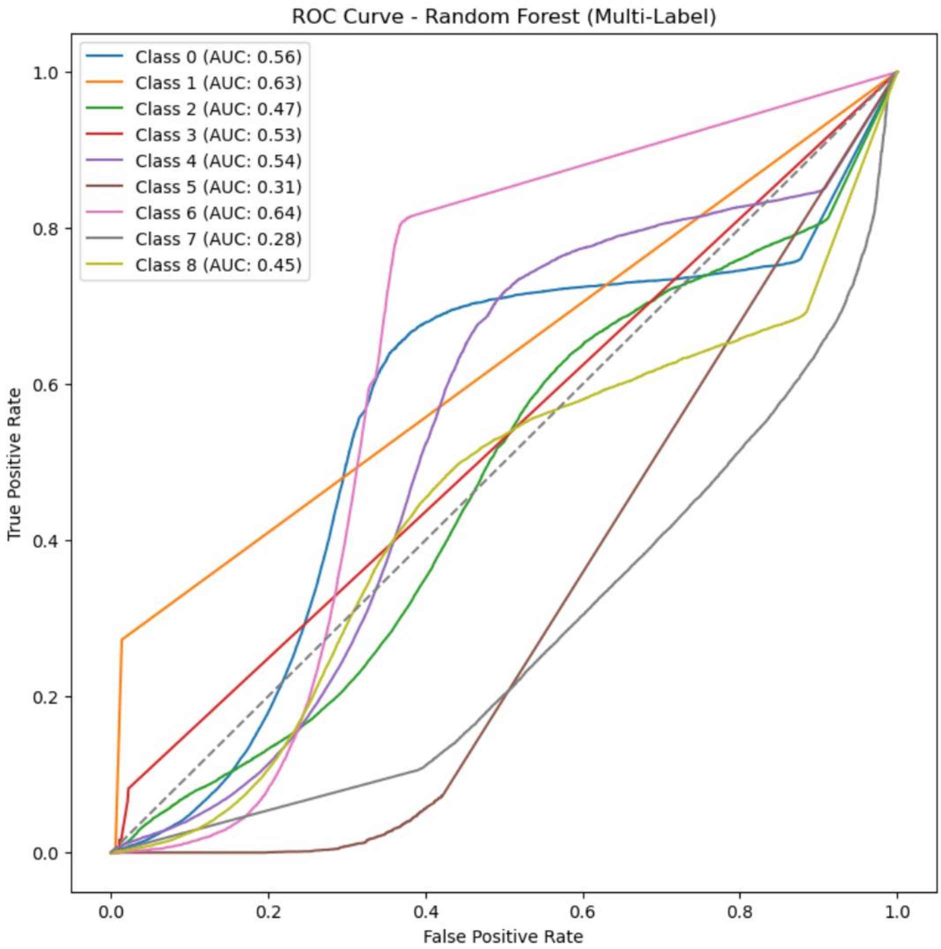

Figure 21: ROC Curve of Random Forest

This model demonstrates limited effectiveness across most classes, with the curves frequently approaching the diagonal line that represents random classification. The inconsistent performance across different classes suggests that the model struggles with certain categories, potentially due to class imbalance or inherent difficulty in distinguishing between certain genres in this multi-label context.

5.1.6 Neural Network Model Evaluation

```
Neural Network (MLP) - Training Hamming Loss: 0.1323
Neural Network (MLP) - Test Hamming Loss: 0.1327
Neural Network (MLP) - Precision Score: 0.8455
Neural Network (MLP) - Recall Score: 0.4484
Neural Network (MLP) - F1 Score: 0.5860
```

Figure 22: Results of Neural Network

With nearly identical Hamming loss values of 0.1323 and 0.1327 for training and test sets respectively, the model demonstrates consistent performance without overfitting, incorrectly predicting approximately 13% of all labels. The high precision score of 0.8455 indicates that when the model assigns a genre label, it is correct about 85% of the time, showing strong confidence in its positive predictions. However, the notably lower recall score of 0.4484 reveals a significant limitation—the model identifies less than half of the genres that should be assigned to songs, suggesting a conservative prediction approach. The resulting F1 score of 0.5860 reflects this imbalance between precision and recall, pointing to a model that prioritizes accuracy in its predictions at the expense of comprehensiveness.

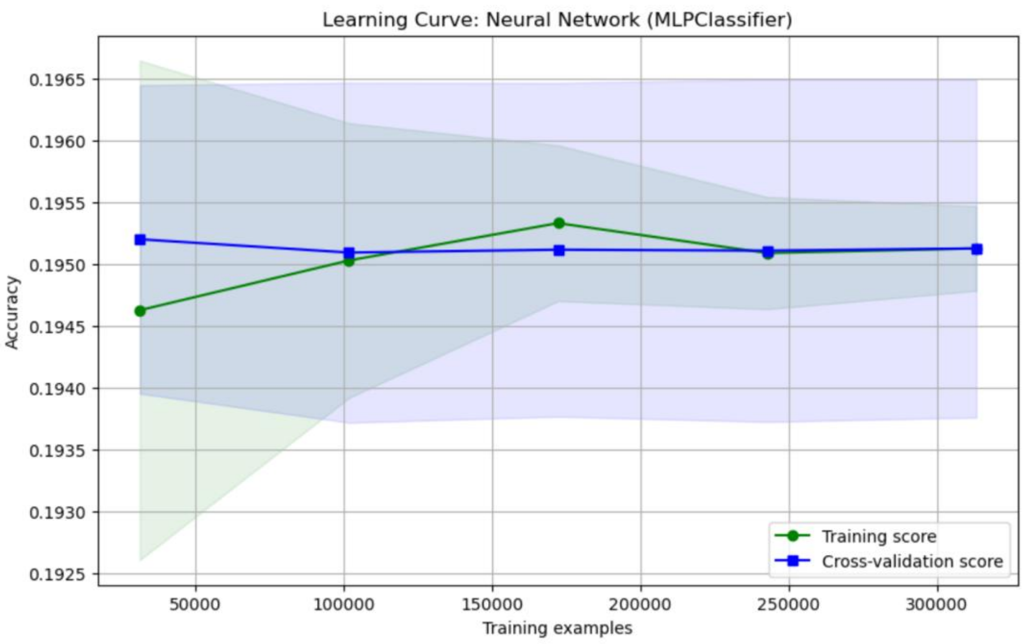

Figure 23: Learning Curve of Neural Network

Both the training score (green line) and cross-validation score (blue line) hover around 0.195 (19.5%) accuracy, indicating remarkably consistent performance across different training set sizes. The narrow-shaded regions represent confidence intervals, suggesting stable performance with minimal variance. Notably, the training and validation scores remain very close to each other throughout the entire range, with almost no gap between them, which strongly indicates that the model is not overfitting.

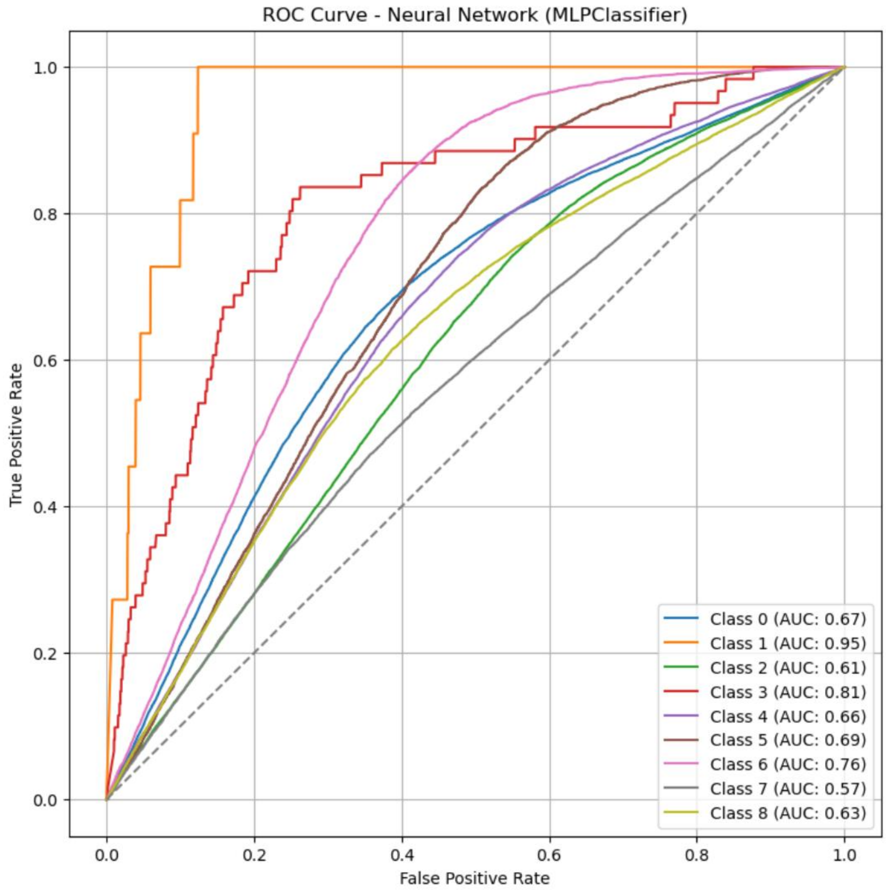

Figure 24: ROC Curve of Neural Network

This significant variation in AUC values indicates that the Neural Network model is much more effective at identifying certain music genres than others, likely reflecting either inherent distinguishing features in those genres that are more easily captured by the model, or potentially an imbalance in the training data that favours certain classes. Overall, the model performs substantially better than the Random Forest model discussed previously, particularly for Classes 1 and 3, suggesting that the neural network architecture is more suitable for this multi-label genre classification task.

5.1.7 Comparison of Model Performance

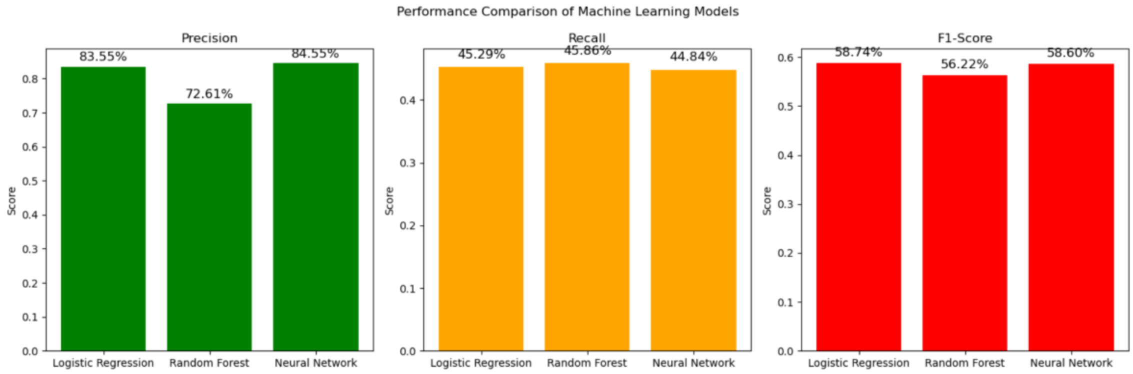

Figure 25: Performance Comparison of Models

From the Precision chart, it is observed that the Neural Network achieved the highest precision at 84.55%, slightly outperforming Logistic Regression at 83.55%, while Random Forest lagged with 72.61%. This indicates that Neural Network and Logistic Regression models made fewer false positive errors compared to Random Forest.

In terms of Recall, all three models performed similarly, with scores around 45% — Logistic Regression at 45.29%, Random Forest at 45.86%, and Neural Network at 44.84%. This shows that all models had a moderate ability to correctly identify the relevant genres, but none achieved particularly high recall.

Looking at the F1-Score, which balances precision and recall, both Logistic Regression and Neural Network scored similarly around 58.7%, whereas Random Forest was slightly lower at 56.22%. This reflects that Logistic Regression and Neural Network maintained a better balance between precision and recall compared to Random Forest.

The recall and F1 scores are moderate, around 45% and 58% respectively, primarily because of dataset imbalance and the nature of the multi-label classification task. In this project, some genres like "easy listening" had more samples compared to other genres with few samples. This imbalance causes the models to become biased toward predicting the majority classes, making them less sensitive in detecting the minority genres. As a result, many relevant genres are missed during prediction, leading to lower recall. Since F1 score is the harmonic mean of precision and recall, the moderate recall also pulls down the overall F1 score, despite having relatively high precision. Additionally, multi-label classification is more challenging because the model must correctly predict multiple labels for each input, further impacting these scores.

Overall, Neural Network and Logistic Regression demonstrated stronger and more balanced performance across all metrics, while Random Forest was relatively weaker, particularly in precision.

5.2 Test Result

```
========================================
 Song Recommendation Model Selection
========================================

Select a model to use for recommendations:
1. Neural Network
2. Logistic Regression
3. Random Forest
q. Quit

Enter your choice (1-3, or q to quit):  1

Selected model: Neural Network

Enter your caption:  i am happy today

Top 5 genres from Neural Network:
- easy listening: 82.18%
- blues: 42.85%
- R&B: 32.48%
- Dance/Electronic: 15.71%
- hip hop: 12.74%

Recommended Songs:
 Sexbomb by Tom Jones (easy listening, confidence: 82.2%)
 Hey Baby (Radio Mix) by DJ Ötzi (easy listening, confidence: 82.2%)
 A Little Less Conversation - JXL Radio Edit Remix by Elvis Presley (easy listening, confidence: 82.2%)
 Maria Maria (feat. The Product G&B) by Santana (blues, confidence: 42.9%)
 Seven Nation Army by The White Stripes (blues, confidence: 42.9%)
 Thong Song by Sisqo (R&B, confidence: 32.5%)
 Lady - Hear Me Tonight by Modjo (Dance/Electronic, confidence: 15.7%)
```

Figure 26: Test Result 1

From Figure 20, Neural Network model is selected to test with the caption. The input is analysed by the Neural Network to predict the most fitting music genres for the sentiment and context of the caption. Based on the caption, the Neural Network predicts the likelihood (confidence score) of different music genres matching the caption, from the highest to lowest. This suggests that the model considers “easy listening” music as the best match for a happy mood, followed by genres like blues and R&B. Using the genre predictions, the system recommends some songs associated with the top genres.

```
Enter your choice (1-3, or q to quit):  2

Selected model: Logistic Regression

Enter your caption:  i am happy today

Top 5 genres from Logistic Regression:
- Folk/Acoustic: 100.00%
- classical: 91.35%
- blues: 63.44%
- R&B: 59.56%
- easy listening: 45.59%

Recommended Songs:
 Sexbomb by Tom Jones (Folk/Acoustic, confidence: 100.0%)
 Go Let It Out by Oasis (Folk/Acoustic, confidence: 100.0%)
 Breathless by The Corrs (Folk/Acoustic, confidence: 100.0%)
 Way down We Go by KALEO (classical, confidence: 91.3%)
 Maria Maria (feat. The Product G&B) by Santana (blues, confidence: 63.4%)
 Seven Nation Army by The White Stripes (blues, confidence: 63.4%)
 Another Way to Die by Jack White (blues, confidence: 63.4%)
 Thong Song by Sisqo (R&B, confidence: 59.6%)
 Say My Name by Destiny's Child (R&B, confidence: 59.6%)
 Try Again by Aaliyah (R&B, confidence: 59.6%)
 Hey Baby (Radio Mix) by DJ Ötzi (easy listening, confidence: 45.6%)
 A Little Less Conversation - JXL Radio Edit Remix by Elvis Presley (easy listening, confidence: 45.6%)
 The Real Slim Shady by Eminem (hip hop, confidence: 34.9%)
 The Next Episode by Dr. Dre (hip hop, confidence: 34.9%)
 Lady - Hear Me Tonight by Modjo (Dance/Electronic, confidence: 32.5%)
```

Figure 27: Test Result 2

The next model is Logistic Regression, tested with the same caption used in Figure 20. The top genres that Logistic Regression predicts are different from the predictions of Neural Network. The most relevant genre predicted is folk/acoustic, followed by classical, blues, R&B and easy listening.

```
Enter your choice (1-3, or q to quit):  3

Selected model: Random Forest

Enter your caption:  i am happy today

Top 5 genres from Random Forest:
- easy listening: 80.69%
- blues: 48.11%
- R&B: 32.42%
- hip hop: 8.81%
- classical: 6.63%

Recommended Songs:
 Sexbomb by Tom Jones (easy listening, confidence: 80.7%)
 Hey Baby (Radio Mix) by DJ Ötzi (easy listening, confidence: 80.7%)
 A Little Less Conversation - JXL Radio Edit Remix by Elvis Presley (easy listening, confidence: 80.7%)
 Maria Maria (feat. The Product G&B) by Santana (blues, confidence: 48.1%)
 Seven Nation Army by The White Stripes (blues, confidence: 48.1%)
 Thong Song by Sisqo (R&B, confidence: 32.4%)
 The Real Slim Shady by Eminem (hip hop, confidence: 8.8%)
```

Figure 28: Test Result 3

Random Forest is tested with the same caption too as shown in Figure 22. It prioritizes “easy listening” with a high confidence of 80.69%, followed by “blues,” “R&B,” “hip hop,” and “classical.”

5.3 Chapter Summary and Evaluation

In this chapter, the implementation of the song recommender system is discussed in detail. The preprocessing steps and the process of applying sentiment analysis and contextual analysis have been explained. The three different machine learning models, Neural Network, Logistic Regression and Random Forest are also discussed and compared. Lastly, the test results of the overall system are clarified and we can know that there are some limitations in the recommendations.

Chapter 6

Discussions and Conclusion

6 Discussions and Conclusion

6.1 Summary

A Caption-Based Song Recommendation System is developed in this project. The system consists of three major components: Text Preprocessing, Multi-Label Genre Classification, and Song Recommendation. To build an effective song recommendation engine, Natural Language Processing (NLP) techniques were used to clean and lemmatize the Instagram captions. These captions were then transformed into numerical representations using the TF-IDF vectorization method. For the multi-label classification of music genres, several machine learning models were explored and trained, including Logistic Regression, Random Forest, and a Neural Network (Multi-Layer Perceptron). Each caption may correspond to more than one genre, and therefore MultiLabelBinarizer was applied to handle the multi-label nature of the problem.

To evaluate model performance, metrics such as Hamming Loss, Precision, Recall, and F1-score were calculated. Visualization tools like ROC curves and learning curves were also used to further understand each model's effectiveness. After successful training, the best-performing models were used to generate music genre predictions based on new caption inputs. These predictions were then mapped to actual songs with corresponding artists using a genre-to-song dictionary.

For the deployment interface, PySimpleGUI was chosen to create a simple and interactive desktop application where users can input their captions, select a preferred model, and receive song recommendations. The use of TF-IDF vectorizer and Scikit-learn models provides a balance between performance and interpretability, while PySimpleGUI offers a lightweight and accessible user interface. Overall, this system combines NLP, machine learning, and GUI design to provide a creative and user-friendly solution for personalized music recommendations.

6.2 Achievements

The project successfully achieved its main objective of building a song recommender system based on Instagram captions using machine learning and natural language processing techniques. The system can analyse user input and suggest songs based on the predicted genre. Key components such as data preprocessing, model training, evaluation, and a user-friendly interface were completed. One of the key strengths of the project is its integration of sentiment analysis and contextual analysis to recommend songs, which sets it apart from many existing systems that do not suggest song genres based on user-generated text like captions. However, limitations such as dataset imbalance and the moderate performance of some models were identified as weaknesses. Overall, the project is functional and provides a solid foundation for further improvement.

6.3 Contributions

The proposed Caption-Based Song Recommendation System showcases creativity and innovation by bridging the gap between social media expression and personalized music discovery. Unlike traditional music recommendation systems that rely on user listening history, this system interprets the emotional and thematic context of user-generated captions to suggest relevant songs. This novel approach introduces a fresh and engaging way to connect users with music that matches their moods or moments.

The system is necessary because current platforms, like Instagram, lack built-in music intelligence that responds contextually to text. By providing automatic song suggestions based on captions, the system enhances user experience and content personalization, especially for those who enjoy pairing their posts with meaningful music.

In terms of marketability, this solution has strong potential. It could be integrated into social media platforms or music streaming services as a plug-in or feature, appealing to both users and influencers who value expressive and aesthetic post creation. Its unique blend of NLP, machine learning, and user interaction opens opportunities for commercial partnerships in the tech, music, and social media industries.

6.4 Limitations and Future Improvements

There are several limitations of this research project. For example, the test result of any caption will always predict “easy listening” genre as the most relevant genre that has the highest probability, leading to biased predictions. This is very likely due to the imbalance of the dataset. The “easy listening” class in the original dataset has significantly more samples than other genres, therefore caused the model to favour that class. Future improvements can be done by applying down sampling method to the overrepresented genres, which can help the model learn a more balanced view of all genres. The dataset itself can also be improved by collecting more data from underrepresented genres to ensure better representation across all classes.

Other than that, another limitation of the current research lies in the choice of machine learning models. For example, the recall score and f1-score of three of the models are similar and they are unable to go beyond a certain accuracy. Data imbalance and model complexity might be the reasons. While models like Logistic Regression, Random Forest, and Multi-Layer Perceptron (MLP) provide decent performance, they may not be the most optimal for handling complex patterns in textual data, especially in a multi-label classification setting. Future improvements can be made by exploring deep learning models such as Recurrent Neural Networks (RNNs) or Transformer-based models to capture the sequential nature and deeper meaning of the captions.

6.5 Issues and Solutions

I have encountered some issues throughout the project. One of the main challenges encountered during the development of this project was time management, particularly when it came to completing and submitting documentation chapters before their respective deadlines. Balancing the need to write each chapter thoroughly while simultaneously conducting research and testing the implementation of the code was a significant pressure point. Often, I found myself rushing to complete the chapters because a considerable amount of time was needed to understand the models, experiment with different techniques, and resolve coding errors. This situation taught me the importance of proper planning and breaking tasks into smaller milestones. I started allocating time for both coding and documentation to manage the workload better. In the future, I plan to begin documentation alongside implementation to avoid last-minute stress and ensure steady progress.

References

Abbasi-Moud, Z., Vahdat-Nejad, H., & Sadri, J. (2021). Tourism recommendation system based on semantic clustering and sentiment analysis. *Expert Systems with Applications*, *167*. https://doi.org/10.1016/j.eswa.2020.114324

Abdelrazek, A., Eid, Y., Gawish, E., Medhat, W., & Hassan, A. (2023). Topic modeling algorithms and applications: A survey. *Information Systems*, *112*. https://doi.org/10.1016/j.is.2022.102131

Adiyansjah, Gunawan, A. A. S., & Suhartono, D. (2019). Music recommender system based on genre using convolutional recurrent neural networks. *Procedia Computer Science*, *157*, 99–109. https://doi.org/10.1016/j.procs.2019.08.146

Andjelkovic, I., Parra, D., & O'Donovan, J. (2019). Moodplay: Interactive music recommendation based on Artists' mood similarity. *International Journal of Human Computer Studies*, *121*, 142–159. https://doi.org/10.1016/j.ijhcs.2018.04.004

Asani, E., Vahdat-Nejad, H., & Sadri, J. (2021). Restaurant recommender system based on sentiment analysis. *Machine Learning with Applications*, *6*, 100114. https://doi.org/10.1016/j.mlwa.2021.100114

Birjali, M., Kasri, M., & Beni-Hssane, A. (2021). A comprehensive survey on sentiment analysis: Approaches, challenges and trends. *Knowledge-Based Systems*, *226*. https://doi.org/10.1016/j.knosys.2021.107134

Bonta, V., Kumaresh, N., & Janardhan, N. (2019). A Comprehensive Study on Lexicon Based Approaches for Sentiment Analysis. *Asian Journal of Computer Science and Technology*, *8*(S2), 1–6. https://doi.org/10.51983/ajcst-2019.8.s2.2037

Fessahaye, F., Perez, L., Zhan, T., Zhang, R., Fossier, C., Markarian, R., Chiu, C., Zhan, J., Gewali, L., & Oh, P. (2019). *T-RECSYS: A Novel Music Recommendation System Using Deep Learning*. https://doi.org/10.1109/ICCE.2019.8662028

Hu, S., Kumar, A., Al-Turjman, F., Gupta, S., Seth, S., & Shubham. (2020). Reviewer Credibility and Sentiment Analysis Based User Profile Modelling for Online Product Recommendation. *IEEE Access*, *8*, 26172–26189. https://doi.org/10.1109/ACCESS.2020.2971087

Introducing some changes to our Web API. (2024, November 27). Spotify for Developers. https://developer.spotify.com/blog/2024-11-27-changes-to-the-web-api

Kumar, S., De, K., & Roy, P. P. (2020). Movie Recommendation System Using Sentiment Analysis from Microblogging Data. *IEEE Transactions on Computational Social Systems*, *7*(4), 915–923. https://doi.org/10.1109/TCSS.2020.2993585

Liu, B. (2022). Sentiment Analysis and Opinion Mining. In *Synthesis lectures on human language technologies*. https://doi.org/10.1007/978-3-031-02145-9

Moscato, V., Picariello, A., & Sperli, G. (2021). An Emotional Recommender System for Music. *IEEE Intelligent Systems*, *36*(5), 57–68. https://doi.org/10.1109/MIS.2020.3026000

Negara, E. S., Triadi, D., & Andryani, R. (n.d.). *Topic Modelling Twitter Data with Latent Dirichlet Allocation Method*. https://doi.org/10.1109/ICECOS47637.2019.8984523

Osman, N. A., Noah, S. A. M., & Darwich, M. (2019). Contextual sentiment based recommender system to provide recommendation in the electronic products domain. *International Journal of Machine Learning and Computing*, *9*(4), 425–431. https://doi.org/10.18178/ijmlc.2019.9.4.821

Pimentel, J. F., Murta, L., Braganholo, V., & Freire, J. (2019). A large-scale study about quality and reproducibility of jupyter notebooks. *IEEE International Working Conference on Mining Software Repositories*, *2019-May*, 507–517. https://doi.org/10.1109/MSR.2019.00077

Rolon-Mérette, D., Ross, M., Rolon-Mérette, T., & Church, K. (2020). Introduction to Anaconda and Python: Installation and setup. *The Quantitative Methods for Psychology*, *16*(5), S3–S11. https://doi.org/10.20982/tqmp.16.5.s003

Roy, D., & Dutta, M. (2022). A systematic review and research perspective on recommender systems. *Journal of Big Data*, *9*(1). https://doi.org/10.1186/s40537-022-00592-5

Sarin, E., Vashishtha, S., Megha, & Kaur, S. (2022). SentiSpotMusic: a music recommendation system based on sentiment analysis. *4th International Conference on Recent Trends in Computer Science and Technology, ICRTCST 2021 - Proceedings*, 373–378. https://doi.org/10.1109/ICRTCST54752.2022.9781862

Singh, P. K., Choudhury, P., Dey, A. K., & Pramanik, P. K. D. (2021). Recommender systems: an overview, research trends, and future directions. *International Journal of Business and Systems Research*, *15*(1), 14. https://doi.org/10.1504/ijbsr.2021.10033303

This page is intentionally left blank to indicate the back cover. Ensure that the back cover is black in color.