CHRIST
(DEEMED TO BE UNIVERSITY)
BANGALORE · INDIA

Uncovering the Game Chat: A Topic Modeling Analysis of Apex Legends Discord Chats using Latent Dirchlet Allocation (LDA) and Naive Bayes Model

by
Satyam Gupta-2248015
Prathana.M -2248063
Harsha KG -2248035

Under the guidance of
Mrs. Umme Salma

Acknowledgments

I would like to express our heartfelt gratitude to the Almighty for providing us with the strength, guidance, and wisdom to successfully complete the machine learning full-fledged project on "**Uncovering the Game Chat: A Topic Modeling Analysis of Apex Legends Discord Chats**".

We would also like to extend our sincere appreciation to **Christ University, Bangalore**, for providing us with the opportunity to pursue this project and for the resources and support provided throughout the project's duration.

We would like to acknowledge and thank our class teacher, **Mrs. Umme Salma**, for her constant encouragement, guidance, and invaluable support throughout the project. Her expertise and mentorship were instrumental in helping us navigate the complexities of the project and achieve its objectives.

Once again, we extend our gratitude to all those who have contributed to the successful completion of this project.

Table of contents

Introduction

Discord is a popular communication platform that is widely used by gamers around the world. In online gaming, players interact with each other through various modes of communication, such as in-game voice and text chats, forums, and social media. The data generated by these communication channels can be used to gain valuable insights into the behavior and social interactions of the players. The study of such data is known as chat data analysis and has gained significant attention in recent years **[1]**.

Latent Dirichlet Allocation (LDA) is a popular topic modeling technique that has been widely used for chat data analysis **[3, 6, 7]**. LDA helps in discovering latent topics from a large corpus of text data. It has been used to identify the major themes and topics of conversation in online gaming chat data **[9]**. Sentiment analysis is another popular technique that has been used to analyze the emotional tone of the text data. It helps in identifying the polarity of the text, i.e., whether it is positive, negative, or neutral. TextBlob, a popular Python library, has been used for sentiment analysis in this study **[8]**. To check the accuracy of TextBlob, a Naive Bayes classifier has been used **[8]**.

The aim of this study is to perform chat data analysis on the Discord server of Apex Legends, a popular battle royale game. Apex Legends has gained immense popularity since its release in 2019 and has a large player base. The chat data generated by the players can provide valuable insights into their behavior and social interactions. In this study, we have used LDA for topic modeling and TextBlob for sentiment analysis. We have also used a Naive Bayes classifier to check the accuracy of TextBlob. The study provides insights into the major topics of conversation and the emotional tone of the chat data generated by the players. Chat data analysis has gained significant attention in the gaming industry in recent years. It helps in understanding the behavior and social interactions of the players, which can be used to improve the game design and player engagement **[5]**. Social network analysis is another technique that has been used for chat data analysis **[9, 10]**. It helps in identifying the key players and communities within the game. The study of chat data is a rapidly evolving field and holds great potential for future research.

Problem statement

The popularity of online games has grown substantially in recent years, with millions of players engaging in multiplayer games such as Apex Legends. These games provide a social platform for players to interact and communicate with each other via chat. The chat data generated during gameplay can provide insights into player behavior, attitudes, and emotions, which can be used to improve the overall gaming experience. However, the analysis of chat data in online games presents several challenges, such as the large volume of data, the complexity of the text, and the need for real-time analysis.

In this project, we aim to analyze the chat data generated in the Apex Legends game server to gain insights into player behavior, attitudes, and emotions. We have used Latent Dirichlet Allocation (LDA) to identify the topics of discussion in the chat data, and textblob for sentiment analysis to identify the emotional tone of the chat messages. To check the accuracy of the sentiment analysis, we have used the Naive Bayes classifier to classify the unknown chats.

Previous studies have investigated the analysis of chat data in online games **[1, 2, 3, 4, 5, 6, 7, 9, 10]**, but there is still a lack of research on analyzing chat data specifically from Apex Legends. This study fills this gap by examining the chat data from the Apex Legends game server. The results of this study will help game developers to gain insights into player behavior, attitudes, and emotions, which can be used to improve the overall gaming experience.

One of the challenges in analyzing chat data is the large volume of data generated during gameplay. In online games, players can generate a massive amount of chat data within a short period. Therefore, we need efficient methods to analyze this data. LDA is a popular technique for topic modeling that can identify the topics of discussion in a large corpus of text data. Using LDA, we can identify the main topics of discussion in the chat data, which can provide insights into the interests and concerns of the players.

Another challenge in analyzing chat data is the complexity of the text. Chat messages in online games can contain a wide range of expressions, including slang, emoticons, and acronyms.

Textblob is a natural language processing (NLP) tool that can handle such complex text and provide sentiment analysis of the chat messages. Sentiment analysis can identify the emotional tone of the chat messages, which can provide insights into the players' emotions and attitudes towards the game.

To check the accuracy of the sentiment analysis, we have used the Naive Bayes classifier. The Naive Bayes classifier is a popular machine learning algorithm for classification tasks and has been used in previous studies for sentiment analysis **[8]**. Using the Naive Bayes classifier, we can evaluate the accuracy of the sentiment analysis and improve the overall quality of the results.

Aim and Objectives

The aim of this study is to analyze the Discord chat from the Apex Legends game server using Latent Dirichlet Allocation (LDA) and sentiment analysis with textblob, and to evaluate the accuracy of textblob's sentiment analysis through Naive Bayes classifier.

The objectives of the study are:

- To identify the topics discussed in the Discord chat from the Apex Legends game server using LDA.
- To analyze the sentiment of the chat messages using textblob and evaluate its accuracy through Naive Bayes classifier.
- To examine the relationship between the identified topics and sentiment in the chat messages.
- To provide insights into the communication patterns and behaviors of players in the Apex Legends game server.
- To compare the findings of this study with previous research on chat data analysis in online games **[1, 2]**, player chat message analysis in online games using clustering **[3]**, sentiment analysis of online game reviews using deep learning **[4]**, and social network analysis of massively multiplayer online game player chat data [**9**].
- To contribute to the existing body of research on text analysis of online communication in collaborative learning environments **[6, 7]** and intentional social action in online games **[5]**.
- To suggest potential implications for game developers and community managers to improve player experience and engagement in the Apex Legends game server.
- To explore the possibility of using hierarchical graph convolutional networks for analyzing social behavior in massively multiplayer online games **[10]**.

By achieving these objectives, this study aims to provide a comprehensive analysis of the Discord chat from the Apex Legends game server, shedding light on the communication patterns and behaviors of players and suggesting potential implications for game developers and community managers.

Literature Review

Analyzing player chat data can provide insights into a wide range of topics, such as player motivations, social interactions, and gameplay strategies. However, analyzing chat data can be challenging due to the large amount of data that is generated and the need to extract meaningful information from unstructured text. In recent years, researchers have developed a variety of methods for analyzing chat data, including natural language processing techniques, sentiment analysis, social network analysis, and clustering. This literature review provides an overview of some of the most important and recent papers that have explored these methods.

Liao et al. (2019)**[1]** provided a comprehensive review of the state-of-the-art methods for analyzing player chat data in online games. They highlighted the importance of chat data analysis in understanding player behaviors and preferences and provided an overview of the various techniques used for analyzing chat data, including natural language processing, sentiment analysis, and social network analysis. They also discussed the challenges and limitations of chat data analysis, such as data privacy concerns and the need for large-scale data collection. Overall, the paper provides a useful resource for researchers and game developers interested in analyzing player chat data.

Wang et al. (2018)**[2]** conducted a survey of natural language processing techniques used for analyzing online social media data. While not specifically focused on online games, the paper provides insights into the challenges and opportunities of working with social media data, which shares many similarities with player chat data. The authors discussed various methods for text preprocessing, such as tokenization, part-of-speech tagging, and named entity recognition, and highlighted the challenges of working with non-standard language and the need for domain-specific lexicons. The paper also provided an overview of the various applications of natural language processing in social media analysis, including sentiment analysis, topic modeling, and opinion mining.

Ismailov and Vasileva (2018)**[3]** proposed a clustering-based approach for analyzing player chat messages in online games. They applied k-means clustering to group similar chat messages together and identify common themes and topics of conversation. They also evaluated the effectiveness of the clustering approach by comparing it to a traditional topic modeling approach. The results showed that clustering can provide a useful alternative to topic modeling for analyzing chat data, particularly in cases where the data is noisy or contains a large number of outliers.

Liu et al. (2018)**[4]** proposed a deep learning-based approach for sentiment analysis of online game reviews. The authors applied convolutional neural networks and long short-term memory networks to analyze the text data and classify reviews as positive, negative, or neutral. They also evaluated the performance of the deep learning models and compared them to traditional machine learning models. The results showed that deep learning models can outperform traditional models in terms of accuracy and efficiency.

Cheung and Lee (2010)**[5]** proposed a theoretical model of intentional social action in online games. The authors argued that players engage in intentional social action, such as forming alliances and collaborating on quests, to achieve social and gameplay-related goals. They also identified various factors that influence players' decisions to engage in intentional social action, including social identity, social capital, and social norms. The paper provides a useful framework for understanding player behaviors in online games and designing game features that encourage social interaction.

Li et al. (2019)**[5]** presented an application of topic modeling for exploratory text analysis in the context of student online discussion posts. The authors used Latent Dirichlet Allocation (LDA) to identify and analyze topics in the discussion posts, as well as to understand the sentiment of the posts. The study demonstrates the usefulness of topic modeling for analyzing large amounts of unstructured text data in the context of educational research.

Another study that used topic modeling is the work by Cai et al. (2018)**[7]**, where the authors propose a method for analyzing students' learning processes in a collaborative learning environment using topic models. They used Latent Dirichlet Allocation (LDA) to identify the topics discussed by students and their relationships with learning outcomes. The study shows that topic models can provide insights into the learning process and help instructors to better understand and support students in collaborative learning environments.

In addition to topic modeling, another popular technique in natural language processing is sentiment analysis. Liu et al. (2018)**[4]** proposed a deep learning-based approach for sentiment analysis of online game reviews. The authors applied convolutional neural networks and long short-term memory networks to analyze the text data and classify reviews as positive, negative, or neutral. They also evaluated the performance of the deep learning models and compared them to traditional machine learning models. The results show that deep learning models can outperform traditional models in terms of accuracy and efficiency.

Furthermore, textblob, a Python library, has emerged as a powerful tool for natural language processing tasks. It provides various functionalities such as sentiment analysis, part-of-speech tagging, noun phrase extraction, and more. Recently, researchers have started to use textblob for analyzing player chat data in online games. One study that used textblob is the work by Ismailov and Vasileva (2018) **[3]**, where the authors applied clustering to analyze player chat messages in online games. They used the polarity scores provided by textblob to group similar chat messages together and identify common themes and topics of conversation. The study shows that textblob can provide a useful tool for analyzing chat data in online games.

To evaluate the accuracy of textblob for sentiment analysis, researchers have started to use machine learning techniques such as Naive Bayes classifier. Naive Bayes classifier is a simple yet effective probabilistic algorithm that can be used for binary and multi-class classification problems. In the context of sentiment analysis, Naive Bayes classifier can be used to classify text as positive, negative, or neutral based on the polarity scores provided by textblob. A study that used Naive Bayes classifier for evaluating the accuracy of textblob is the work by Patel and Patel (2019) **[8]**, where the authors applied Naive Bayes classifier to classify movie reviews as positive or negative based on the polarity scores provided by textblob. The results show that textblob combined with Naive Bayes classifier can provide accurate sentiment analysis results.

Kasurinen et al. (2020)**[9]** paper analyzed player chat data in a massively multiplayer online game to study the social behavior of players. They used social network analysis to identify different types of social relationships between players and found that players with stronger social connections were more likely to engage in cooperative gameplay. The study shows the potential for social network analysis to provide insights into the social dynamics of online games. This is relevant to our goal as we may want to analyze the social behavior of players in the Apex Legends server using the gaming discussion channel.

Xiao et al. (2019)**[10]** paper proposed a framework for analyzing player chat data based on deep learning and network embedding techniques. They used a graph-based approach to represent the social network of players and trained a deep learning model to predict the topics of conversation in chat messages. The study shows the potential of combining deep learning and social network analysis for analyzing player chat data.

In conclusion, natural language processing techniques have become an essential tool for analyzing player chat data in online games. Researchers have used various techniques such as topic modeling, sentiment analysis, and clustering to analyze chat data and gain insights into player behaviors and preferences. Recent studies have also highlighted the usefulness of textblob for analyzing chat data and evaluating its accuracy using machine learning techniques such as Naive Bayes classifier. With the increasing popularity of online games and the growing importance of chat data analysis, natural language processing techniques are expected to play a crucial role in understanding player behaviors and designing better game features."

Preliminaries

About Discord and the Apex legends server:

Discord is a communication platform used by gamers and non-gamers to create and join virtual spaces where they can interact with others via text, voice, and video, share video, and collaborate on projects. Discord servers are organized into text and voice channels, which are usually dedicated to specific topics and can have different rules. Users can be assigned different roles within a server, giving them varying levels of permissions and access. Discord servers are an essential tool for gaming communities, providing a reliable and versatile platform for communication, organization, and collaboration.

Apex Legends community on Discord is one of the most active and vibrant gaming communities on the platform, offering channels for players to connect, share tips and tricks, discuss strategy and gameplay mechanics, and host events and tournaments. The gaming discussion channel of an Apex Legends server can be analyzed to gain insights into the behavior of human players.

Steps to extract data from discord channels:

- Join the discord server
- Collect your user token
- Insert the token in the discord extracter (https://github.com/Tyrrrz/DiscordChatExporter)
- Select the channel we wish to extract the data from. In this case, we have selected the gaming discxussion channel from the Apex legends server.
- Specify the type of file we want to extract the data into. Csv in this case
- After a few minutes, the csv file is ready.

Dataset Description

- **Author ID:** This field is a unique identifier for each message sent in the channel. It is important for identifying and tracking specific messages in the channel, and can be used to group messages by topic or conversation. Message IDs can also be used to analyze message length, frequency, and patterns over time.

- **Author:** This field represents the username or display name of the user who sent the message. It is important for identifying individual users and their messaging patterns within the channel, and can be used to analyze the engagement level of specific users, their posting frequency, and the sentiment of their messages.

- **Content:** This field includes the text content of the message, including any links, emojis, or other media. It is important for understanding the topics and themes of the channel, and can be used to identify common keywords or topics, as well as the sentiment of the conversation. Analyzing the length, tone, and content of messages can provide insights into user behavior and engagement.

- **Timestamp:** This field represents the date and time when the message was sent. It is important for analyzing message frequency, patterns, and trends over time. By grouping messages by time intervals (e.g. daily, weekly, monthly), analysts can identify peak activity times and patterns in the conversation, as well as track the evolution of topics and themes over time.

NLP

Natural Language Processing (NLP) is a subfield of artificial intelligence that focuses on the interaction between computers and humans in natural language. NLP has become increasingly important in handling text data because it allows machines to understand, interpret and generate human language. The importance of NLP in text data lies in its ability to perform various tasks, including sentiment analysis, language translation, named entity recognition, summarization, and speech recognition. With NLP, machines can analyze and derive insights from vast amounts of text data, which can help organizations make better decisions, improve customer experience, and enhance overall operational efficiency.

- **Tokenization**

Tokenization is the process of breaking text into individual words or sentences. This step is crucial in natural language processing (NLP) because it helps machines learn the context and meaning of the text. Once tokenization is performed on a corpus, the resulting tokens can be used to prepare a vocabulary that is used in further steps to train the NLP model.

There are several methods and libraries available to convert text into tokens. The simplest technique is word (white space) tokenization, which splits the corpus based on white space or a certain delimiter. Another technique is character tokenization, which splits words at the character level. This technique helps to overcome issues faced by word tokenization and can handle out-of-vocabulary (OOV) cases, as it deals with the characters of the corpus. However, it also limits the size of the vocabulary, as it only deals with the 26 alphabets and special characters.

- **Stop Words**

Stopwords are common words in any language that do not add much meaning to a sentence and can be safely ignored without sacrificing the sentence's meaning. Examples of stopwords include "the," "is," "at," "which," and "on." In some search engines, these common, short function words can cause problems when searching for phrases that include them, particularly in names such as "The Who" or "Take That."

- **POS Tagging**

The part of speech (POS) explains how a word is used in a sentence. There are eight main parts of speech: nouns, pronouns, adjectives, verbs, adverbs, prepositions, conjunctions, and interjections. Most POS are divided into sub-classes. POS tagging is the process of labeling words with their appropriate part-of-speech, which is a supervised learning solution that uses features like the previous word, next word, is first letter capitalized, etc. The Natural Language Toolkit (NLTK) has a function to get POS tags, and it works after the tokenization process.

- **Lemmatization and Stemming**

Lemmatization and stemming are text pre-processing techniques used in NLP models to break a word down to its root meaning to identify similarities. Lemmatization reduces the word to its root form, such as reducing the word "better" to its root word "good." Stemming, on the other hand, refers to the process of removing suffixes from words to represent or cluster multiple forms of words in a single base form. This is a common vocabulary normalization technique used to eliminate the small meaning difference of pluralization, possessive endings, or even various verb forms.

- **Importance of extracting urls, mentions and hashtags:**

There are several reasons why extracting URLs, hashtags, and mentions from chat in gaming Discord servers using NLP techniques can be important:

- Content moderation: NLP techniques can extract URLs, hashtags, and mentions from chat in gaming Discord servers to help server moderators understand what users are discussing and identify problematic content that needs to be removed.
- User engagement: Analyzing extracted URLs, hashtags, and mentions can provide valuable information about user interests and topics of discussion, which can help server moderators create more engaging conversations and events.
- Advertising and marketing: Extracted URLs and hashtags can be used to promote gaming-related products or services, and server moderators can identify potential advertising opportunities or partnerships based on this information.
- Data analysis: Extracting URLs, hashtags, and mentions can provide insights into user behavior and interests, which can be used to improve the user experience and develop new features and services that meet user needs.

Overall, using NLP techniques to extract URLs, hashtags, and mentions from chat in gaming Discord servers can provide valuable insights that can be used to improve content moderation, user engagement, and advertising/marketing efforts.

- **Handling Emoticon ‘ :-) ’ and emoji in gaming chat:**

An emoticon is a short form of “Emotion & Icon”.It is a representation of a facial expression such as a smile or frown, formed by various combinations of keyboard characters and used to convey the writer’s feelings or intended tone. Emoji is a small digital image or icon used to express an idea or emotion. These are small enough to insert into the text. In Japanese “e” means picture and “moji” means character. Emoji’s can be handled by removing it from the text or replacing it with the meaningful word relating to the emoji.

- **Latent Dirichlet Allocation:**

Latent Dirichlet Allocation (LDA) is a popular algorithm used for topic modeling in natural language processing (NLP). It is an unsupervised machine learning technique that can identify latent topics in a large corpus of text data. In the context of gaming chat, LDA can be used to identify the most frequently discussed topics among players.

The algorithm works by assuming that each document in the corpus is a mixture of a small number of topics, and that each word in the document is generated from one of those topics. The goal is to discover the underlying topics and their associated word distributions that best explain the observed data.LDA requires specifying the number of topics beforehand and iteratively learns the word distributions for each topic and the document-topic distributions until convergence. Once trained, LDA can be used to infer the topic distribution for new documents, which can be used for further analysis or visualization.

- **TextBlob library**

TextBlob is a Python library that provides simple API for common natural language processing (NLP) tasks such as sentiment analysis, part-of-speech tagging, and noun phrase extraction. TextBlob's sentiment analysis uses a pattern analyzer to classify text into positive, negative, or neutral categories based on the presence of positive and negative words. The library also provides a trained Naive Bayes classifier that can be used to classify text into custom categories.

- **Naive Bayes model**

Naive Bayes is a classification algorithm based on Bayes' theorem, which calculates the probability of a hypothesis given the evidence. In the context of sentiment analysis, the evidence is the presence of words in the text, and the hypothesis is the sentiment of the text (positive, negative, or neutral). The Naive Bayes classifier uses a multi-nomial distribution to estimate the probability of each class given the presence of words in the text.

Proposed Work

The process of extracting valuable insights from raw data is nothing short of a creative endeavor. It's a journey that requires a curious mind, a meticulous attention to detail, and an unwavering commitment to the scientific method.

The data science process begins with a question, a hypothesis, or a problem to be solved. It's the starting point for the journey, and it's the spark that ignites the creative process. From there, the data scientist must collect and prepare the data, exploring its nuances and uncovering hidden patterns and trends.

Next comes the analysis phase, where the data is explored, visualized, and modeled. It's a process that requires a blend of technical skill and intuition, as the data scientist seeks to understand the underlying structures and relationships in the data.

As the analysis unfolds, the data scientist may encounter unexpected results or anomalies, prompting a return to the earlier stages of the process. It's a fluid and iterative journey, where each step informs the next and the data scientist must remain open-minded and flexible.

Finally, the data science process culminates in the presentation of findings. It's the moment when the creative process meets the practical, as the data scientist communicates insights to stakeholders and decision-makers.

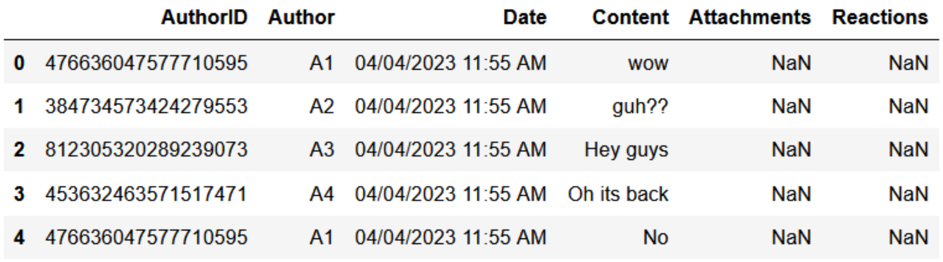
The steps involved in a datascience project are

1. **Data Preprocessing:** This step involves getting the data in a format that is suitable for analysis.The data obtained from the Discord server contain many irrelevant or sensitive information.

1.1. Anonymizing the data

It's not ethical (and probably not even legal) to make the account names of the people in a dataset public without their permission, so we'll anonimize them first.

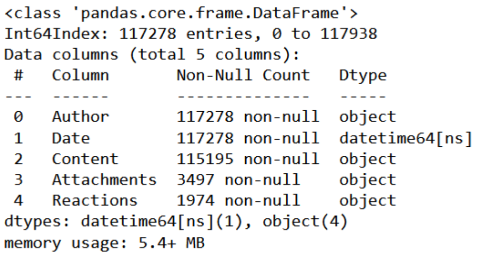
Takes a pandas Series object of 'names', creates a dictionary where each unique name is replaced with an anonymized label 'A' followed by a unique integer index, and returns the resulting anonymized dictionary and passed the values to a column name Author.



	AuthorID	Author	Date	Content	Attachments	Reactions
0	476636047577710595	A1	04/04/2023 11:55 AM	wow	NaN	NaN
1	384734573424279553	A2	04/04/2023 11:55 AM	guh??	NaN	NaN
2	812305320289239073	A3	04/04/2023 11:55 AM	Hey guys	NaN	NaN
3	453632463571517471	A4	04/04/2023 11:55 AM	Oh its back	NaN	NaN
4	476636047577710595	A1	04/04/2023 11:55 AM	No	NaN	NaN

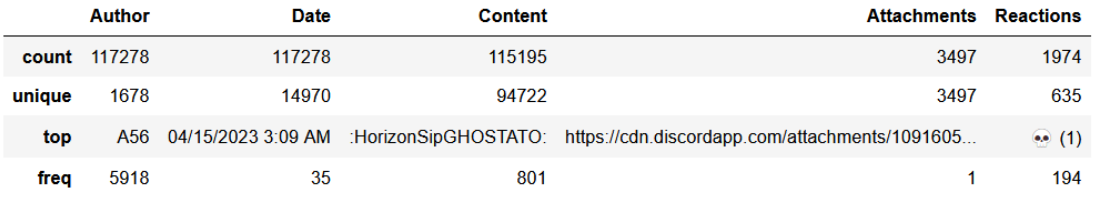
Fig: Anonymizing the data

Next we did the basic techniques like getting the statistical inference of each column, removing duplicates checking null values, replacing null values with specific values.



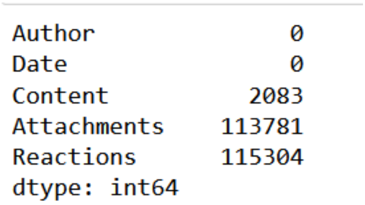
```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 117278 entries, 0 to 117938
Data columns (total 5 columns):
 #   Column       Non-Null Count   Dtype
---  ------       --------------   -----
 0   Author       117278 non-null  object
 1   Date         117278 non-null  datetime64[ns]
 2   Content      115195 non-null  object
 3   Attachments  3497 non-null    object
 4   Reactions    1974 non-null    object
dtypes: datetime64[ns](1), object(4)
memory usage: 5.4+ MB
```

Fig: Summary of Dataset



	Author	Date	Content	Attachments	Reactions
count	117278	117278	115195	3497	1974
unique	1678	14970	94722	3497	635
top	A56	04/15/2023 3:09 AM	:HorizonSipGHOSTATO:	https://cdn.discordapp.com/attachments/1091605...	💀 (1)
freq	5918	35	801	1	194

Fig: Statistical Summary of Each Columns



```
Author              0
Date                0
Content          2083
Attachments    113781
Reactions      115304
dtype: int64
```

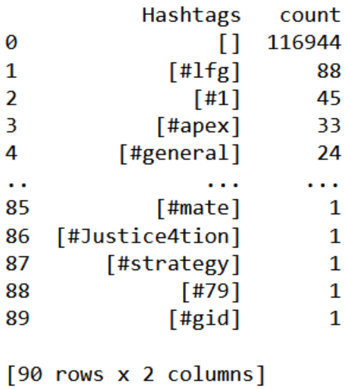
Fig: Sum of Null values Each Columns

- Replaced the null values of Content with “Not Captured” and remove the columns Attachments and Reactions as both the columns were having more than 80% of null values.

- Next, the timestamp column stored as a datetime64 so it is important to convert this to a date and time.. This will allow us to easily sort the data by date and time.

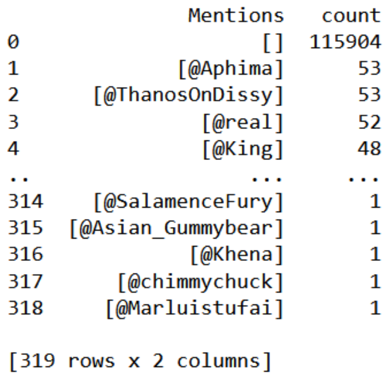
1.3. Hashtag, Mentions and URL extraction

Hashtags are a common feature in Discord chat, and it is important to extract them for analysis. The first step in this process is to extract all the hashtags present in the 'content' column. This can be done using regular expressions. Once all the hashtags have been extracted, we count the frequency of each hashtag, and create a column for each unique hashtag in the dataset. Same procedure we follow for Hashtag, Mentions and URL extraction.



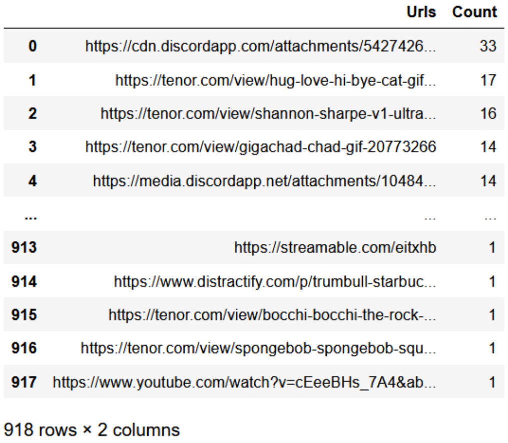
```
              Hashtags   count
0                   []  116944
1               [#lfg]      88
2                [#1]       45
3              [#apex]      33
4           [#general]      24
..                 ...     ...
85             [#mate]       1
86    [#Justice4tion]        1
87         [#strategy]       1
88              [#79]        1
89             [#gid]        1

[90 rows x 2 columns]
```

Fig: Count of Hashtags



```
                  Mentions   count
0                       []  115904
1                [@Aphima]      53
2         [@ThanosOnDissy]      53
3                  [@real]      52
4                  [@King]      48
..                     ...     ...
314      [@SalamenceFury]       1
315    [@Asian_Gummybear]       1
316                [@Khena]       1
317         [@chimmychuck]       1
318        [@Marluistufai]       1

[319 rows x 2 columns]
```

Fig: Count of Mentions



	Urls	Count
0	https://cdn.discordapp.com/attachments/5427426...	33
1	https://tenor.com/view/hug-love-hi-bye-cat-gif...	17
2	https://tenor.com/view/shannon-sharpe-v1-ultra...	16
3	https://tenor.com/view/gigachad-chad-gif-20773266	14
4	https://media.discordapp.net/attachments/10484...	14
...	...	...
913	https://streamable.com/eitxhb	1
914	https://www.distractify.com/p/trumbull-starbuc...	1
915	https://tenor.com/view/bocchi-bocchi-the-rock-...	1
916	https://tenor.com/view/spongebob-spongebob-squ...	1
917	https://www.youtube.com/watch?v=cEeeBHs_7A4&ab...	1

918 rows × 2 columns

Fig: Count of URLs

1.4. Text Cleaning

- **Remove url's , puntuations , mentions, numbers, emojis etc and also convert all letters in lower case**

  Defines a function called "clean" that takes a string as input and applies several regex-based cleaning operations to it, including removing mentions, URLs, hashtags, numbers, punctuation, and extra whitespace, emojis.

- **Lemmatization, POS tagging and Tokenization**

  Lemmatization, POS tagging, and tokenization are all important techniques used in natural language processing (NLP) to analyze text data, including game chat.

Tokenization is the process of breaking down a text into individual words or tokens, which can then be analyzed further.

POS tagging is the process of labeling each token with its corresponding part of speech, such as noun, verb, or adjective.

Lemmatization is the process of reducing each word to its base or dictionary form, known as a lemma, which can help to identify the core meaning of the word and improve analysis accuracy.

Together, these techniques can help to extract valuable insights from game chat data, such as identifying frequently used words or topics of conversation.

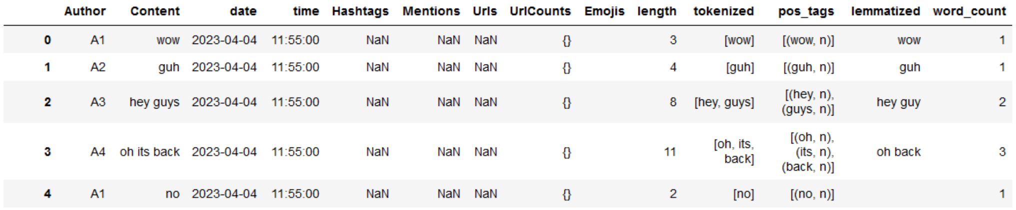
- **Remove Stop Words**

  In game chat, stop words are common words such as "the", "and", "is", etc. that do not carry much meaning and can be removed to focus on the important words.

  To remove stop words in game chat, we can use NLP libraries like NLTK or spaCy. These libraries provide pre-built stop word lists that can be used to filter out stop words from game chat.

- **Adding a Word Count Column**

  We can see the average length of a user's message by adding a column that keeps track of their word count per message.



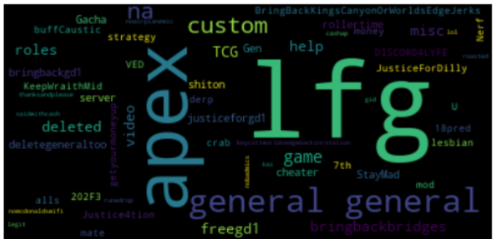
	Author	Content	date	time	Hashtags	Mentions	Urls	UrlCounts	Emojis	length	tokenized	pos_tags	lemmatized	word_count
0	A1	wow	2023-04-04	11:55:00	NaN	NaN	NaN	{}		3	[wow]	[(wow, n)]	wow	1
1	A2	guh	2023-04-04	11:55:00	NaN	NaN	NaN	{}		4	[guh]	[(guh, n)]	guh	1
2	A3	hey guys	2023-04-04	11:55:00	NaN	NaN	NaN	{}		8	[hey, guys]	[(hey, n), (guys, n)]	hey guy	2
3	A4	oh its back	2023-04-04	11:55:00	NaN	NaN	NaN	{}		11	[oh, its, back]	[(oh, n), (its, n), (back, n)]	oh back	3
4	A1	no	2023-04-04	11:55:00	NaN	NaN	NaN	{}		2	[no]	[(no, n)]		1

Fig: After Lemmatization, POS tagging and Tokenization, removing stop words

2. Exploratory Data Analysis

Once the data has been cleaned and prepared, we can use various visualization techniques to gain insights from the data

- **WordCloud Representation to visualize the maximum times each features(Hashtags, Mentions)**



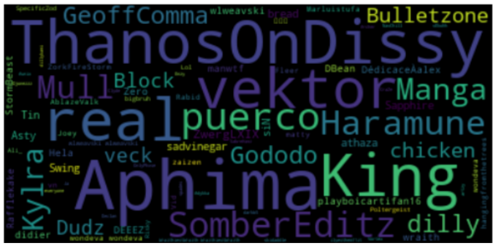
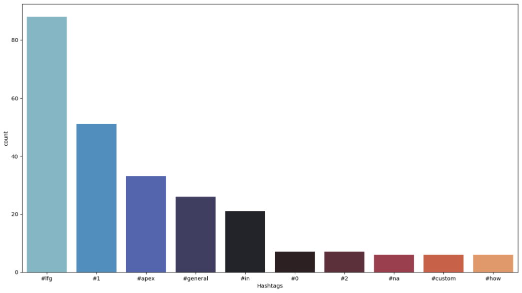


Fig: HashTags **Fig: Mentions**

- **Top 10 Hashtags and Mentions**



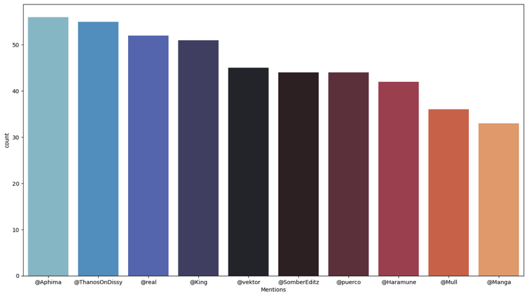


Fig: HashTags **Fig: Mentions**

- **Histogram to visualize the length of the text**

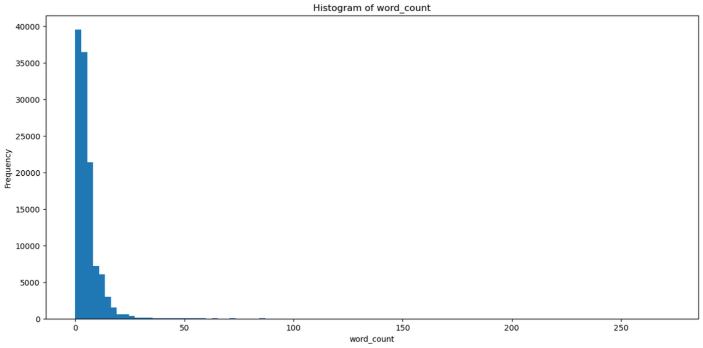


Fig:Histogram of word count

- **User Activity Over Time**

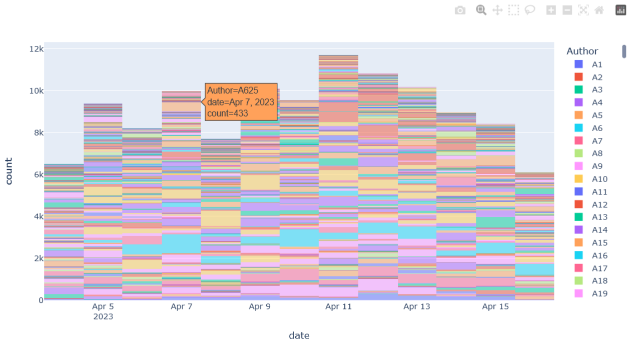


Fig: User Activity Over Time

- **Message Sent Per Author**

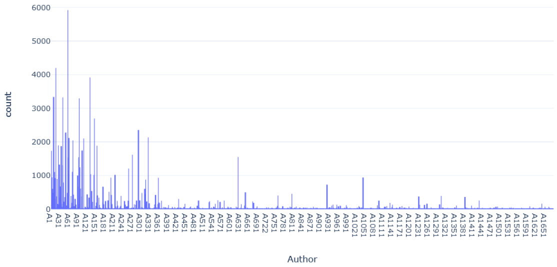


Fig: Message Sent Per Author

Dropping all authors who have sent less messages than a minimum amount, we're filtering out those who don't significantly alter our data.

- **Top 10 Authors by Message Count**

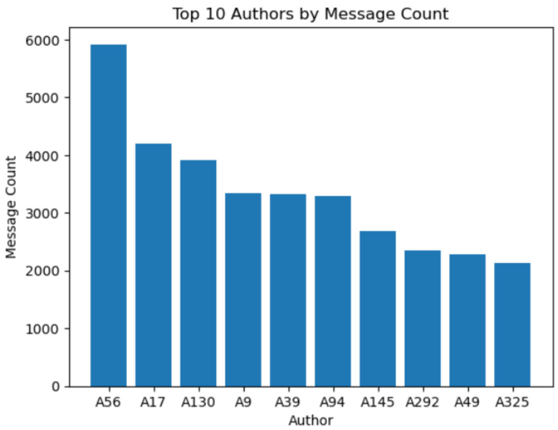
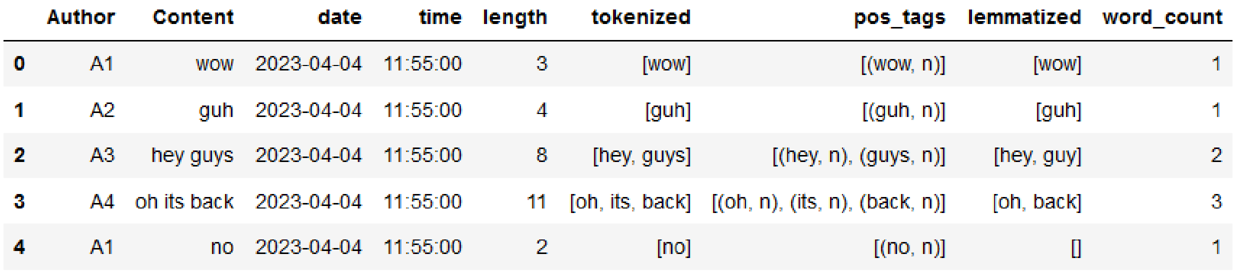


Fig: Top Ten Authors by Message Count

- **Now we create a separate dataframe with columns that we required for model building**



	Author	Content	date	time	length	tokenized	pos_tags	lemmatized	word_count
0	A1	wow	2023-04-04	11:55:00	3	[wow]	[(wow, n)]	[wow]	1
1	A2	guh	2023-04-04	11:55:00	4	[guh]	[(guh, n)]	[guh]	1
2	A3	hey guys	2023-04-04	11:55:00	8	[hey, guys]	[(hey, n), (guys, n)]	[hey, guy]	2
3	A4	oh its back	2023-04-04	11:55:00	11	[oh, its, back]	[(oh, n), (its, n), (back, n)]	[oh, back]	3
4	A1	no	2023-04-04	11:55:00	2	[no]	[(no, n)]	[]	1

Fig: Dataframe for Model Building

3. Model Building

Based on the problem statement we have chose Latent Dirichlet Allocation (LDA) a probabilistic model for topic modelling and then we perform sentiment analysis, which involves classifying each row of text as positive, negative, or neutral. Then train a machine learning model to classify this text data based on its sentiment.

3.1. Latent Dirchlet Allocation(LDA) for Topic Modelling

Latent Dirichlet Allocation (LDA) is a statistical model used for topic modeling, which can be applied to game chat data to identify recurring topics or themes within the conversations. LDA assumes that each document in the corpus (e.g. game chat messages) is a mixture of topics, and each topic is a distribution of words. By analyzing the frequency of words and their co-occurrence patterns in the messages, LDA can uncover the underlying topics and their relative importance in the corpus.

- **Create a document-term matrix using CountVectorizer**

  The purpose of creating a document-term matrix using CountVectorizer is to represent the text data in a numerical format that can be used for machine learning and natural language processing tasks such as topic modeling.

  CountVectorizer is a technique used to convert a collection of text documents to a matrix of token counts. It essentially counts the frequency of each word (or n-gram) in each document, and creates a sparse matrix representation where each row represents a document and each column represents a unique word in the corpus.

- **Fit an LDA model with 3 topics**
- **Extract the topic-word matrix and print the top 10 words for each topic**

3.2. Perform Sentiment Analysis using Textblob Library

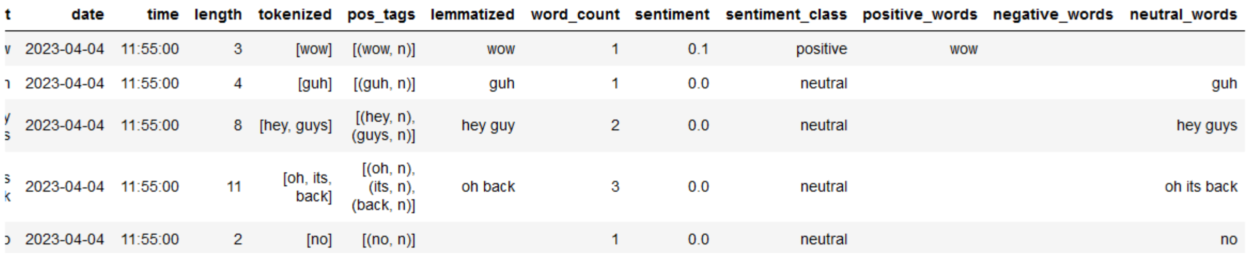
Sentiment analysis using TextBlob is a natural language processing technique that involves analyzing a piece of text to determine whether the overall sentiment expressed in it is positive, negative, or neutral.

The TextBlob library provides a simple interface for performing sentiment analysis by assigning polarity scores to words based on their semantic orientation (positive or negative) and using these scores to determine the overall sentiment of the text.

This technique can be used for classifying game chat messages into positive, negative, and neutral categories, which can be useful for analyzing player feedback, identifying potential issues with the game, and improving the player experience.

Performs sentiment analysis using TextBlob library by creating a 'sentiment' column based on lemmatized sentences. It then classifies sentiments as positive, negative, or neutral using Textblob library. The sentiment polarity ranges from -1 to 1, where a negative value indicates negative sentiment, a positive value indicates positive sentiment, and a value of 0 indicates neutral sentiment.. Finally, it extracts positive, negative, and neutral words from the Content' column and saves them in separate columns.

Output



t	date	time	length	tokenized	pos_tags	lemmatized	word_count	sentiment	sentiment_class	positive_words	negative_words	neutral_words
v	2023-04-04	11:55:00	3	[wow]	[(wow, n)]	wow	1	0.1	positive	wow		
ı	2023-04-04	11:55:00	4	[guh]	[(guh, n)]	guh	1	0.0	neutral			guh
y s	2023-04-04	11:55:00	8	[hey, guys]	[(hey, n), (guys, n)]	hey guy	2	0.0	neutral			hey guys
s k	2023-04-04	11:55:00	11	[oh, its, back]	[(oh, n), (its, n), (back, n)]	oh back	3	0.0	neutral			oh its back
ɔ	2023-04-04	11:55:00	2	[no]	[(no, n)]		1	0.0	neutral			no

Fig: Dataframe with Sentiment_class and word colum for positive, negative and neutral

3.3. Implementation of a simple text classification model using the Mutinomial Naive Bayes algorithm

The input data consists of two columns: data['lemmatized'](Independent Variable) which contains the preprocessed text data(X), and data['sentiment_class'](Target Variable) which contains the corresponding sentiment class label for each text sample.

The Naive Bayes classifier is trained on this data to learn the patterns in the text that are associated with each sentiment class. ***The Multinomial Naive Bayes algorithm is commonly used for text classification tasks because it is effective in modeling the frequency distributions of words in text data.***

3.4. Model Validation

- **Confusion Matrix and Classification report**

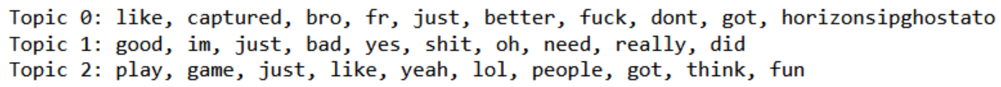
3.5. Validating the model on a set of new Random chats to identiy the nature of clusters

# Define the new chat messages

new_messages = ["hi!morning", "fuck!bad game", "play yaar", "what man", "lost-game", "good shot", "played well"]

Result and Discussion

- **Latent Dirchlet Allocation(LDA) for Topic Modelling**



```
Topic 0: like, captured, bro, fr, just, better, fuck, dont, got, horizonsipghostato
Topic 1: good, im, just, bad, yes, shit, oh, need, really, did
Topic 2: play, game, just, like, yeah, lol, people, got, think, fun
```

Fig: Three Topics obtained after Latent Dirchlet Allocation(LDA) Model

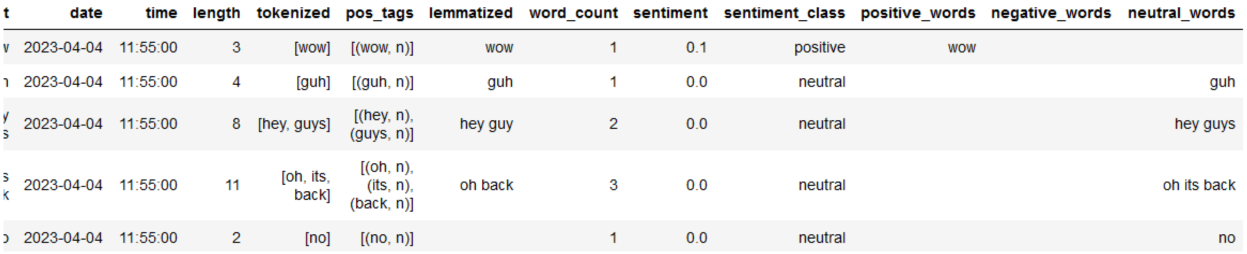
Inference:

Topic 0 appears to be focused on casual conversation, using slang and profanity. It's likely a discussion between friends, with topics ranging from personal experiences to opinions and reactions. ***The words "like," "bro," "fr" (short for "for real"), "fuck," and "got" are frequently used, indicating an informal tone.***

Topic 1 seems to be more about expressing emotions and opinions, with words such as "good," "bad," "im" (short for "I'm"), "yes," and "oh" indicating a more introspective and thoughtful discussion. It may include topics such as personal struggles, achievements, or reactions to events in the news or media.

Topic 2 revolves around gaming, with words such as "play," "game," "lol" (short for "laugh out loud"), and "fun" suggesting a discussion about favorite games, strategies, or experiences. It may also include debates about gaming culture or the industry in general.

- **Sentiment Analysis using Textblob Library**



t	date	time	length	tokenized	pos_tags	lemmatized	word_count	sentiment	sentiment_class	positive_words	negative_words	neutral_words
v	2023-04-04	11:55:00	3	[wow]	[(wow, n)]	wow	1	0.1	positive	wow		
n	2023-04-04	11:55:00	4	[guh]	[(guh, n)]	guh	1	0.0	neutral			guh
y s	2023-04-04	11:55:00	8	[hey, guys]	[(hey, n), (guys, n)]	hey guy	2	0.0	neutral			hey guys
s k	2023-04-04	11:55:00	11	[oh, its, back]	[(oh, n), (its, n), (back, n)]	oh back	3	0.0	neutral			oh its back
o	2023-04-04	11:55:00	2	[no]	[(no, n)]		1	0.0	neutral			no

Fig: Dataframe with sentiment_class and word colum for positive, negative and neutral

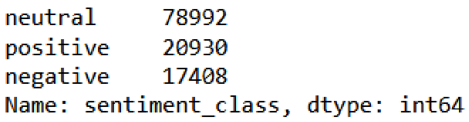
Positive class: Messages are related to favourable view or feeling towards the game.

Negative class: Messages are related to unfavourable or negative feedbacks.

Neutral class: Messages are related to casual conversation or socializing.

Based on polarity of each texts it is classified to sentiment_class based on the sentiment score.Then based on the score each content words are extracted to either of the class.

- **Count of Each class**



```
neutral     78992
positive    20930
negative    17408
Name: sentiment_class, dtype: int64
```

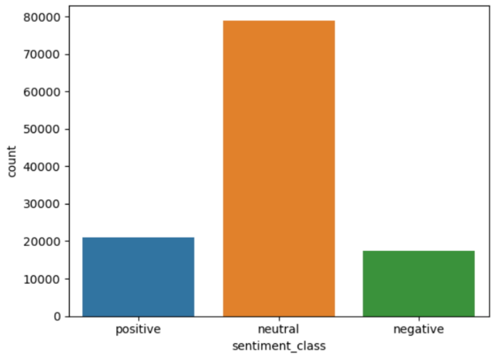
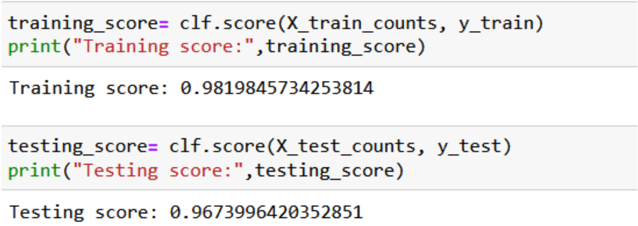


Fig: Count of each sentiment class

Most of the chats was of social conversations

- **Implementation of a simple text classification model using the Mutinomial Naive Bayes algorithm**



```
training_score= clf.score(X_train_counts, y_train)
print("Training score:",training_score)
```

```
Training score: 0.9819845734253814
```

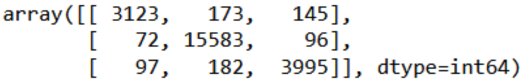
```
testing_score= clf.score(X_test_counts, y_test)
print("Testing score:",testing_score)
```

```
Testing score: 0.9673996420352851
```

Fig: Accuracy score of training and testing for Naive Bayes Model

Accuracy score of Training is greater than testing thus, we can validate absence of overfitting.

- **Confusion Matrix**

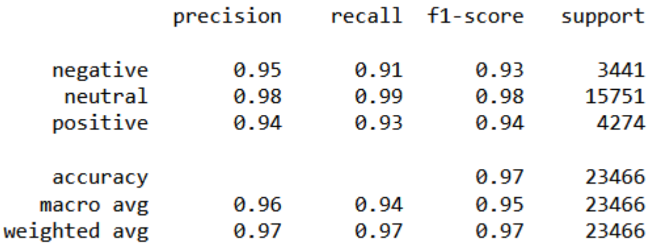


```
array([[ 3123,   173,   145],
       [   72, 15583,    96],
       [   97,   182,  3995]], dtype=int64)
```

Fig: Confusion Matrix

Model has performed well overall, with high counts of true positives and true negatives and relatively low counts of false positives and false negatives.

- **Classification Report**

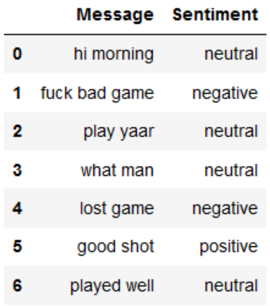


```
              precision    recall  f1-score   support

    negative       0.95      0.91      0.93      3441
     neutral       0.98      0.99      0.98     15751
    positive       0.94      0.93      0.94      4274

    accuracy                           0.97     23466
   macro avg       0.96      0.94      0.95     23466
weighted avg       0.97      0.97      0.97     23466
```

Fig: Classification Report

Classifier is performing very well, with high precision, recall, and F1-score for all three sentiment classes, and an overall accuracy of 0.97 on the testing data.

- **Validating the model on a set of new Random chats to identiy the nature of clusters**



	Message	Sentiment
0	hi morning	neutral
1	fuck bad game	negative
2	play yaar	neutral
3	what man	neutral
4	lost game	negative
5	good shot	positive
6	played well	neutral

Fig: Dataframe with preprocessed new messages and their predicted sentiments

Perfectly Classified using the pre defined Multinomial Naive Bayes model.s

Conclusion and Future Work

Based on the topic modeling, it appears that the conversations in the Apex Legends game chat server are multifaceted and encompass a variety of topics. The topics identified include expressing emotions and reactions while playing the game, frustration and negative experiences with the game, discussing gameplay mechanics and strategies, having fun while playing the game, and discussing the game in general.

Moreover, the Sentiment analysis using Textblob Library identified three distinct clusters, with one related to casual conversation or socializing, one related to negative sentiment or feedback related to the game, and one related to positive sentiment or feedback related to the game. These results suggest that players in the Apex Legends game chat server engage in a wide range of conversations and discussions related to the game, with both positive, negative and neutral sentiment expressed.

Naive Bayes model evaluate the performance of the classified sentiments with the preprocessed text (lemmatized) and thus, can be used to classify the chat of gamers. In addition to providing valuable insights for game developers, the analysis of conversations in the Apex Legends game chat server can also help understand the behavior of gamers and potential threats to the gaming community.

For example, the analysis can help identify instances of toxic behavior, such as harassment, bullying, or hate speech. Game developers can use this information to implement features that discourage or penalize such behavior, making the gaming environment safer and more enjoyable for everyone. Furthermore, the analysis of conversations can help identify potential vulnerabilities or weaknesses in the game, such as glitches or exploits that can be exploited by malicious players.

By addressing these issues, game developers can improve the overall integrity of the game and prevent cheating or other unfair practices that can harm the gaming community. Overall, the analysis of conversations in the Apex Legends game chat server can provide valuable insights for game developers, gamers, and researchers alike. By understanding the topics, sentiments, and behaviors expressed in the game chat, we can improve the gaming experience and create a more welcoming and inclusive gaming community.

References

[1] Liao, H., Fan, M., Huang, Y., & Liu, C. (2019). A survey of chat data analysis in online games. IEEE Access, 7, 165936-165946.

[2] Wang, Z., Li, J., Xu, H., & Li, H. (2018). Natural language processing techniques for social media data analysis: A survey. IEEE Access, 6, 53025-53045.

[3] Ismailov, F., & Vasileva, E. (2018). Analyzing player chat messages in online games using clustering. In Proceedings of the 2018 5th International Conference on Control, Decision and Information Technologies (CoDIT) (pp. 1039-1043). IEEE.

[4] Liu, Y., Zhang, X., Feng, Y., & Chen, W. (2018). Deep learning for sentiment analysis of online game reviews. In Proceedings of the 2018 IEEE International Conference on Big Data and Smart Computing (BigComp) (pp. 329-336). IEEE.

[5] Cheung, C. M., & Lee, M. K. (2010). A theoretical model of intentional social action in online games. Decision Support Systems, 49(1), 24-34.

[6] Li, J., Liang, Y., & Zhu, Y. (2019). Exploratory text analysis of student online discussion posts using topic modeling. In Proceedings of the 2019 IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE) (pp. 1284-1289). IEEE.

[7] Cai, Z., Wang, Y., Liu, Y., & Wei, Y. (2018). Topic models for analyzing students' learning processes in a collaborative learning environment. IEEE Transactions on Learning Technologies, 11(1), 64-75.

[8] Patel, P., & Patel, N. (2019). Sentiment analysis of movie reviews using textblob and Naive Bayes classifier. In Proceedings of the 2019 2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT) (pp. 810-813). IEEE.

[9] Kasurinen, J., Hietajärvi, L., Kinnunen, J., & Kankainen, A. (2020). Social network analysis of massively multiplayer online game player chat data. Entertainment Computing, 34, 100330. doi: 10.1016/j.entcom.2020.100330

[10] Xiao, R., Huang, Y., Yang, C., Wang, X., & Chua, T. S. (2019). Hierarchical graph convolutional networks for analyzing social behavior in massively multiplayer online games. IEEE Transactions on Games, 11(4), 368-377. doi: 10.1109/TG.2019.2901