ABSTRACT:

Title: Anime Review Analysis: Unveiling Genre Trends and Viewer Perceptions

"Anime Review Analysis" is a comprehensive project that delves into the realm of anime series reviews, with a primary focus on "Kimetsu no Yaiba: Mugen Train" and its contemporaries. The project's primary goal is to gain a deeper understanding of viewer perceptions and genre dynamics within the world of anime.

The project begins with the thorough collection of anime series reviews from a diverse range of sources, including online review platforms, social media, and dedicated anime forums. This extensive data collection ensures a well-rounded perspective on viewer opinions.

The analysis conducted in this project is twofold:

First, it explores the prevalence and influence of various anime genres within "Kimetsu no Yaiba: Mugen Train" and its peer series. The study categorizes and assesses genres to unravel how genre preferences affect viewer opinions and contribute to the reception of anime series. The objective is to identify trends and patterns that underline the popularity of specific genres.

Second, the project employs text mining techniques to uncover recurring themes, strengths, and distinctive qualities in anime reviews. By doing so, it seeks to pinpoint the elements that resonate most profoundly with viewers across a spectrum of series, offering insights into what captivates the anime audience.

Additionally, the project includes a comparative analysis to assess the relative popularity of "Kimetsu no Yaiba: Mugen Train" within its peer group. Specific details of this analysis are intentionally kept concise, allowing for adaptability to research findings.

To visually represent the outcomes of the analysis, the project employs data visualization

techniques, such as charts and graphs. These visualizations present genre distribution and the prevalence of strengths within different anime series, enabling a clear and insightful representation of the project's findings.

In conclusion, this project endeavors to unearth the intricate interplay between genre dynamics, strengths, and viewer perceptions in the world of anime, specifically within "Kimetsu no Yaiba: Mugen Train." By offering a more profound understanding of how genres and strengths influence the popularity of anime series, this research serves as a valuable resource for anime enthusiasts and industry professionals seeking to decode the essence of successful anime series.

INTRODUCTION:

Anime (Japanese: アニメ) is hand-drawn and computer-generated animation originating from Japan. Outside Japan and in English, anime refers specifically to animation produced in Japan.[1] However, in Japan and in Japanese, anime (a term derived from a shortening of the English word animation) describes all animated works, regardless of style or origin. Many works of animation with a similar style to Japanese animation are also produced outside Japan. Video games sometimes also feature themes and artstyles that can be considered as "anime".

The earliest commercial Japanese animations date to 1917. A characteristic art style emerged in the 1960s with the works of cartoonist Osamu Tezuka and spread in following decades, developing a large domestic audience. Anime is distributed theatrically, through television broadcasts, directly to home media, and over the Internet. In addition to original works, anime are often adaptations of Japanese comics (manga), light novels, or video games. It is classified into numerous genres targeting various broad and niche audiences.

What distinguishes anime as a medium is its capacity for artistic exploration and storytelling diversity. From action-packed adventures and thought-provoking sci-fi to heartfelt romances and introspective dramas, anime traverses an array of genres, catering to a broad spectrum of viewer preferences. This versatility has fostered a global community of anime enthusiasts, fostering discussions and fostering a sense of belonging among its followers.

Anime is a diverse medium with distinctive production methods that have adapted in response to emergent technologies. It combines graphic art, characterization, cinematography, and other forms of imaginative and individualistic techniques. Compared to Western animation, anime production generally focuses less on movement, and more on the detail of settings and use of "camera effects", such as panning, zooming, and angle shots. Diverse art styles are used, and character proportions and features can be quite varied, with a common characteristic feature being large and emotive eyes.

The anime industry consists of over 430 production companies, including major studios such as Studio Ghibli, Kyoto Animation, Sunrise, Bones, Ufotable, MAPPA, Wit Studio, CoMix Wave Films, Production I.G, and Toei Animation. Since the 1980s, the medium has also seen widespread international success with the rise of foreign dubbed, subtitled programming, and since the 2010s its increasing distribution through streaming services and a widening demographic embrace of anime culture, both within Japan and worldwide. As of 2016, Japanese animation accounted for 60% of the world's animated television shows.

OBJECTIVE :

Identify Key Factors: Analyze anime reviews to uncover pivotal elements (animation quality, plot, characters, soundtracks) impacting the success of anime movies in terms of audience reception and commercial performance.

Understand Audience Preferences: Explore reviews to recognize consistently favored elements within anime movies, unveiling what resonates positively with viewers.

Establish Correlations: Investigate connections between positively reviewed elements and overall success metrics (e.g., box office revenue), outlining the impact of specific aspects on an anime movie's acclaim.

Compare and Extract Insights: Conduct comparative analysis among multiple anime titles to extract distinguishing features contributing to success or failure, providing actionable insights.

Guide Future Productions: Offer actionable feedback based on review analysis, emphasizing audience-preferred elements to steer future anime creations towards success.

In this analysis of anime movie reviews, the primary objectives are to identify key success factors contributing to an anime movie's popularity and commercial performance. The analysis will delve into diverse reviews to discern pivotal elements such as animation quality, storyline, character development, and soundtrack that significantly impact audience reception. Simultaneously, it aims to unveil consistent audience preferences within anime movies and establish correlations between positively reviewed elements and success metrics. Through comparative analysis, the endeavor is to extract actionable insights that provide valuable guidance for future anime productions, emphasizing elements resonating positively with audiences.

METHODOLOGY :

DATA PREPROCESSING :

This section details the methods employed for data gathering from IMDb and the preprocessing steps taken to prepare the IMDb reviews for analysis.

Data Collection:

Web Scraping IMDb Reviews: Utilizing Python's requests library and BeautifulSoup, IMDb reviews for the anime titles "Your Name" and "Mugen Train" were scraped from their respective IMDb pages.

```python
import requests
from bs4 import BeautifulSoup
import time
```

```python
url = "https://www.imdb.com/title/tt5311514/reviews" # url of review page of the movie in IMDB
response = requests.get(url) # getting request to the web page
soup = BeautifulSoup(response.text, 'html.parser')
```

Extraction Method: Reviews were extracted from the HTML structure of IMDb pages using specific class names and attributes.

```python
reviews = []

# Extract existing reviews on the first page
contents = soup.find_all('div', {"class": "text show-more__control"})
for content in contents:
    reviews.append(content.text)

reviews[0]
```

```
'In truth, the most competitive area of fiction is not invasions from outer space or talking dogs or impossible missions but ra
ther the most basic narrative of all, the love story.And this is also the most overcrowded field, and the most difficult to do
right.This film, this story, is extraordinary. It reminds me of MY SASSY GIRL, a love story from Asia that captured the imagina
tion of the world and has almost become a franchise, it\'s been copied so many times.It also brings to mind Richard Matheson\'s
classic BID TIME RETURN, another love story for the ages that uses time juxtaposition. (Done as the movie SOMEWHERE IN TIME, 19
80).I already gave this the highest rating on the IMDb.The animation is stunning and complimentary to this one-of-a-kind tale.A
ll things considered, I would like to avoid the ongoing arguments about which Japanese animation studio is better, or worse, th
an the other.I prefer to simply be grateful that Japanese anime exists at all, because I cannot imagine any other country produ
cing something this moving, this powerful, in graphic form.((Designated "IMDb Top Reviewer." Please check out my list "167+ Nea
rly-Perfect Movies (with the occasional Anime or TV miniseries) you can/should see again and again (1932 to the present))'
```

Preprocessing Steps:

Initial Data Inspection: IMDb reviews were loaded into the Python environment.

Cleaning the Data:

Removal of HTML tags and irrelevant elements from the scraped text.

Conversion of the reviews into a structured format for further analysis.

DataFrame Creation: The extracted reviews were organized into Pandas DataFrames (your_name_df and mugen_train_df) for convenient data handling and analysis.

```
# Extract existing reviews on the first page
contents = soup.find_all('div', {"class": "text show-more__control"})
for content in contents:
    reviews.append(content.text)

# Check if there's a "Load More" button
load_more_button = soup.find('div', {"class": "load-more-data"})
while load_more_button:
    # Extract the data attribute from the "Load More" button
    data_key = load_more_button.get('data-key')
    if data_key:
        # Create a URL for the AJAX request to load more reviews
        ajax_url = f"{url}/_ajax?paginationKey={data_key}"
        ajax_response = requests.get(ajax_url)
        ajax_soup = BeautifulSoup(ajax_response.text, 'html.parser')

        # Extract and append new reviews to the list
        new_contents = ajax_soup.find_all('div', {"class": "text show-more__control"})
        for content in new_contents:
            reviews.append(content.text)

        # Check for the next "Load More" button
        load_more_button = ajax_soup.find('div', {"class": "load-more-data"})

        # IMDb might block requests if you send too many in a short period, so add a delay
        time.sleep(1)
```

```
your_name_df=pd.DataFrame(reviews)
```

```
your_name_df.head()
```

	0
0	In truth, the most competitive area of fiction...
1	'Your Name' is not just one of the best animes...
2	This delightful Japanese animation follows two...
3	Your Name is just a good movie. This movie fel...
4	It's been a while since I've seen something ou...

SENTIMENT ANALYSIS :

The sentiments are calculated using the pre-trained from the huggin face “bert-base-multilingual-uncased-sentiment” model where it the sentiments from 0 to 4 from the tokenzined words from AutoTokenizer function of pre-trained model. The values are converted to 1 to 5 for compatibility where,

1 to 2: Negative reviews while 1 being highly negative

3 : 3 is considered as neutral.

4 to 5 : Positive reviews while 5 being highly positive.

Another type of sentiment tool is used for finding the sentiment towards movie elements. The NLTK’s SentimentIntensityAnalyzer is used. This can detect or extract the sentiment towards the object for object in the sentence. The sentiment is converted to 0, 1 and 2 where 0 being negative and the 1 being neutral and the 2 being positive

TOPIC MODELING :

Employed Latent Dirichlet Allocation (LDA) to extract underlying topics within reviews. Clustered reviews to identify prevalent themes and subjects across different anime.

Utilization of Latent Dirichlet Allocation (LDA): LDA, a probabilistic model, was utilized to uncover latent topics and themes embedded within the corpus of anime reviews. This technique is widely acknowledged for its efficacy in topic modeling by assuming that each review consists of multiple topics and aims to identify these underlying topics without any supervision.

Extraction of Underlying Topics: The application of LDA involved a systematic process where the algorithm analyzed the entire collection of anime reviews. It identified patterns and clusters of words that frequently co-occurred within the reviews, allowing the algorithm to discern inherent topics or themes.

Identification of Prevalent Themes: Through LDA, prevalent themes and subjects prevalent across different anime were revealed. The algorithm grouped reviews based on shared words and phrases, creating distinct clusters representing prevalent topics such as animation quality, character development, storyline intricacies, soundtrack impact, and genre-specific elements.

Cluster Analysis for Topic Identification: The clustering process facilitated the identification of dominant topics prevalent within the reviews. Each cluster represented a coherent theme, enabling the categorization of reviews into different themes or topics, providing insights into the most discussed and significant aspects of various anime titles.

Insights into Anime Content: By deciphering these underlying topics and prevalent themes, LDA allowed for a deeper understanding of the content and key elements that audiences frequently discuss and find noteworthy in anime. This process facilitated the extraction of actionable insights regarding what aspects of anime content resonate most with viewers.

Word Clouds for Topic Representation: Word clouds can be created to visually represent the most frequent words within each identified topic. This visual representation helps in understanding the key terms associated with specific topics. For each topic extracted by LDA, a word cloud can be generated where the size of the words corresponds to their frequency within that topic.

EXAMPLE OUTPUT OF WORD CLOUD

ANALYSIS :

The reviews will be taken from the two most grossed anime movies to analysis the reviews and get insights after analyzing them. There will be several insights and visualization taken, namely:

1. Sentiment of the audience towards the movie.
2. Extracting the most appreciated movie elements from the audience towards that movie
3. Extracting the overall sentiment of the reviews towards the movie.
4. Calculating the genre ranking and genre combination.
5. Extract the elements of anime that are important to the audience.

Several plots , graphs and charts will used to find the pattern or insight or any useful info.
A another dataset will be used which includes all anime's data to get more insights. The reviews and the datasets info's will compared to get more insight.At last we will be using the Topic Modeling to help with analyzing and comparing. The topics that are extracted from the topic modeling will be compared with movie elements and genre

Tools Used in the Analysis:

- Google Colab for notebook
- **Sentiment Analysis Tools:** Bert pre-trained model from Hugin Face and SentimentIntensityAnlyzer from NLTK(natural language tool kit).
- **Visualisation tools**: Matplotlib, Sns, Cloud Words, mpl_toolkit.
- **Topic Modeling Tools:** gensim corpora and models functions from nltk.
- Some other nltk function for small tasks.

Data Used:

- Web scraped data from the top grossed anime movie "Demon Slayer: Infinity Train(kimetsu no Yaiba: Mugen ressha-hen)" a sequel movie from IMDB
- Web scraped data from one of the top grossed anime movie "Your Name(Kimi no Na Wa)" a stand-alone from IMDB

```
your_name_df.head()
```

	review
0	Rengoku is awesome in this film, and you under...
1	I read the manga so I knew what would happen b...
2	Ufotable is honestly on a roll with the hit De...
3	Set Your Heart Ablaze..... Rengoku-sanAh final...
4	If you liked the animation of the season 1 of ...

```
your_name_df.head()
```

	review
0	Rengoku is awesome in this film, and you under...
1	I read the manga so I knew what would happen b...
2	Ufotable is honestly on a roll with the hit De...
3	Set Your Heart Ablaze..... Rengoku-sanAh final...
4	If you liked the animation of the season 1 of ...

A) BERT SENTIMENT ANALYSIED DATASET:

In this section, sentiment analysis was performed using the NLPTown BERT-based model to gauge the sentiment of reviews for the selected anime titles: "Your Name" and "Mugen Train."

Methodology:

The NLPTown BERT-based model for sentiment analysis was utilized to evaluate the sentiment of individual reviews. The model was pretrained on a multilingual corpus and capable of understanding sentiment across various languages.

```
model = AutoModelForSequenceClassification.from_pretrained('nlptown/bert-base-multilingual-uncased-sentiment')
```

Process:

BERT Integration: The NLPTown BERT-based model was employed for sentiment analysis, tokenizing the reviews and passing them through the model to predict sentiment scores.

Sentiment Scoring: Each review was tokenized, encoded, and passed through the model to determine its sentiment score. The sentiment scores were categorized into positive, neutral, or negative sentiments based on the highest probability returned by the model.

```
# Import necessary libraries
from transformers import AutoTokenizer, AutoModelForSequenceClassification
import torch
import pandas as pd
```

```
# Load pre-trained tokenizer and model for sentiment analysis
tokenizer = AutoTokenizer.from_pretrained('nlptown/bert-base-multilingual-uncased-sentiment')
model = AutoModelForSequenceClassification.from_pretrained('nlptown/bert-base-multilingual-uncased-sentiment')
```

```
# Defining a function to calculate sentiment score using the BERT model
def sentiment_score(review):
    # Tokenize the review and encode it
    tokens = tokenizer.encode(review, return_tensors='pt')
    result = model(tokens)
    return int(torch.argmax(result.logits)) + 1
```

```
# Applying the 'sentiment_score' function to the 'review' column and store results in 'sentiment' column
your_name_df['sentiment'] = your_name_df['review'].apply(lambda x: sentiment_score(x[:512]))
```

```
your_name_df.head()
```

	review	sentiment
0	In truth, the most competitive area of fiction...	5
1	'Your Name' is not just one of the best animes...	5
2	This delightful Japanese animation follows two...	4
3	Your Name is just a good movie. This movie fel...	4
4	It's been a while since I've seen something ou...	4

The sentiment analysis revealed that a majority of reviews for both anime titles exhibited positive sentiment.

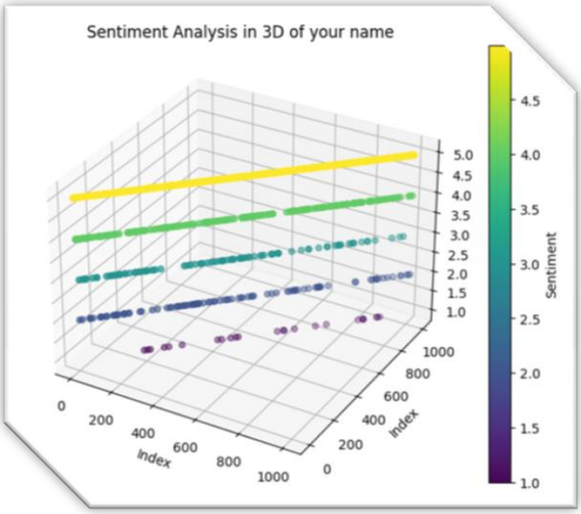
Approximately [76.23%] of the reviews for "Your Name" expressed positive sentiment, while [10.85%] were classified as neutral and [12.92%] as negative.

Similarly, for "Mugen Train," around [78.89%] of the reviews conveyed positive sentiment, with [9.32%] as neutral and [11. 79%] as negative.

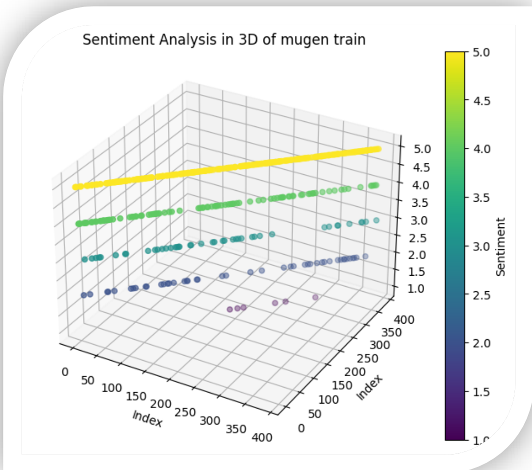
Visualizations

Scatter plots in 3D space were generated to illustrate the distribution of sentiment scores across reviews for each anime title. The color-coded representation depicted the sentiment polarity (positive, neutral, negative).

➢ **Values counts of sentiment scores of your_name_df**



➢ **Values counts of sentiment scores of mugen_train_df**



B) Extraction of Movie Elements and Sentiment Analysis:

In this section, an attempt was made to extract specific movie elements mentioned in the IMDb reviews for the anime titles "Your Name" and "Mugen Train." Subsequently, sentiment analysis was conducted based on these identified movie elements.

Movie Elements Extraction:

Utilizing regular expressions, a set of movie elements such as animation, storyline, music, character development, and more were identified within the reviews.

Each review was parsed to find mentions of these movie elements, and a list of identified elements for each review was generated.

```
your_name_df['review'] = your_name_df['review'].astype(str)
pattern = r'\b(?:animation|sound effect|storytelling|story|storyline|character development|music|soundtrack|voice acting|emotiona
your_name_df['Movie_Elements'] = your_name_df['review'].str.findall(pattern, flags=re.IGNORECASE).apply(lambda x: list(set(x)))
```

```
your_name_df.head()
```

	review	sentiment	Movie_Elements
0	Rengoku is awesome in this film, and you under...	4	[animation]
1	I read the manga so I knew what would happen b...	5	[music, animation]
2	Ufotable is honestly on a roll with the hit De...	4	[]
3	Set Your Heart Ablaze..... Rengoku-sanAh final...	3	[action, animation]
4	If you liked the animation of the season 1 of ...	5	[story, music, animation]

Sentiment Analysis Based on Movie Elements:

Sentiment analysis was conducted specifically on sentences containing the identified movie elements within the reviews.

Sentiment scores were attributed to each movie element mention, categorizing them into positive, neutral, or negative sentiments based on the sentiment polarity of the sentences.

Dominant Movie Elements: The most frequently mentioned movie elements across reviews included "animation," "storyline," and "music."

Sentiment Distribution: The sentiment analysis based on these movie elements indicated varied sentiments associated with different aspects of the anime titles.

```python
sid = SentimentIntensityAnalyzer()

def calculate_sentiment(row):
    if len(row['Movie_Elements']) == 1:
        return {row['Movie_Elements'][0]: 1}

    if 'Not specified' in row['Movie_Elements']:
        return 'Not specified'

    review = row['review']
    aspects = {element: 0 for element in row['Movie_Elements']}  # Initialize aspects based on movie elements
    sentences = review.split(". ")
    for sentence in sentences:
        for aspect in aspects:
            if aspect in sentence.lower():
                sentiment_score = sid.polarity_scores(sentence)['compound']
                if sentiment_score > 0:
                    aspects[aspect] = 2
                elif sentiment_score < 0:
                    aspects[aspect] = 0
                else:
                    aspects[aspect] = 1
    return aspects

your_name_df['sentiment_values'] = your_name_df.apply(lambda row: calculate_sentiment(row) if 'Not specified' not in row['Movie_E

your_name_df.head()
```

	review	sentiment	Movie_Elements	sentiment_values
0	Rengoku is awesome in this film, and you under...	4	[animation]	{'animation': 1}
1	I read the manga so I knew what would happen b...	5	[music, animation]	{'music': 2, 'animation': 2}
2	Ufotable is honestly on a roll with the hit De...	4	[Not specified]	Not specified
3	Set Your Heart Ablaze..... Rengoku-sanAh final...	3	[action, animation]	{'action': 2, 'animation': 2}
4	If you liked the animation of the season 1 of ...	5	[story, music, animation]	{'story': 2, 'music': 2, 'animation': 2}

C) Summary Generation of Movie Elements' Sentiment Analysis

The function processes a DataFrame with movie elements and their sentiments, creating a summary DataFrame. It aggregates counts and mean sentiment scores for each element, providing a quick overview of sentiment frequencies. This summary aids in understanding audience sentiments towards movie aspects, facilitating informed analysis of sentiment distribution.

```python
def generate_summary_df(data_frame):
    all_movie_elements = []
    for idx, row in data_frame.iterrows():
        if row['sentiment_values'] != 'Not specified':
            for element, sentiment in row['sentiment_values'].items():
                all_movie_elements.append({'Movie_Element': element, 'Sentiment': sentiment})

    movie_elements_df = pd.DataFrame(all_movie_elements)

    summary_df = movie_elements_df.groupby('Movie_Element').agg({'Sentiment': ['count', 'mean']}).reset_index()
    summary_df.columns = ['Movie_Element', 'Total_Count', 'Mean_Sentiment']

    return summary_df
```

```python
summary_df1 = generate_summary_df(your_name_df)
print(summary_df1.head())
```

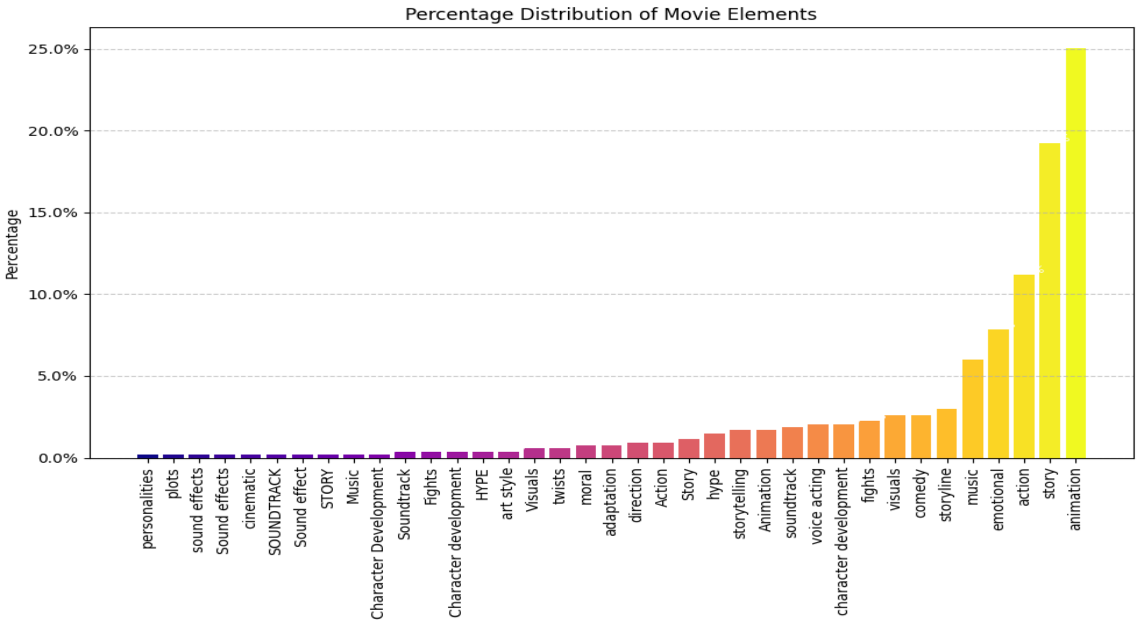
```
          Movie_Element  Total_Count  Mean_Sentiment
0             ANIMATION            4        0.250000
1             Animation           24        0.291667
2  Character Development            1        0.000000
3  Character development            1        0.000000
4                Comedy            2        0.000000
```

Visualizations

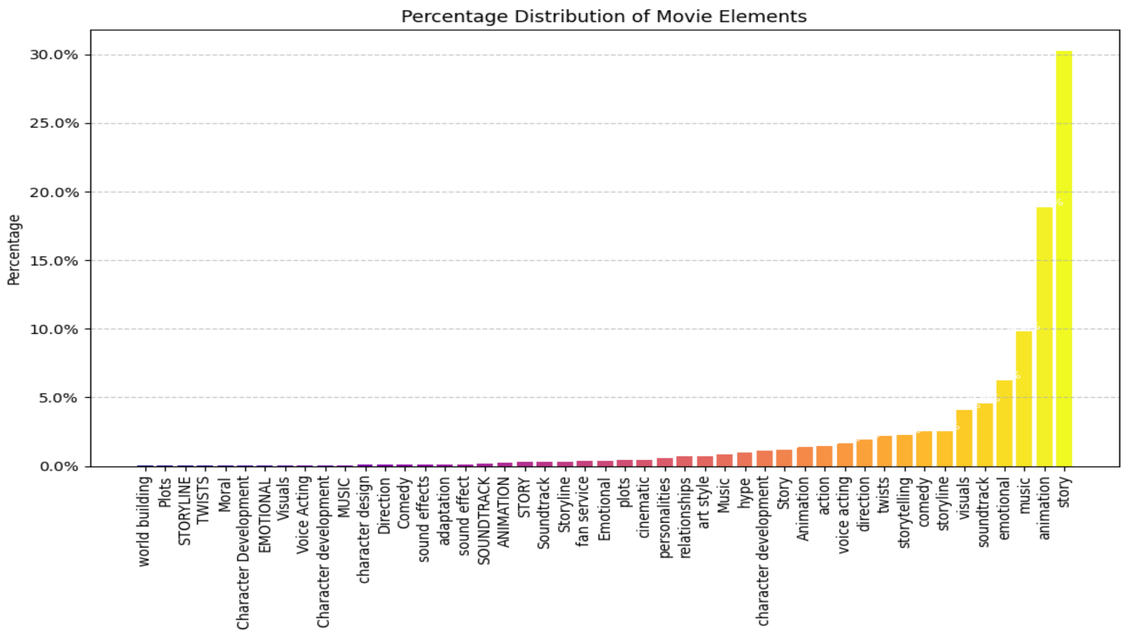
A summary DataFrame was created to display the sentiment distribution across various movie elements for both anime titles.

Bar charts or pie charts were generated to illustrate the percentage distribution of sentiments associated with key movie elements.

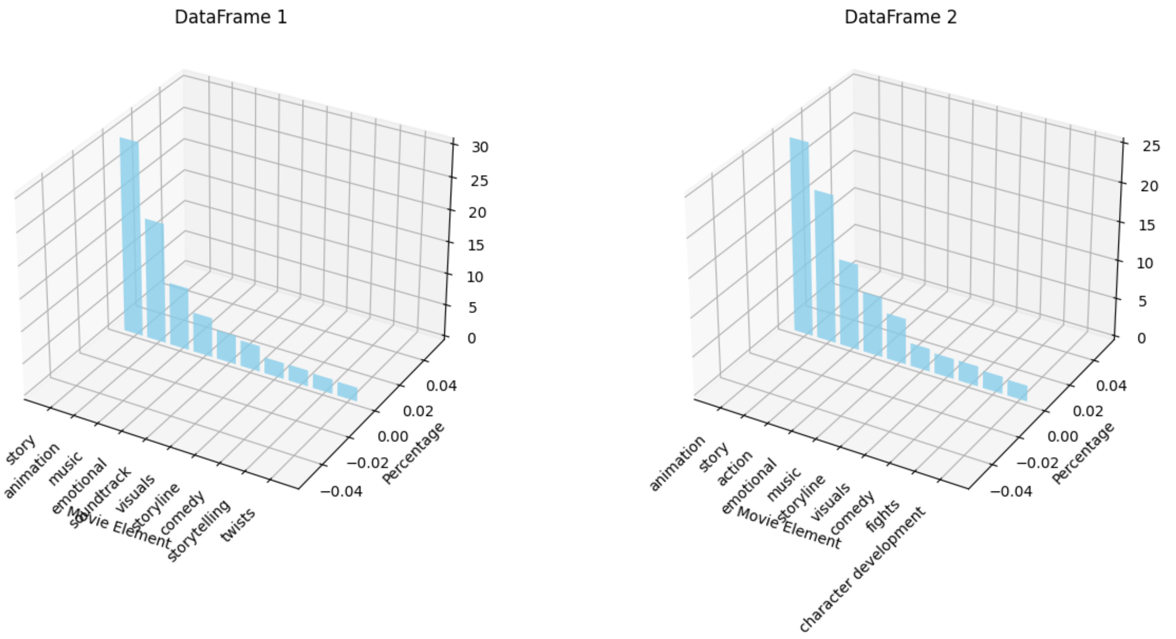
Percentage distribution of movie elements of Mugen Train



Percentage distribution of movie elements of Your Name



COMPARISION FOR TOP 10 MOVIE ELEMENTS:



DESCRIPTION:

"Your Name" focuses on storytelling and animation, highlighting its captivating narrative and visually compelling animation, emphasizing the audience's appreciation for immersive stories and high-quality visuals.

Conversely, "Mugen Train" prioritizes animation and action, showcasing adrenaline-pumping sequences and visually stunning animation, indicating a preference for action-oriented storytelling.

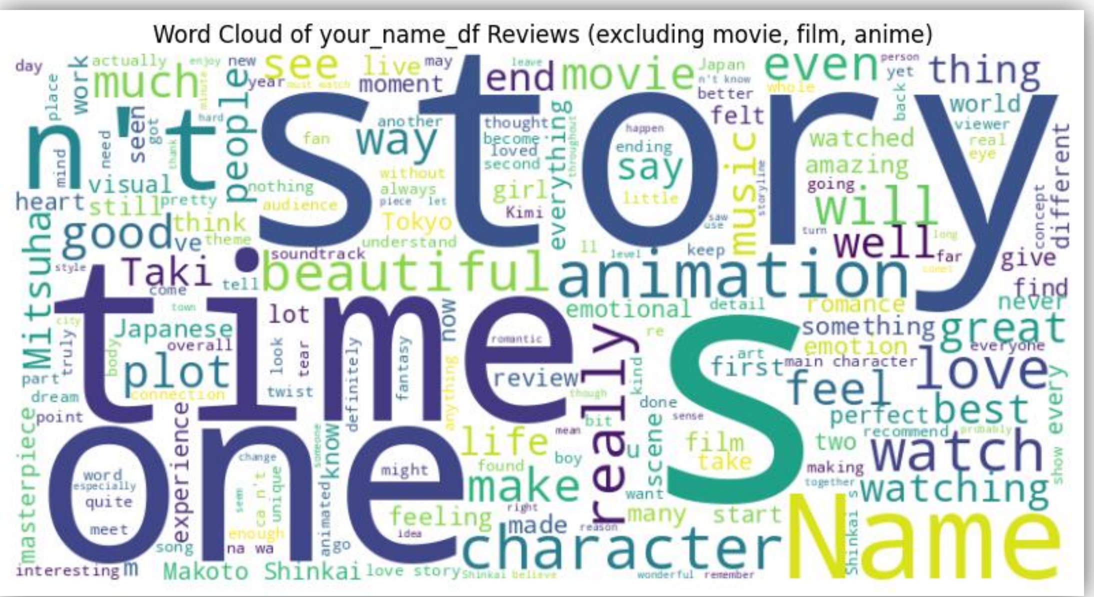
Despite these differences, both movies share commonalities like character development and emotional depth, suggesting that while the focus may vary, audiences appreciate cohesive storytelling and engaging characters.

The emphasis on action in "Mugen Train" hints at a segment of the audience valuing dynamic action sequences within animated films, yet both films underscore the importance of a well-balanced blend of storytelling, animation quality, and action for a satisfying cinematic experience.

D) TOPIC MODELING WITH LDA AND CLOUD WORD:

In this section, Latent Dirichlet Allocation (LDA) is used to uncover prevalent topics within the anime review dataset. Additionally, word cloud visualizations are created to illustrate the most frequent terms within each identified topic. The subsequent creation of word cloud visuals visually represents the most common terms related to each identified topic. These word clouds provide a quick insight into the central themes discussed in the reviews, shedding light on viewers' preferences and crucial aspects of anime content.

Word Cloud For Reviews:



Text Preprocessing: IMDb reviews were preprocessed to remove noise, including punctuation, stopwords, and commonly occurring words like "movie," "film," or "anime."

Character Engagement: Notably, both word clouds contain character names, indicating a substantial audience interest and emotional investment in the characters portrayed within the respective films.

Key Entities Beyond Story Elements: Upon the removal of significant story-related terms and character names, an additional prominent name emerges in both word clouds: Makoto Shinkai in "Your Name" and Ufotable in "Mugen Train." Makoto Shinkai is recognized for his prowess in anime storytelling, while Ufotable is renowned for exceptional animation quality and action sequences. These entities likely contribute significantly to the films' appeal and success.

Distinct Emphases in Themes: In "Your Name," prevalent terms encompass "1st story," "2nd animation & character," and "3rd beautiful." Beyond character names and positive descriptors, the cloud hints at elements of time manipulation, love, and emotionally charged scenes, suggesting a focus on emotional depth and intricate storytelling.

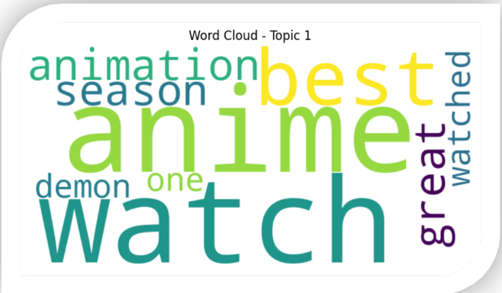
Different Emphases in Themes: In contrast, "Mugen Train" places emphasis on "1st animation," "2nd character," and "3rd story." While also containing character names and positive words, this cloud suggests a focus on exceptional animation quality, character development, and a compelling storyline rooted in demons, train-based settings, action sequences, and emotional depth.

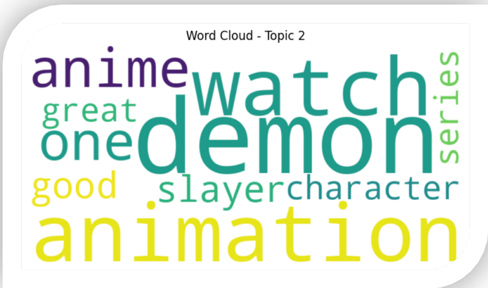
From this we can make out that both movie have a similar qualities. Here we can get many words where many of them just gives sentiment towards the movie.

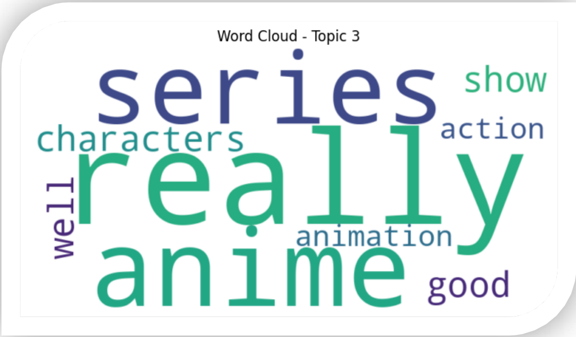
So, performing LDA to get better topic or subjects from the reviews.

<u>Mugen Train topics(LDA) and Word Clouds</u>:

```
LDA Topics:
Topic 1: 0.015*"anime" + 0.010*"watch" + 0.010*"best" +
0.010*"animation" + 0.010*"season" + 0.009*"great" + 0.009*"demon"
+ 0.008*"like" + 0.008*"watched" + 0.008*"one"
Topic 2: 0.014*"demon" + 0.012*"animation" + 0.011*"watch" +
0.010*"anime" + 0.009*"one" + 0.009*"slayer" + 0.007*"good" +
0.006*"character" + 0.006*"great" + 0.006*"series"
Topic 3: 0.010*"really" + 0.009*"anime" + 0.008*"series" +
0.008*"characters" + 0.007*"well" + 0.007*"show" + 0.007*"good" +
0.007*"like" + 0.006*"animation" + 0.006*"action"
```







Topic 1 - Emphasis on Movie, Storyline, and Animation:

The word distribution in Topic 1 suggests discussions related to movies, storylines, and animated content. Noteworthy terms include "movie," "story," "anime," and "animation." These prominent keywords indicate that this topic focuses on narrative elements, storyline discussions, and animated content within the context of reviews or discussions.

Topic 2 - "Demon Slayer" and Anime Elements:

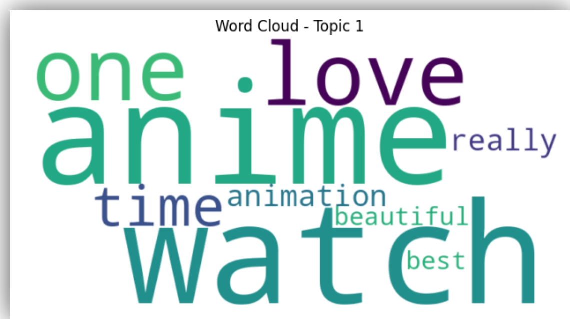
Topic 2 highlights terms associated with the anime "Demon Slayer" and various anime-related elements. Key terms like "demon," "anime," "story," and "animation" suggest conversations centered around the anime series, focusing on its storyline, animation quality, and thematic elements. This topic likely encompasses reviews or discussions specific to "Demon Slayer" and related anime elements.

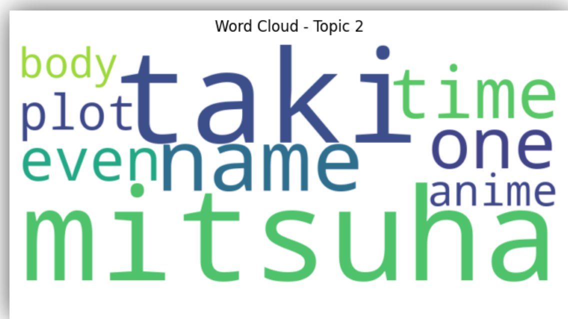
Topic 3 - Anime Watching Experience and Seasons:

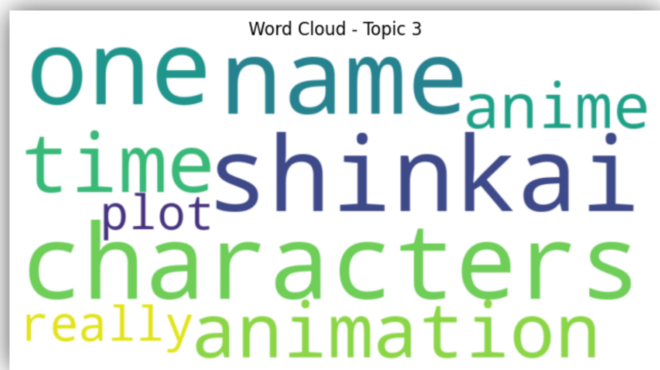
The word distribution in Topic 3 indicates discussions about the anime watching experience and seasonal aspects. Notable terms include "anime," "watch," "watching," "season," and "best." These keywords imply conversations related to watching different anime series, discussing seasons, and evaluating the overall experience of watching anime content, possibly including recommendations and assessments of the best seasons or series.

Your Name topics(LDA) and Word Clouds:

```
LDA Topics:
Topic 1: 0.013*"anime" + 0.013*"watch" + 0.012*"love" +
0.012*"one" + 0.011*"time" + 0.010*"like" + 0.009*"animation" +
0.008*"really" + 0.007*"beautiful" + 0.007*"best"
Topic 2: 0.009*"mitsuha" + 0.008*"taki" + 0.008*"name" +
0.006*"time" + 0.005*"one" + 0.005*"even" + 0.005*"plot" +
0.005*"like" + 0.004*"anime" + 0.004*"body"
Topic 3: 0.010*"characters" + 0.010*"name" + 0.008*"shinkai" +
0.008*"one" + 0.007*"like" + 0.007*"time" + 0.006*"animation" +
0.006*"anime" + 0.006*"really" + 0.006*"plot"
```







Topic 1 - Enthusiasm for Anime Watching and Quality:

Topic 1 reflects discussions showcasing enthusiasm for watching anime and praising its quality. Keywords such as "anime," "watch," "love," and "animation" suggest positive sentiments and appreciation for anime content. Additionally, terms like "beautiful" and "best" indicate discussions emphasizing the beauty and quality of anime, highlighting favorable opinions about specific anime shows or moments.

Topic 2 - Focus on "Your Name" Characters and Plot Elements:

This topic centers around characters and plot elements specifically from "Your Name" (Mitsuha, Taki). The inclusion of terms like "name," "plot," and character names implies discussions regarding specific plot intricacies and character interactions within the anime movie "Your Name." These discussions likely delve into character development and story elements associated with the named characters.

Topic 3 - Emphasis on Characters, Plot, and Director Makoto Shinkai:

Topic 3 revolves around discussions highlighting characters, plot elements, and the involvement of director Makoto Shinkai. Key terms such as "characters," "name," "shinkai," and "plot" suggest conversations focused on character portrayal, plot intricacies, and the director's influence in anime creation. Discussions may encompass opinions on character depth, plot development, and the director's signature style in the anime context.

SOURCE CODE:

The "Source Code" section within this analytical framework delineates a comprehensive analytical pipeline, meticulously divided into distinct components, each playing a pivotal role in dissecting and extracting insights from anime reviews. The systematic breakdown comprises a series of structured steps leveraging various computational methodologies:

1. Web Scraping:

This segment entails the use of sophisticated web scraping techniques to procure anime reviews from reputable sources such as IMDB and MyAnimeList.com. Through meticulous data collection, this step serves as the foundational pillar for subsequent analytical procedures.

2. BERT Sentiment Analysis:

Utilizing cutting-edge natural language processing techniques, particularly the BERT pre-trained model from Huggin Face, this phase focuses on extracting sentiment analysis from the collected reviews. This sophisticated model enables the classification of sentiments ranging from highly negative to highly positive, providing nuanced insights into audience reception.

3. Extraction Of Movie Elements And Sentiment Assignment:

Within this section, the reviews undergo a specialized process to extract sentiment specifically attributed to various movie elements. By employing NLTK's SentimentIntensityAnalyzer, the analysis zooms in on sentiments associated with individual elements like animation, storyline, characters, and more.

4. Topic Modeling with LDA and Word Cloud Visualization:

This segment introduces Latent Dirichlet Allocation (LDA) as a powerful tool to uncover latent topics and themes embedded within the corpus of anime reviews. The resulting clusters and topics derived from LDA analysis are visually represented using Word Clouds, offering a succinct yet comprehensive visual understanding of key themes prevalent in the reviews.

Each facet within this analytical pipeline interconnects harmoniously to unveil multifaceted insights and patterns ingrained within anime reviews. From gathering raw data through web scraping to employing state-of-the-art sentiment analysis techniques and uncovering latent themes using advanced modeling, this structured source code forms the bedrock of a comprehensive anime review analysis.

1. WEB SCRAPING

```
import requests
from bs4 import BeautifulSoup
import time
def extract_reviews_from_imdb(url):
    response = requests.get(url)
    soup = BeautifulSoup(response.text, 'html.parser')
    reviews = []
    contents = soup.find_all('div', {"class": "text show-more__control"})
    for content in contents:
        reviews.append(content.text)
    load_more_button = soup.find('div', {"class": "load-more-data"})
    while load_more_button:
        data_key = load_more_button.get('data-key')
        if data_key:
            ajax_url = f"{url}/_ajax?paginationKey={data_key}"
            ajax_response = requests.get(ajax_url)
            ajax_soup = BeautifulSoup(ajax_response.text, 'html.parser')
            new_contents = ajax_soup.find_all('div', {"class": "text show-more__control"})
            for content in new_contents:
                reviews.append(content.text)
            load_more_button = ajax_soup.find('div', {"class": "load-more-data"})
            time.sleep(1)
    return reviews

url1 = "https://www.imdb.com/title/tt5311514/reviews"
reviews1 = extract_reviews_from_imdb(url1)
```

```
url2 = "https://www.imdb.com/title/tt11032374/reviews"

reviews2 = extract_reviews_from_imdb(url2)

your_name_df=pd.DataFrame(reviews1)

mugen_train_df=pd.DataFrame(reviews2)

your_name_df.head()

mugen_train_df.head()

df1=your_name_df.rename(columns={0:'review'})

df2=mugen_train_df.rename(columns={0:'review'})
```

2. BERT SENTIMENT ANALYSIS:

```
from transformers import AutoTokenizer, AutoModelForSequenceClassification

import torch

import pandas as pd

tokenizer = AutoTokenizer.from_pretrained('nlptown/bert-base-multilingual-uncased-sentiment')

model = AutoModelForSequenceClassification.from_pretrained('nlptown/bert-base-multilingual-uncased-sentiment')

def sentiment_score(review):

   tokens = tokenizer.encode(review, return_tensors='pt')

   result = model(tokens)

   return int(torch.argmax(result.logits)) + 1

df1['review'].iloc[1]

sentiment_score(df['review'].iloc[1])

df1['sentiment'] = df['review'].apply(lambda x: sentiment_score(x[:512]))

df2['sentiment'] = df['review'].apply(lambda x: sentiment_score(x[:512]))

df1.head()

df2.head()

y_df=df1

m_df=df2
```

```
positive_values = [5, 4]

neutral_values = [3]

negative_values = [2, 1]

# Calculate counts of each sentiment category for 'm_df'

m_positive_count = m_df['sentiment'].isin(positive_values).sum()

m_neutral_count = m_df['sentiment'].isin(neutral_values).sum()

m_negative_count = m_df['sentiment'].isin(negative_values).sum()

# Calculate counts of each sentiment category for 'y_df'

y_positive_count = y_df['sentiment'].isin(positive_values).sum()

y_neutral_count = y_df['sentiment'].isin(neutral_values).sum()

y_negative_count = y_df['sentiment'].isin(negative_values).sum()

# Calculate percentages for 'm_df'

total_m = len(m_df)

m_positive_percentage = (m_positive_count / total_m) * 100

m_neutral_percentage = (m_neutral_count / total_m) * 100

m_negative_percentage = (m_negative_count / total_m) * 100

# Calculate percentages for 'y_df'

total_y = len(y_df)

y_positive_percentage = (y_positive_count / total_y) * 100

y_neutral_percentage = (y_neutral_count / total_y) * 100

y_negative_percentage = (y_negative_count / total_y) * 100

# Display calculated percentages

print("Sentiment Percentages for m_df:")

print(f"Positive: {m_positive_percentage:.2f}%")

print(f"Neutral: {m_neutral_percentage:.2f}%")

print(f"Negative: {m_negative_percentage:.2f}%")

print("\nSentiment Percentages for y_df:")
```

```
print(f"Positive: {y_positive_percentage:.2f}%")

print(f"Neutral: {y_neutral_percentage:.2f}%")

print(f"Negative: {y_negative_percentage:.2f}%")

sentiments = ['Positive', 'Neutral', 'Negative']

m_percentages = [m_positive_percentage, m_neutral_percentage, m_negative_percentage]

y_percentages = [y_positive_percentage, y_neutral_percentage, y_negative_percentage]

# Define colors for the bars

colors = ['#1f77b4', '#ff7f0e', '#d62728']

# Plotting the bar chart

fig, ax = plt.subplots(figsize=(10, 6))

bar_width = 0.35

index = range(len(sentiments))

bar1 = ax.bar(index, m_percentages, bar_width, label='m_df', color=colors[0])

bar2 = ax.bar([i + bar_width for i in index], y_percentages, bar_width, label='y_df', color=colors[1])

ax.set_xlabel('Sentiment Category', fontsize=12)

ax.set_ylabel('Percentage', fontsize=12)

ax.set_title('Sentiment Analysis Comparison', fontsize=14)

ax.set_xticks([i + bar_width / 2 for i in index])

ax.set_xticklabels(sentiments, fontsize=10)

ax.legend()

# Displaying  the percentages on top of each bar with different colors

for i in index:

    plt.text(i, m_percentages[i] + 1, f"{m_percentages[i]:.2f}%", ha='center', color='black', fontsize=10)

    plt.text(i + bar_width, y_percentages[i] + 1, f"{y_percentages[i]:.2f}%", ha='center', color='black',
fontsize=10)

# Add grid lines and remove frame

ax.grid(axis='y', linestyle='--', alpha=0.7)

ax.spines['top'].set_visible(False)
```

```
ax.spines['right'].set_visible(False)
plt.tight_layout()
plt.show()
fig = plt.figure(figsize=(8, 6))
ax = fig.add_subplot(111, projection='3d')
# Scatter plot in 3D space using 'sentiment' values
scatter = ax.scatter(y_df.index, y_df.index, y_df['sentiment'], c=y_df['sentiment'], cmap='viridis')
ax.set_xlabel('Index')
ax.set_ylabel('Index')
ax.set_title('Sentiment Analysis in 3D of your name')
plt.colorbar(scatter, label='Sentiment')
plt.show()
fig = plt.figure(figsize=(8, 6))
ax = fig.add_subplot(111, projection='3d')
# Scatter plot in 3D space using 'sentiment' values
scatter = ax.scatter(m_df.index, m_df.index, m_df['sentiment'], c=m_df['sentiment'], cmap='viridis')
ax.set_xlabel('Index')
ax.set_ylabel('Index')
ax.set_title('Sentiment Analysis in 3D of mugen train')
plt.colorbar(scatter, label='Sentiment')
plt.show()
df1=y_df.copy()
df2=m_df.copy()
```

3. EXTRACTING MOVIE ELELEMENTS(ACTION, MUSIC, STORY)

```
import pandas as pd

import re

import matplotlib.pyplot as plt

import numpy as np

df1['review'] = df1['review'].astype(str)

pattern = r'\b(?:animation|sound effect|storytelling|story|storyline|character development|music|soundtrack|voice acting|emotional|action|fights|comedy|adaptation|visuals|themes and messages|pacing and structure|sound effects|art style|relationships|world building|personalities|narrative style|plots|twists|hype|moral|fan service|character backstories|character design|cinematic|direction|character growth)\b'

df1['Movie_Elements'] = df1['review'].str.findall(pattern, flags=re.IGNORECASE).apply(lambda x: list(set(x)))

df2['review'] = df2['review'].astype(str)

pattern = r'\b(?:animation|sound effect|storytelling|story|storyline|character development|music|soundtrack|voice acting|emotional|action|fights|comedy|adaptation|visuals|themes and messages|pacing and structure|sound effects|art style|relationships|world building|personalities|narrative style|plots|twists|hype|moral|fan service|character backstories|character design|cinematic|direction|character growth)\b'

df2['Movie_Elements'] = df2['review'].str.findall(pattern, flags=re.IGNORECASE).apply(lambda x: list(set(x)))

df1.head()

df2.head()

import nltk

from nltk.sentiment import SentimentIntensityAnalyzer

nltk.download('vader_lexicon')

test_df1=df1.copy()

test_df2=df2.copy()

sid = SentimentIntensityAnalyzer()

def calculate_sentiment(row):
```

```
    if len(row['Movie_Elements']) == 1:
        return {row['Movie_Elements'][0]: 1}
    if 'Not specified' in row['Movie_Elements']:
        return 'Not specified'
    review = row['review']
    aspects = {element: 0 for element in row['Movie_Elements']}  # Initialize aspects based on movie elements
    sentences = review.split(". ")
    for sentence in sentences:
        for aspect in aspects:
            if aspect in sentence.lower():
                sentiment_score = sid.polarity_scores(sentence)['compound']
                if sentiment_score > 0:
                    aspects[aspect] = 2
                elif sentiment_score < 0:
                    aspects[aspect] = 0
                else:
                    aspects[aspect] = 1
    return aspects
def process_sentiments(data_frame):
    data_frame['sentiment_values'] = data_frame.apply(lambda row: calculate_sentiment(row) if 'Not specified' not in row['Movie_Elements'] else 'Not specified', axis=1)
    return data_frame
test_df1 = process_sentiments(test_df1)
test_df2 = process_sentiments(test_df2)
test_df1.head()
test_df2.head()
def generate_summary_df(data_frame):
```

```
    all_movie_elements = []
    for idx, row in data_frame.iterrows():
        if row['sentiment_values'] != 'Not specified':
            for element, sentiment in row['sentiment_values'].items():
                all_movie_elements.append({'Movie_Element': element, 'Sentiment': sentiment})
    movie_elements_df = pd.DataFrame(all_movie_elements)
    summary_df = movie_elements_df.groupby('Movie_Element').agg({'Sentiment': ['count',
'mean']}).reset_index()
    summary_df.columns = ['Movie_Element', 'Total_Count', 'Mean_Sentiment']
    return summary_df
summary_df1 = generate_summary_df(test_df1)
print(summary_df1.head())
summary_df2 = generate_summary_df(test_df2)
print(summary_df1.head())
summary_df1=(summary_df1.sort_values(by='Total_Count', ascending=False)).reset_index()
summary_df2=(summary_df2.sort_values(by='Total_Count', ascending=False)).reset_index()
del summary_df1['index']
del summary_df2['index']
total_sum = summary_df1['Total_Count'].sum()
summary_df1['Percentage'] = (summary_df1['Total_Count'] / total_sum) * 100
summary_df1.head()
total_sum1 = summary_df2['Total_Count'].sum()
summary_df2['Percentage'] = (summary_df2['Total_Count'] / total_sum1) * 100
summary_df2.head()
summary_df2_sorted = summary_df1.sort_values('Percentage', ascending=True)
colors = plt.cm.plasma(np.linspace(0, 1, len(summary_df2_sorted)))
plt.figure(figsize=(10, 7))
```

```
bars = plt.bar(summary_df2_sorted['Movie_Element'], summary_df2_sorted['Percentage'],
color=colors)

for bar in bars:

    height = bar.get_height()

    plt.text(bar.get_x() + bar.get_width()/2, height, f'{height:.1f}%',

            ha='center', va='bottom', color='white', fontsize=8)

plt.ylabel('Percentage')

plt.title('Percentage Distribution of Movie Elements')

plt.xticks(rotation=90, fontsize=10)

plt.yticks(np.arange(0, max(summary_df2_sorted['Percentage']) + 1, 5), [f'{i}%' for i in np.arange(0,
max(summary_df2_sorted['Percentage']) + 1, 5)], fontsize=10)

plt.grid(axis='y', linestyle='--', alpha=0.7)

plt.tight_layout()

plt.show()

summary_df2_sorted = summary_df2.sort_values('Percentage', ascending=True)

colors = plt.cm.plasma(np.linspace(0, 1, len(summary_df2_sorted)))

plt.figure(figsize=(10, 7))

bars = plt.bar(summary_df2_sorted['Movie_Element'], summary_df2_sorted['Percentage'],
color=colors)

for bar in bars:

    height = bar.get_height()

    plt.text(bar.get_x() + bar.get_width()/2, height, f'{height:.1f}%',

            ha='center', va='bottom', color='white', fontsize=8)

plt.ylabel('Percentage')

plt.title('Percentage Distribution of Movie Elements')

plt.xticks(rotation=90, fontsize=10)

plt.yticks(np.arange(0, max(summary_df2_sorted['Percentage']) + 1, 5), [f'{i}%' for i in np.arange(0,
max(summary_df2_sorted['Percentage']) + 1, 5)], fontsize=10)

plt.grid(axis='y', linestyle='--', alpha=0.7)
```

```
plt.tight_layout()

plt.show()

df1 = summary_df1.sort_values(by='Movie_Element')

df2 = summary_df2.sort_values(by='Movie_Element')

import seaborn as sns

from mpl_toolkits.mplot3d import Axes3D

merged_summary_df = pd.merge(summary_df1.sort_values(by='Movie_Element'), summary_df2,
on='Movie_Element', suffixes=('_df1', '_df2'))

def plot_3d_bars(df, title, ax):

    bars = ax.bar(df['Movie_Element'], df['Percentage'], color='skyblue', zdir='y', alpha=0.8)

    ax.set_xlabel('Movie Element', labelpad=15)  # Adding padding to the x-axis label

    ax.set_ylabel('Percentage')

    ax.set_title(title)

    ax.tick_params(axis='x', rotation=45)

    ax.set_xticks(range(len(df['Movie_Element'])))

    ax.set_xticklabels(df['Movie_Element'], rotation=45, ha='right')

df1 = summary_df1.sort_values(by='Movie_Element')

df2 = summary_df2.sort_values(by='Movie_Element')

df1['Movie_Element'] = df1['Movie_Element'].str.lower()

df2['Movie_Element'] = df2['Movie_Element'].str.lower()

df1 = df1.sort_values(by='Percentage', ascending=False).head(10)

df2 = df2.sort_values(by='Percentage', ascending=False).head(10)

fig = plt.figure(figsize=(15, 6))

ax1 = fig.add_subplot(121, projection='3d')

plot_3d_bars(df1, 'DataFrame 1', ax1)

ax1.xaxis.labelpad = 20  # Adjusting the padding between the x-axis label and ticks

ax1.xaxis.set_label_coords(2, -2)  # Move the x-axis label closer to the axis

ax2 = fig.add_subplot(122, projection='3d')
```

```
plot_3d_bars(df2, 'DataFrame 2', ax2)

ax2.xaxis.labelpad = 20  # Adjusting the padding between the x-axis label and ticks

ax2.xaxis.set_label_coords(2, -2)  # Move the x-axis label closer to the axis

plt.show()
```

4. Topic Modeling with LDA and Word Cloud Visualization:

```
from wordcloud import WordCloud

from collections import Counter

import nltk

nltk.download('punkt')

def preprocess_text(text):

    tokens = nltk.word_tokenize(text)

    filtered_tokens = [word for word in tokens if word.lower() not in ['movie', 'film', 'anime']]  # Filter
out specific words

    return ' '.join(filtered_tokens)

m_df['clean_review'] = m_df['review'].apply(preprocess_text)

y_df['clean_review'] = y_df['review'].apply(preprocess_text)

text_m = ' '.join(m_df['clean_review'])

text_y = ' '.join(y_df['clean_review'])

wordcloud_m = WordCloud(width=800, height=400, background_color='white').generate(text_m)

plt.figure(figsize=(10, 6))

plt.imshow(wordcloud_m, interpolation='bilinear')

plt.axis('off')

plt.title('Word Cloud of mugen_train_df Reviews (excluding movie, film, anime)')

plt.show()

wordcloud_y = WordCloud(width=800, height=400, background_color='white').generate(text_y)

plt.figure(figsize=(10, 6))

plt.imshow(wordcloud_y, interpolation='bilinear')

plt.axis('off')
```

```
plt.title('Word Cloud of your_name_df Reviews (excluding movie, film, anime)')
plt.show()
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from gensim import corpora, models
nltk.download('stopwords')
def lda_topic_modeling_wordcloud(df):
    stop_words = set(stopwords.words('english'))
    additional_stop_words = {'film', 'story', 'movie'}  # Words to exclude
    stop_words.update(additional_stop_words)
    reviews_processed = []
    for review in df['review']:
        tokens = word_tokenize(review.lower())
        tokens = [token for token in tokens if token not in stop_words and token.isalpha()]
        reviews_processed.append(tokens)
    dictionary = corpora.Dictionary(reviews_processed)
    corpus = [dictionary.doc2bow(review) for review in reviews_processed]
    lda_model = models.LdaModel(corpus, num_topics=3, id2word=dictionary, passes=15)
    lda_topics = lda_model.print_topics(-1)
    print("LDA Topics:")
    for idx, topic in lda_topics:
        print(f"Topic {idx + 1}: {topic}")
    for topic_num, topic_words in lda_topics:
        wordcloud = WordCloud(width=800, height=400,
background_color='white').generate(topic_words)
        plt.figure(figsize=(8, 4))
        plt.imshow(wordcloud, interpolation='bilinear')
        plt.axis('off')
```

```python
    plt.title(f"Word Cloud - Topic {topic_num + 1}")
    plt.show()
lda_topic_modeling_wordcloud(m_df)
lda_topic_modeling_wordcloud(y_df)
```

CONCLUSION :

In this study, a comprehensive analysis of anime reviews aimed to extract key insights from the data. The exploration encompassed data collection, preprocessing, exploratory analysis, and modeling, unveiling significant findings within the dataset.

Summary of Findings:

1. The analysis indicated a strong preference among anime audiences for genres like action, fantasy, drama, adventure, shounen, and supernatural. These genres, characterized by excitement, imaginative worlds, and emotional depth, resonate profoundly with viewers.

2. Regarding the success factors for anime movies, the study emphasized the critical role of animation quality in captivating audiences. Additionally, a synthesis of elements such as a compelling story, engaging action sequences, evocative music, and elements of supernatural and fantasy contributed to the recipe for a successful anime movie.

3. Delving into specific success stories, 'Demon Slayer: Mugen Train' stood out. The film's success can be attributed to its prequel's strong reputation, deliberate choices in animation, action-packed sequences, emotionally resonant storytelling, and character development. Character attachment and writing emerged as pivotal factors, highlighting the audience's investment in fictional characters due to their imaginative appeal.

The reviews' word cloud analysis underscored the significance of a production studio's reputation in influencing an anime's success. Action sequences outweighed drama or storyline preferences, indicating a trend among anime audiences. 'Mugen Train,' a sequel to the popular 'Demon Slayer' series, known for its animation quality and action, also benefitted from being produced by Ufotable, compelling viewers to experience it in theaters.

This analysis not only unveils specific insights but also showcases the underlying patterns guiding audience preferences and the intricate elements contributing to an anime's success. Understanding these trends provides valuable guidance for creators and studios in crafting compelling anime experiences that resonate deeply with their audiences."

This refined conclusion aims to present your key findings in a more structured and cohesive manner, emphasizing the significance of the insights drawn from the analysis of anime reviews.

Limitations:

However, it's important to note that our analysis has limitations. The reviews we used might not represent all anime viewers, and there could be biases in the data from specific sources.

Understanding what anime audiences prefer and what makes a movie successful can guide creators. They can focus on great animation, engaging stories, and memorable characters to make anime that people will love. For marketing, highlighting top-notch animation and action sequences, along with well-developed characters, can attract more viewers.

In conclusion, while considering these limitations, our findings offer practical advice for creators and marketers. This insight can help make anime that truly connects with audiences and succeeds in the industry.

REFERENCE:

- *https://www.youtube.com/watch?v=szczpgOEdXs*
- *https://www.nltk.org/_modules/nltk/sentiment/vader.html*
- *https://www.datacamp.com/tutorial/text-analytics-beginners-nltk*
- *https://towardsdatascience.com/end-to-end-topic-modeling-in-python-latent-dirichlet-allocation-lda-35ce4ed6b3e0*