Modeling Music Preferences with Fuzzy Logic for Personalized Recommendation Systems

Qodri Azkarayan – 13523010[1,2]
Bachelor's Program in Informatics Engineering
School of Electrical Engineering and Informatics
Bandung Institute of Technology, Ganesha Street No. 10, Bandung
[1]13523010@mahasiswa.itb.ac.id, [2]azkarayan05@gmail.com

***Abstract*—This paper presents a fuzzy logic-based music recommendation system to model subjective and dynamic user preferences. Traditional recommendation systems often rely on rigid algorithms that struggle to account for human preferences' complex and transient nature. The proposed system uses fuzzy membership functions and rule-based inference to address these limitations and integrate variables such as age, mood, listening time, and tempo. By categorising music into genres and suggesting specific songs, the system demonstrates adaptability to diverse user contexts. Testing results validate its effectiveness, with personalised recommendations aligning closely with individual profiles. This study underscores fuzzy logic's potential in enhancing recommendation systems, providing a flexible, human-like approach to music personalisation.**

Keywords*—*Fuzzy Logic, Music Recommendation, User Profiling, Personalized Music, Recommendation System

I. Introduction

Digital music streaming services have revolutionised how individuals interact with music, offering unlimited access to an extensive library of songs. However, this abundance of choices presents a significant challenge in ensuring users can easily discover music aligned with their unique tastes and preferences. Personalised music recommendation systems are designed to enhance user experience by tailoring suggestions to individual profiles. These systems are crucial for user satisfaction and vital to retaining and engaging users on streaming platforms.

Traditional recommendation systems typically rely on collaborative filtering, content-based filtering, or hybrid approaches. While these methods have proven effective, they often struggle to address human preferences' complex and dynamic nature. Collaborative filtering heavily depends on user activity data, which may not reflect temporary factors such as listening times or fluctuating moods. Similarly, content-based systems rely on predefined musical attributes, limiting their flexibility in adapting to specific user context inputs. These limitations underscore the need for a more adaptive approach that mirrors human-like music recommendation methods.

This paper proposes a fuzzy logic-based music recommendation system to address these challenges. Inspired by how humans make decisions in uncertain or ambiguous situations, fuzzy logic is well-suited to model dynamic and subjective variables. Unlike rigid algorithms, fuzzy logic allows for integrating uncertain inputs, such as user mood, age, preferred tempo, and time of day, to generate more personalised recommendations. The proposed approach delivers music genre recommendations tailored to users' diverse and evolving contexts by leveraging fuzzy membership functions and rule-based inference systems.

The study aims to demonstrate how fuzzy logic can enhance the flexibility and adaptability of music recommendation systems. The proposed system incorporates various user inputs into a comprehensive model, offering context-aware, user-centric recommendations. To illustrate its effectiveness, the system categorises music into genres such as Classical, Ballad, Pop, Hip-hop, and EDM. It recommends specific songs within each genre based on fuzzy inference results.

This paper is organised as follows: Section 1 introduces the study's content. Section 2 provides a literature review, discussing existing music recommendation approaches and their limitations. Section 3 details the implementation, including source code and the design of fuzzy membership functions and rules. Section 4 describes the experiment, presenting test cases and examples of recommendations generated for diverse user profiles. Finally, Section 5 concludes the paper and suggests directions for future research. This study contributes to developing personalised music recommendation systems by bridging the gap between user expectations and traditional algorithmic methods.

II. Literature Review

A. Fuzzy Logic

Collaborative Fuzzy logic is a computational paradigm that handles imprecision and uncertainty in data by emulating the way humans reason and make decisions. It was introduced by Lotfi A. Zadeh[1] in 1965 through his seminal work on fuzzy sets, which provide a mathematical means of representing vagueness and ambiguity. Unlike classical logic, which relies on binary true/false values, fuzzy logic allows values to exist within a continuum between 0 and 1. This makes it particularly useful in situations where crisp distinctions are difficult to define, such as "hot," "cold," or "comfortable."

In fuzzy logic, elements have varying degrees of membership in a set, represented by membership functions. For instance, the temperature "30°C" can belong to the "hot" category with a degree of 0.8 and the "warm" category with a degree of 0.4. Depending on the application, membership functions can take shapes like triangular, trapezoidal, or Gaussian.

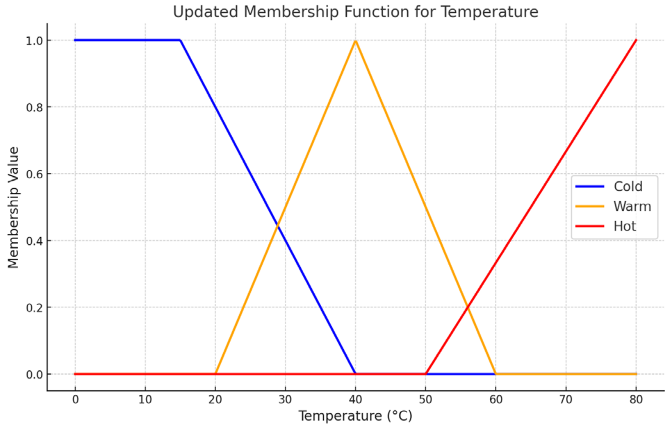

Figure 1: Example of Membership Function for the Temperature Feature

Decisions in fuzzy logic are based on "if-then" rules that mirror human reasoning. For example, if the temperature is "high" and the humidity is "low," then the fan speed should be "fast." These rules are processed through inference systems to derive outputs. A fuzzy inference system processes inputs using fuzzy rules to produce outputs, with common methods including Mamdani and Sugeno models. Defuzzification techniques are applied to convert fuzzy conclusions back into precise outputs. For example, the fuzzy result "fan speed is medium-to-fast" can be translated into a specific RPM value.

Fuzzy logic has several advantages. It mimics how humans make decisions based on vague or incomplete information and adapts to changing conditions, making it ideal for real-world applications. It is computationally simple compared to other machine learning methods and is effective in applications like control systems and decision support.

Applications of fuzzy logic include control systems in appliances like air conditioners, washing machines, and cameras to provide optimal settings. In medical diagnostics, it helps interpret patient data to suggest probable conditions. It also plays a role in recommendation systems by dynamically modelling human preferences, such as personalised music recommendations.

B. Existing Recommendation Systems

Collaborative Filtering recommendation systems have become indispensable in various domains, including e-commerce, entertainment, education, and social media. Collaborative Filtering (CF) is one of the most prevalent techniques among these systems. Collaborative Filtering predicts a user's preferences based on the interactions and preferences of similar users. While effective in generating personalised recommendations, CF suffers from well-documented limitations.

One of the most significant challenges of CF is the cold-start problem, which arises when new users or items lack sufficient interaction data. This limitation reduces the system's effectiveness in offering relevant recommendations, particularly for users who have just joined the platform or for items recently added to the database. Additionally, CF methods are inherently dependent on large volumes of historical data, which poses a problem in contexts where interaction data is sparse or inconsistent.

To address these challenges, researchers have explored hybrid models. For instance, Khairova et al.[2], 2024 introduced a system that integrates CF with fuzzy logic, creating a model capable of adapting to sparse datasets. The hybrid approach significantly improves adaptability and recommendation quality by incorporating fuzzy logic's ability to handle uncertainty and ambiguity. Despite such advancements, CF-based models remain constrained by their reliance on historical data, which limits their capacity to adapt to dynamic, real-time contexts.

Another widely used approach, Content-Based Filtering (CBF), focuses on item attributes rather than user interactions. CBF systems recommend items similar to those the user has already interacted with based on characteristics such as genre, price, or description. While this method avoids the cold-start issue associated with CF, it introduces other problems. For example, CBF systems often lack diversity in recommendations, as they tend to overemphasise similarity and fail to introduce novel items. Additionally, these systems struggle to account for subjective user factors like mood or temporal preferences.

Recent studies emphasise integrating dynamic and subjective inputs into recommendation frameworks. For example, Wan & Lu[3], 2024, proposed a fuzzy logic framework to enhance CBF systems by incorporating evolving user preferences. Their research demonstrates that fuzzy logic enables modelling complex, subjective factors, providing recommendations that better align with a user's current context or mood. However, such implementations require significant computational resources, which can limit their practicality in real-time scenarios.

Hybrid Systems, which combine CF and CBF approaches, are often seen as a promising solution to overcome the limitations of both techniques. These systems leverage the strengths of CF and CBF, resulting in more robust and accurate recommendations. However, hybrid methods also face challenges, particularly in computational complexity. They require the integration of multiple algorithms and large-scale data processing, which can hinder their scalability and real-time performance.

Kulkarni & Rodd[4], 2020, conducted an extensive review of hybrid recommendation systems and suggested the inclusion of fuzzy logic to enhance personalisation. These systems can offer context-aware recommendations using fuzzy sets to model subjective and dynamic inputs. Despite their potential, hybrid systems remain underutilised in domains requiring real-time adaptability due to their resource-intensive nature.

C. *Factor of Song Preference*

Recent research emphasises the dynamic interplay between these contextual and musical attributes to better cater to individual needs and habits. Demographic variables, such as age and personality traits, profoundly impact music preferences. Studies have shown that different age groups exhibit distinctive listening habits. For instance, research by Ferwerda et al.[5] (2014) revealed that older audiences tend to gravitate towards slower and more melodic genres, reflecting a preference for soothing and contemplative experiences. Conversely, younger listeners often favour more upbeat and energetic tracks that align with their active and social lifestyles. Personality traits also play a significant role, with extroverts typically drawn to high-energy music that mirrors their sociable and outgoing nature.

Psychological states, particularly mood, are pivotal in shaping music preferences. Music is often used to regulate emotions, whether to amplify a current feeling or shift to a desired emotional state. Li et al.[6] (2022) highlighted how mood-aware recommendation systems utilise real-time data from wearable technologies to detect emotional states and tailor music suggestions accordingly. Such systems have the potential to enhance user engagement by aligning music choices with the listener’s psychological needs at any given moment.

Situational contexts, including the time of day and user activities, further refine listening behaviours. Temporal patterns in music consumption reveal distinct preferences based on daily routines. For example, research by Moes[7] (2023) found that listeners commonly choose upbeat and energising music during the morning to kickstart their day while opting for relaxing genres in the evening to unwind. Activities such as exercising, commuting, or studying also shape music selection, as users seek tracks that complement their environment and goals.

Finally, musical features such as tempo, genre, and timbre are crucial in defining preferences. The tempo of a track often correlates with its perceived energy, making faster beats ideal for active moments and slower tempos suitable for relaxation. Matrosova[8] (2024) demonstrated that these features align with individual moods and enhance the overall listening experience by providing emotional resonance. Genres, too, play a significant role, serving as familiar anchors for users while allowing room for exploration through blends and discoveries.

D. *skfuzzy*

According to Warner[9] (2022) from GitHub, skfuzzy is a Python library that provides tools for working with fuzzy logic. The library includes tools for creating fuzzy sets, defining fuzzy rules, and implementing fuzzy inference systems. Membership functions such as triangular, trapezoidal, and Gaussian can be used to map input values to degrees of membership. Fuzzy rules allow relationships between inputs and outputs to be defined. These rules are evaluated using inference engines to produce actionable outputs from fuzzy inputs.

The control submodule in skfuzzy focuses on building and simulating fuzzy control systems. Inputs and outputs are defined using ctrl.Antecedent and ctrl.Consequent, respectively. Fuzzy rules are used to combine inputs and outputs to model system behaviour. For example, defining a rule like "If temperature is hot, then fan speed is high" establishes the relationship between input and output variables. These rules are grouped into a ctrl.ControlSystem can then simulate and compute output values based on input conditions, such as determining the fan speed for a temperature of 80°F.

III. Implementation

The implemented system provides a sophisticated, personalised music recommendation engine that leverages fuzzy logic principles to align music genres with subjective user preferences. These preferences include age, mood, listening time, and preferred tempo, which are inherently variable and imprecise. By incorporating fuzzy inference rules and membership functions, the system translates these nuanced inputs into a recommendation score that maps directly to specific music genres. This innovative approach allows the system to bridge the gap between the fluidity of human emotions and the structured categorisation of music genres.

At the system's core lies the seamless integration of several libraries that enable its functionality. The program begins by importing essential libraries: Numpy (as np), which facilitates numerical operations and array manipulation critical for defining and managing the ranges of fuzzy variables; Skfuzzy, a specialised library for fuzzy logic computations, with its control submodule particularly useful for defining and managing fuzzy control systems, including input variables (antecedents), output variables (consequents), and rules; and the Random library, used to select songs from a predefined list, ensuring a dynamic and varied user experience based on the recommended genre. Combining these tools establishes a robust foundation for creating fuzzy logic variables, rules, and recommendations.

The initial setup involves defining fuzzy logic variables such as age, mood, listening time, and tempo. These variables are assigned specific ranges and divided into meaningful linguistic categories (e.g., "young," "happy," "morning," "fast"). Membership functions, such as trapezoidal (trapmf) or triangular (trimf), represent these categories. This step is crucial as it transforms quantitative inputs into qualitative descriptors that the fuzzy logic system can process. For example, age might be categorised into "young," "middle-aged," and "senior"; mood could range from "sad" to "happy"; listening time might have categories like "morning," "afternoon," and "evening"; and tempo can be described as "slow," "moderate," or "fast." These membership functions provide the system's structural basis for evaluating user inputs and establishing connections between them.

```python
import numpy as np
import skfuzzy as fuzz
from skfuzzy import control as ctrl
import random

age = ctrl.Antecedent(np.arange(10, 61, 1), 'age')
mood = ctrl.Antecedent(np.arange(0, 11, 1), 'mood')
listening_time = ctrl.Antecedent(np.arange(0, 25, 1), 'listening_time')
tempo = ctrl.Antecedent(np.arange(0, 201, 1), 'tempo')
recommendation = ctrl.Consequent(np.arange(0, 101, 1), 'recommendation')

age['young'] = fuzz.trapmf(age.universe, [10, 10, 20, 30])
age['adult'] = fuzz.trapmf(age.universe, [20, 35, 40, 50])
age['senior'] = fuzz.trapmf(age.universe, [40, 50, 60, 60])
mood['sad'] = fuzz.trimf(mood.universe, [0, 0, 5])
mood['neutral'] = fuzz.trimf(mood.universe, [3, 5, 7])
mood['happy'] = fuzz.trimf(mood.universe, [5, 10, 10])
listening_time['morning'] = fuzz.trapmf(listening_time.universe, [0, 0, 9, 12])
listening_time['afternoon'] = fuzz.trimf(listening_time.universe, [11, 15, 18])
listening_time['evening'] = fuzz.trimf(listening_time.universe, [18, 19, 21])
listening_time['night'] = fuzz.trimf(listening_time.universe, [21, 24, 24])
tempo['slow'] = fuzz.trapmf(tempo.universe, [0, 0, 60, 90])
tempo['moderate'] = fuzz.trimf(tempo.universe, [80, 120, 140])
tempo['fast'] = fuzz.trapmf(tempo.universe, [130, 160, 200, 200])

recommendation['Classical'] = fuzz.trimf(recommendation.universe, [0, 0, 20])
recommendation['Ballad'] = fuzz.trimf(recommendation.universe, [15, 35, 50])
recommendation['Hip-hop'] = fuzz.trimf(recommendation.universe, [45, 60, 75])
recommendation['EDM'] = fuzz.trimf(recommendation.universe, [70, 85, 90])
recommendation['Pop'] = fuzz.trimf(recommendation.universe, [85, 100, 100])
```

Figure 2. Library Import and Program Setups
Source: https://github.com/qodriazka/fuzzy-logic-song-rec

A key feature of the system is its dynamic rule-generation mechanism. This process iterates through all possible combinations of input variables and their respective categories. Each combination is evaluated to determine the most appropriate music genre recommendation. For instance, a user categorised as "young" with a "happy" mood and a preference for a "fast" tempo might be recommended the "Pop" genre. Conversely, a "senior" user feeling "sad" might be directed toward "Classical" music. Fallback rules ensure that the system provides meaningful recommendations even in less straightforward cases, such as suggesting "Ballad" for "moderate" tempo preferences or "Hip-hop" as a general option.

These rules are implemented using the ctrl.Rule function, where the antecedent represents the logical combination of input categories, and the consequent specifies the resulting music genre. All rules are appended to a central list, forming the fuzzy control system's core decision-making logic. This comprehensive rule set ensures the system can handle various input combinations effectively.

```python
rules = []
for age_group in age.terms.keys():
    for mood_level in mood.terms.keys():
        for time_period in listening_time.terms.keys():
            for tempo_speed in tempo.terms.keys():
                if age_group == 'young' and mood_level == 'happy':
                    if tempo_speed == 'fast':
                        rule = ctrl.Rule(age[age_group] & mood[mood_level] & listening_time[time_period] & tempo[tempo_speed], recommendation['Pop'])
                    else:
                        rule = ctrl.Rule(age[age_group] & mood[mood_level] & listening_time[time_period] & tempo[tempo_speed], recommendation['EDM'])
                elif age_group == 'senior' and mood_level == 'sad':
                    rule = ctrl.Rule(age[age_group] & mood[mood_level] & listening_time[time_period] & tempo[tempo_speed], recommendation['Classical'])
                elif tempo_speed == 'moderate':
                    rule = ctrl.Rule(age[age_group] & mood[mood_level] & listening_time[time_period] & tempo[tempo_speed], recommendation['Ballad'])
                else:
                    rule = ctrl.Rule(age[age_group] & mood[mood_level] & listening_time[time_period] & tempo[tempo_speed], recommendation['Hip-hop'])
                rules.append(rule)
```

Figure 3. Rule Creation

Source: https://github.com/qodriazka/fuzzy-logic-song-rec

The fuzzy control system is the computational engine that processes user inputs to generate a recommendation score. Once received, inputs are matched to predefined fuzzy categories using membership functions. This approach allows the system to interpret the inputs as nuanced values rather than rigid data points. The compiled rule set then evaluates these inputs to determine their influence on the recommendation output. For instance, a combination of "young," "happy," and "fast" might result in a high likelihood of recommending "Pop." At the same time, inputs like "middle-aged," "calm," and "moderate" tempo could lead to a "Ballad" recommendation. The control system computes a recommendation score based on the degree of membership in each fuzzy category and the outcomes of the rules. This score is mapped to a specific music genre, delivering a tailored recommendation aligned with the user's preferences.

```python
recommendation_ctrl = ctrl.ControlSystem(rules)
recommendation_sim = ctrl.ControlSystemSimulation(recommendation_ctrl)
```

Figure 4. Fuzzy Control System
Source: https://github.com/qodriazka/fuzzy-logic-song-rec

Each recommended genre corresponds to a predefined collection of song titles stored in a structured dictionary. This dictionary serves as a repository, where each genre is a key linked to a list of tracks that embody its essence and emotional tone. For example, the Classical genre features soothing and melodic tracks that evoke calmness or introspection. Hip-hop includes rhythmically dynamic and lyrically vibrant songs that reflect high energy and urban culture. Ballads highlight emotional depth and storytelling through slower, more poignant tracks. EDM offers energetic, beat-driven tracks perfect for upbeat or party moods, while Pop comprises catchy, mainstream hits known for their broad appeal. This carefully curated list ensures the system recommends genres and delivers high-quality song suggestions within each category.

```python
song_titles = {
    'Classical': [
        "Black Swan - BTS", "Feel My Rhythm - Red Velvet", "Miracles in December - EXO",
        "Summer Rain - Gfriend", "Nxde - (G)I-dle", "Butterfly - Loona",
        "Angel - NCT 127", "Secret Garden - Oh My Girl"
    ],
    'Ballad': [
        "This Love - Davichi", "If It Is You - Jung Seunghwan", "Fine - Taeyeon",
        "Me After You - Paul Kim", "Breathe - Lee Hi", "Unspoken Words - Davichi",
        "Through the Night - IU", "You, Clouds, Rain - Heize", "Hug Me - Jung Joon Il"
    ],
    'Hip-hop': [
        "Any Song - Zico", "Spicy - CL", "Zoom - Jessi", "Mommae - Jay Park",
        "Daechwita - Agust D", "Law - Bibi", "Uh Oh - (G)I-dle", "Don't Go Insane - DPR Ian"
    ],
    'EDM': [
        "Bang Bang Bang - Big Bang", "Crazy - Le Sserafim", "POP/STARS - K/DA",
        "Hard Carry - GOT7", "Whiplash - Aespa", "Rising Sun - TVXQ",
        "DDU-DU DDU-DU - Blackpink", "Follow - Monsta X", "Cherry Bomb - NCT 127",
        "Kick It - NCT 127"
    ],
    'Pop': [
        "Likey - Twice", "Lovesick Girls - Blackpink", "Shhh - Kiss Of Life",
        "Left & Right - Seventeen", "Say My Name - ZB1", "Cheer Up - Twice",
        "Boy With Luv - BTS", "Dynamite - BTS", "Fancy - Twice",
        "Butter - BTS"
    ]
}
```

Figure 5. Song Title Dictionary
Source: https://github.com/qodriazka/fuzzy-logic-song-rec

The user experience begins with the program prompting personal preferences and listening to context through intuitive questions. Users are asked to provide their age and mood on a scale ranging from sad to happy, the hour of the day they typically listen to music, and their preferred tempo in beats per minute (BPM).

```
try:
    user_age = get_valid_input("Enter your age (10-60): ", 10, 60, int)
    user_mood = get_valid_input("Enter your mood (0-10, 0=sad, 10=happy): ", 0, 10, int)
    user_listening_time = get_valid_input("Enter the hour of the day you listen to music (0-24): ", 0, 24, int)
    user_tempo = get_valid_input("Enter your preferred tempo (BPM, 0-200): ", 0, 200, int)
    print(f"Inputs: age={user_age}, mood={user_mood}, listening_time={user_listening_time}, tempo={user_tempo}")

    recommendation_sim.input['age'] = user_age
    recommendation_sim.input['mood'] = user_mood
    recommendation_sim.input['listening_time'] = user_listening_time
    recommendation_sim.input['tempo'] = user_tempo
    recommendation_sim.compute()
```

Figure 6. User Input
Source: https://github.com/qodriazka/fuzzy-logic-song-rec

To ensure input validity, the program includes a helper function that validates responses in real-time, preventing errors and guiding users to provide appropriate inputs.

```
def get_valid_input(prompt, min_value, max_value, input_type):
    while True:
        try:
            user_input = input_type(input(prompt))
            if min_value <= user_input <= max_value:
                return user_input
            print(f"Input must be between {min_value} and {max_value}. Please try again.")
        except ValueError:
            print("Invalid input. Please enter a valid value.")
```

Figure 7. Input Validation
Source: https://github.com/qodriazka/fuzzy-logic-song-rec

Once collected, the inputs are categorised into fuzzy variables such as "young," "happy," "afternoon," or "moderate" using the defined membership functions. These categorised inputs are fed into the fuzzy control system, which simulates the rules to compute the recommendation score. This score determines the most suitable music genre for the user, such as "Classical," "Ballad," "Hip-hop," "EDM," or "Pop." Finally, the program randomly selects three song titles from the genre-specific list and presents them to the user. This process ensures each recommendation is unique and personalised, providing a rich and engaging music discovery experience.

```
if recommendation_score <= 20:
    recommended_genre = 'Classical'
elif recommendation_score <= 50:
    recommended_genre = 'Ballad'
elif recommendation_score <= 75:
    recommended_genre = 'Hip-hop'
elif recommendation_score <= 90:
    recommended_genre = 'EDM'
else:
    recommended_genre = 'Pop'

if recommended_genre in song_titles:
    selected_songs = random.sample(song_titles[recommended_genre], 3)
    print(f"Your recommendation score is: {recommendation_score:.2f}")
    print(f"Recommended genre: {recommended_genre}")
    print("Here are 3 song suggestions:")
    for song in selected_songs:
        print(f"- {song}")
else:
    raise ValueError(f"No songs found for genre: {recommended_genre}")
```

Figure 8. Recommendation Computation
Source: https://github.com/qodriazka/fuzzy-logic-song-rec

IV. Experiment

A. *Test Case 1*

```
Enter your age (10-60): 17
Enter your mood (0-10, 0=sad, 10=happy): 10
Enter the hour of the day you listen to music (0-24): 13
Enter your preferred tempo (BPM, 0-200): 160
Categorized Inputs:
  Age: young
  Mood: happy
  Listening Time: afternoon
  Tempo: fast
Your recommendation score is: 94.17
Recommended genre: Pop
Here are 3 song suggestions:
- Cheer Up - Twice
- Fancy - Twice
- Dynamite - BTS
```

Figure 9. First Test Case
Source: https://github.com/qodriazka/fuzzy-logic-song-rec

This test case demonstrates the execution of the fuzzy logic-based music recommendation system. A user provides inputs in this system, and the program processes them to output a tailored music recommendation. The user enters 17 as their age, indicating they are young. They rate their mood as 10 on a scale of 0 to 10, signalling they are very happy. The listening time is specified as 13, corresponding to the early afternoon, and the user indicates a tempo preference of 160 BPM, suggesting they prefer fast-paced music.

The input values are mapped to fuzzy categories. The age value of 17 corresponds to the category young since it falls within the range defined by the trapezoidal membership function for young individuals (10–30). A mood value of 10 fully activates the fuzzy category happy, as it is the maximum value for the triangular membership function of happiness. The listening time of 13 falls within the category afternoon, as the triangular membership function for this category peaks at around 15 and extends from 11 to 18. A tempo of 160 BPM activates the category fast, as it lies within the range defined by the trapezoidal membership function for fast tempo (130–200 BPM).

The categorised inputs (young, happy, afternoon, fast) are fed into the fuzzy control system, which evaluates the rules to compute the recommendation score. Being young and happy contributes to a high likelihood of recommending upbeat genres like "Pop" or "EDM." The fast tempo preference further reinforces the system's inclination toward genres like "Pop."

The fuzzy control system calculates a recommendation score of 94.17, which falls within the range for the Pop genre (85–100). Based on this score, "Pop" is the recommended genre. Three songs are then randomly selected from the predefined list for the "Pop" genre: "Cheer Up" by Twice, "Fancy" by Twice, and "Dynamite" by BTS.

This test case exemplifies the ability of the fuzzy logic system to process nuanced, subjective inputs (e.g., mood and tempo) in combination with objective factors (e.g., age and time of day). Fuzzy membership functions enable the system to handle imprecise data, while the rule-based approach captures relationships between inputs to produce a contextually appropriate output. The high recommendation score reflects strong rule activations for the chosen genre, validating the system's logic for recommending "Pop" music to a young, happy individual who prefers fast tempos.

B. Test Case 2

```
Enter your age (10-60): 45
Enter your mood (0-10, 0=sad, 10=happy): 2
Enter the hour of the day you listen to music (0-24): 20
Enter your preferred tempo (BPM, 0-200): 70
Categorized Inputs:
  Age: adult
  Mood: sad
  Listening Time: evening
  Tempo: slow
Your recommendation score is: 39.11
Recommended genre: Ballad
Here are 3 song suggestions:
- You, Clouds, Rain - Heize
- If It Is You - Jung Seunghwan
- This Love - Davichi
```

Figure 10. Second Test Case
Source: https://github.com/qodriazka/fuzzy-logic-song-rec

In this test case, the user inputs their age as 45, mood as two on a scale where 0 represents "sad" and 10 means "happy," the listening time as 20 (indicating 8 PM on a 24-hour clock), and their preferred tempo as 70 BPM. Based on the system's fuzzy logic categorisation, the age of 45 is classified as "adult," while the mood value of 2 is categorised as "sad." The listening time of 20 is identified as "evening," and the tempo of 70 BPM is considered "slow." These inputs collectively form the basis for the system's recommendation.

The fuzzy control system processes these inputs and calculates a recommendation score 39.11. This score falls within the range that suggests the "Ballad" genre. From the system's curated database, three songs are selected to match the emotional tone of this genre: "You, Clouds, Rain" by Heize, "If It Is You" by Jung SeungHwan, and "This Love" by Davichi. These song choices reflect the emotional depth and introspective qualities often associated with Ballads, resonating particularly well with a "sad" mood and a "slow" tempo preference.

When comparing this test case to the first, some key differences arise. In the earlier test case, the user may have been younger and thus classified as "young" instead of "adult." This distinction can influence the outcome, as age plays a role in shaping the genre recommendations. Additionally, the earlier test case may have involved a happier mood, which could result in the system recommending a more upbeat genre such as "Pop" or "Hip-hop." In contrast, this test case's "sad" mood drives the recommendation toward the emotionally rich and slower-paced Ballad genre.

The listening time also plays a significant role. In this case, the "evening" categorisation aligns well with reflective or mellow genres like Ballads or Classical music. If the earlier test case had a listening time classified as "morning" or "afternoon," the system might have leaned toward more energetic or dynamic genres. Tempo further reinforces this difference. This test case's "slow" tempo of 70 BPM supports the Ballad recommendation. In contrast, a faster tempo in the earlier case could have resulted in recommendations for more vibrant genres like "EDM" or "Pop."

The recommendation score also highlights the difference in context. The moderate score of 39.11 here aligns with the introspective tone of Ballads, whereas the first test case may have yielded a higher score that mapped to a genre with more energy or variety. The system's adaptability ensures that it provides personalised recommendations based on even subtle differences in user inputs. In this case, the combination of a "sad" mood, "slow" tempo, and "evening" listening time aligns perfectly with the Ballad genre, offering the user a highly personalised music selection.

V. Conclusion

The proposed fuzzy logic-based music recommendation system bridges the gap between rigid traditional algorithms and the dynamic nature of human preferences. The system models subjective and contextual variables with high adaptability by incorporating user inputs such as age, mood, listening time, and tempo into fuzzy membership functions. The rule-based inference mechanism enables the system to produce genre recommendations tailored to individual profiles, addressing static preferences and transient emotions.

Through the implementation and testing of the system, it was demonstrated that fuzzy logic provides a robust framework for handling uncertainty and ambiguity in user inputs. Based on real-time inputs, the system's ability to recommend diverse music genres, from Classical to EDM, underscores its versatility. Additionally, the curated song database enhances user engagement by offering specific, relatable suggestions aligned with each genre.

While the system successfully integrates fuzzy logic for personalised recommendations, it also highlights areas for future improvement. Incorporating real-time data from wearable devices or integrating user feedback loops could further refine the recommendation's accuracy. Expanding the rule set and input variables, such as activity type or location, could enhance its adaptability for broader use cases.

Overall, this study demonstrates the potential of fuzzy logic in transforming recommendation systems. It provides a flexible, user-centric approach that mirrors the complexity of human decision-making. This work lays the foundation for developing advanced, real-time adaptive recommendation engines in the digital music domain.

Acknowledgement

I express my deepest gratitude to God Almighty for His endless guidance, blessings, and strength throughout the journey of completing this paper. I also thank Dr. Ir. Rila Mandala, M.Eng., Ph.D., for his invaluable IF1220 Discrete

Mathematics lecturer role. His guidance and comprehensive explanations have significantly contributed to my understanding of the subject matter, providing a strong foundation for this work.

References

[1] L. A. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, no. 3, pp. 338–353, Jun. 1965, doi: https://doi.org/10.1016/s0019-9958(65)90241-x.

[2] N. Khairova, Nataliia Sharonova, Dmytro Sytnikov, Mykyta Hrebeniuk, and Polina Sytnikova, "Recommendation System Based on a Compact Hybrid User Model Using Fuzzy Logic Algorithms," Apr. 2024, doi: https://doi.org/10.31110/colins/2024-2/005.

[3] L. Wan and J. Lu, "Intelligent Music Recommendation System: Using Algorithms to Improve the Accuracy of Personalized Music Experience," *International Journal of High Speed Electronics and Systems*, Dec. 2024, doi: https://doi.org/10.1142/s0129156425401962.

[4] S. Kulkarni and S. F. Rodd, "Context Aware Recommendation Systems: A review of the state of the art techniques," Computer Science Review, vol. 37, p. 100255, Aug. 2020, doi: https://doi.org/10.1016/j.cosrev.2020.100255.

[5] B. Ferwerda and M. Schedl, "Enhancing Music Recommender Systems with Personality Information and Emotional States: A Proposal.," Jan. 2014.

[6] J. Li et al., "Towards Ubiquitous Personalized Music Recommendation with Smart Bracelets," vol. 6, no. 3, pp. 1–34, Sep. 2022, doi: https://doi.org/10.1145/3550333.

[7] J. Moes, "Context-based User Playlist Analysis for Music Recommendation," *studenttheses.uu.nl*, 2023. https://studenttheses.uu.nl/handle/20.500.12932/45445.

[8] K. Matrosova, "Modeling and Influencing Music Preferences on Streaming Platforms," *Hal.science*, Dec. 2024, doi: https://hal.science/tel-04865002.

[9] J. D. Warner, Ed., "scikit-fuzzy," *GitHub*, Feb. 22, 2023. https://github.com/scikit-fuzzy/scikit-fuzzy.

STATEMENT

Hereby, I declare that the paper I have written is my work, not an adaptation or translation of someone else's paper, and is not plagiarised.

Bandung, 8 January 2025

Qodri Azkarayan
13523010