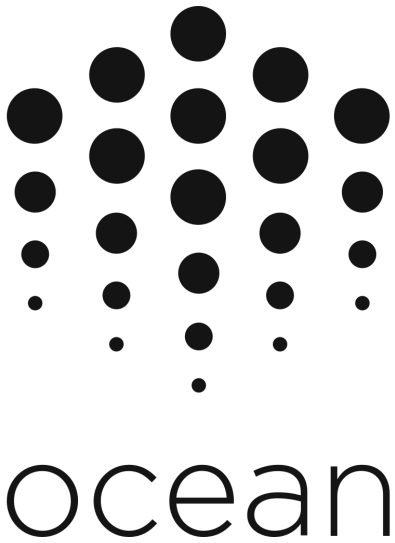
Ocean Discord Community Dynamics analysis

Hugo Moura

March 11, 2024



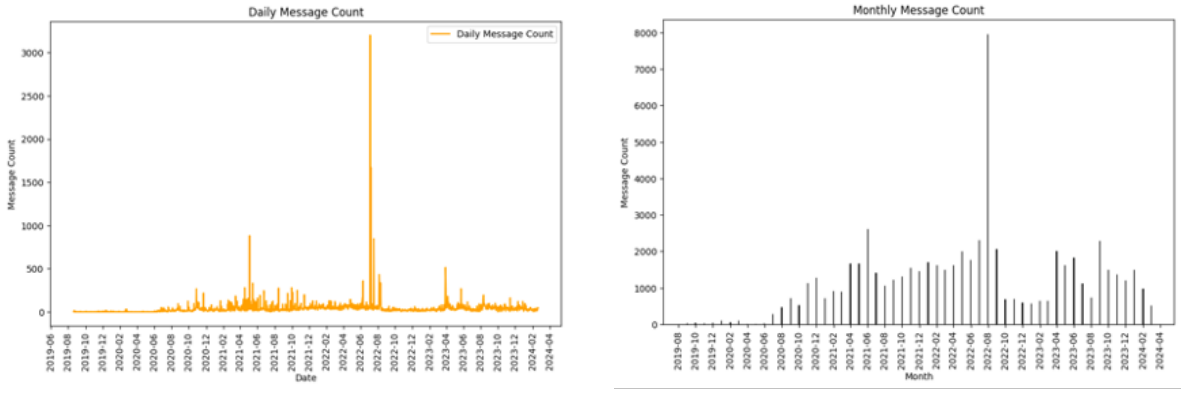
Contents

1 Introduction

Welcome to our exploration of the Ocean Protocol's Discord community. Our analysis is focused in discovering how the community interacts within the Discord platform. The goal is to share practical findings that impact the community's growth and direction. This challenge invites us to understand community behavior, offering a chance to apply analytical skills to real-world data. This report begins by examining general trends, showing the evolution of messages over time to identify patterns and outliers. Following this, the correlations section investigates the relationship between $OCEAN price and sentiment, server activity, messages, new users, and active contributors. community questions categorizes frequently asked questions on themes. Community activity ranks the most active contributors, analyzing peak activity times. The Scam & Spam analysis section introduces a machine learning model for identifying potential spam messages in order to maintain a secure community environment. Technical issues categorizes common technical questions, their sources and category. Lastly, the prediction model develops a forecasting tool, predicting future server activity being versatile in applications beyond the Ocean Protocol community. Each section delivers concise findings, contributing to the understanding of the $OCEAN community discord dynamics.

2 Data Analysis : General Trends

The two presented graphs, show daily and monthly message counts within the Ocean Protocol's Discord community, revealing patterns in user engagement. In both graphs, a peak/outlier in communication is evident during July-August 2022, marked by a surge of approximately 8000 messages. This indicates a pivotal event or discussion during that specific period marked by the market volatility and cryptocurrency crash in the middle of 2022 (Figure 2). Concurrently, the elevated message counts in May-June 2021 underscores also a trend of activity during that month. Furthermore, the monthly graph accentuates a substantial uptick in message counts for April and August 2023.



(a) Overall daily messages in Discord Server

(b) Overall monthly messages in Discord Server

Figure 1: Overall messages in Discord Server

For helping purposes the graph for the $OCEAN price is presented highlighting the three spotted peaks in user engagement present in Figure 1.

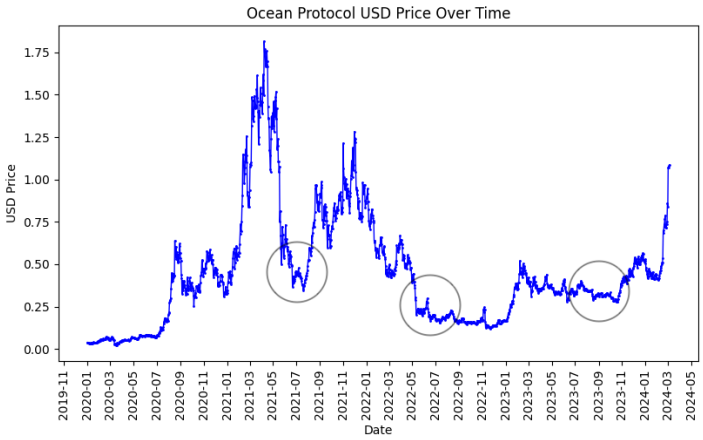


Figure 2: $OCEAN price chart

In our comprehensive analysis of Discord channels, we zoomed in on the trends observed in four channels aiming at understanding market dynamics and overall message counts. These channels, namely "GET STARTED -\gm," "GENERAL -\general-chat," "GET STARTED -\introduce-yourself," and "ECOSYSTEM -\ambassadors," were selected based on their anticipated correlation with market trends. Our investigation specifically delved into the monthly message count trends within each of these channels.

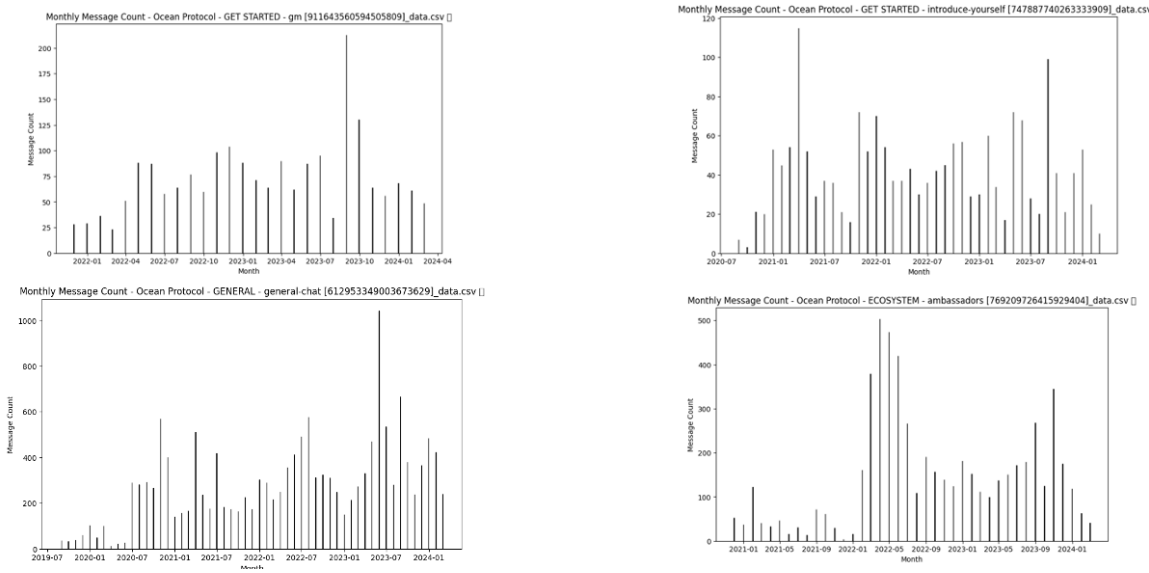
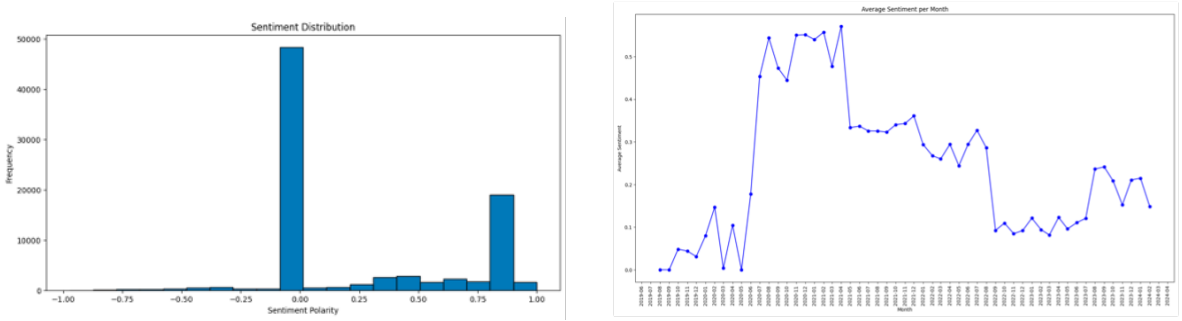


Figure 3: Channels analysis

For the channel "GET STARTED - \gm," we observe that its peak is in September 2023, correlating with the overall graph message count per month, where we also observe a peak of around 225 messages. Similarly, for the other channels, we can see that July, September, and October were months where users were very active. In the case of the "GET STARTED -\introduce-yourself" channel, a distinctive peak occurs around April 2021, aligning with a market peak. This intriguing observation implies that users tend to be more inclined to introduce themselves during periods of positive market performance. These insights contribute to our understanding of user behavior and preferences, providing valuable context to the broader trends identified in our analysis for trends in channels.

3 Data Analysis : Correlations

In this section, we focused in the correlations between specific metrics and relationships between the price of $OCEAN. Key metrics such as sentiment, the number of messages, new users, and active individuals on the server. The aim was to uncover potential insights into how market dynamics and user engagement may act together exploring correlations. First we took a look into the sentiment evolution within the community, presented in figure 4.



(a) Sentiment distribution

(b) Average sentiment per month

Figure 4: Overall sentiment in Discord Server

Correlations values are analysed merging the datasets. The values range from -1 to 1, where 1 indicates a strong positive correlation, -1 indicates a strong negative correlation, and 0 indicates no linear correlation. It's interesting to find a correlation between sentiment and the $OCEAN price in the community. A calculated correlation value of 0.7 indicates a strong positive correlation, suggesting that as the sentiment in the community increases, the $OCEAN price goes up. On the flip side, when the market is not doing well, and people talk about prices going down and express worries, it can affect how everyone feels in the community. This back-and-forth interaction shows how people feel and market impact connection. It's like a dance where both the community's mood and the market's ups and downs influence each other. Understanding this connection can help us see the complex relationship between how people in the community feel and what's happening with the cryptocurrency prices in the Ocean Protocol world. Here some examples of calculated sentiment for different messages:

```
Looks superb Thank you | 0.765
im not even sure what this server's for tbh | - 0.2411
```

Furthermore, the correlation of -0.37 between the message count and the $OCEAN price implies a negative relationship between the two variables. In practical terms, as the number of messages in the community increases, the $OCEAN price tends to decrease, and vice versa. While not a strong negative correlation, this insight suggests that there might be some degree of inverse influence between community activity and the cryptocurrency's price. It's worth exploring the reasons behind this relationship, increased discussions are driven by concerns during price declines or if high community activity somehow contributes to market movements.

```
Correlation between message count and price: -0.3711800943094737
Correlation between sentiment and price: 0.6950579024151842
Correlation between new users and price: -0.3439570177200358
```

In our analysis of the Ocean Protocol community, an interesting connection emerges between the number of new users and the $OCEAN price. Surprisingly, the value of -0.34 show that, when the cryptocurrency's price experiences a decline, we observe an increase in the count of new users joining the community.
In our data analysis, a distinctive metric caught our attention — the frequency of the phrase "joined the server" appearing in the content of messages within the Ocean Protocol community. Here the two highest points, in 2021-05 and 2022-07 follow the two crashes in the $OCEAN price.

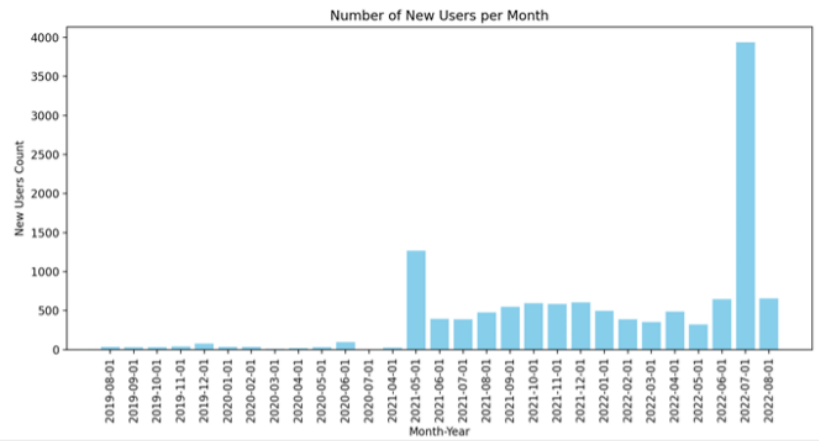


Figure 5: Number of new users in the server per month

4 Data Analysis : Community activity

In the course of our analysis, we delved now into the number of messages exchanged within the Discord server. Expanding upon this initial exploration, our focus now shifts towards an interesting examination of user engagement by ranking individuals based on diverse metrics. These metrics follow the quantity of messages sent, words written, characters utilized, attachments shared, and reactions received. Our goal is to reveal patterns of user interaction, focusing on the most active contributors in different dimensions.
The 5 most active users with the respective number of interactions are the following:

User	Message Count
MEE6#4876	17878
Deleted User	5825
blockchainlugano	1199
kreigdk	1116
dotunwilfred.eth	1048

Table 1: Message Counts by User

Important to note that here, the number of interactions are specifically the number of times a user interacts with content, the attachements or reactions doesn't count for our analysis. Here's the summary for the first 10 users average daily message count:

Rank	User	Message Rate (messages per hour)
0	MEE6#4876	15.385542
1	Deleted User	12.085062
2	bigimeyagazzz	9.678571
3	OceanDiffusion#4502	7.494118
4	OceanGPT#0740	5.473684
5	alexcos20	4.580247
6	blockchainlugano	3.090206
7	bhavingala	2.546599
8	doteth	2.538760
9	denkobetona	2.363636
10	birususama	1.701587

Table 2: User Rankings by Message Rate

Displayed below is a plot illustrating the number of messages over time for the 19 most active users. While identifying a clear pattern from this graph may be challenging, participation from these top

users is evident, particularly during the middle of the year 2022. This spike in activity aligns with the second market downturn in the timeframe of our analytical period.

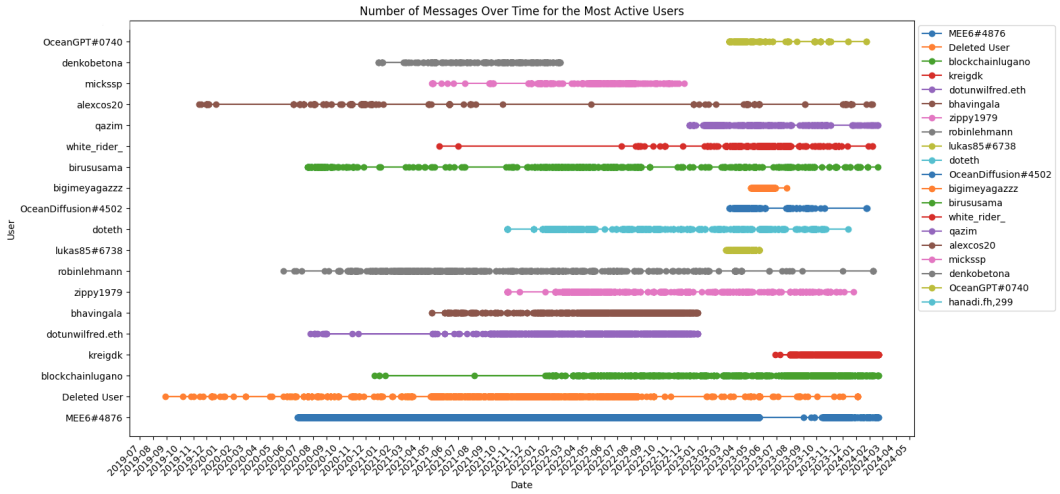


Figure 6: Example users graph over time

Here also presented the date where each user was the most active and we can clearly see that the year with most activity was 2022 appearing 7 times out of 11 most active users.

Author	**Most Active Date**	**Number of Messages**
MEE6#4876	2022-07-05	1556
Deleted User	2022-07-05	1220
lukas85#6738	2023-03-29	198
OceanDiffusion#4502	2023-03-29	119
blockchainlugano	2023-08-11	42
kreigdk	2023-12-28	31
zippy1979	2022-03-02	25
robinlehmann	2022-06-22	28
doteth	2022-02-09	18
dotunwilfred.eth	2022-07-22	13
bhavingala	2022-04-08	11

Table 3: Users Statistics

It's apparent that the users identified show patterns that are raising suspicion of potential bot-like behavior. To ensure the accuracy of subsequent analyses and conclusions we transition to the bot detection section and have a non bot analysis giving more important insights.

Author	Most Active Hour	Message Count
Deleted User	22	718
MEE6#4876	21	1411
OceanDiffusion#4502	11	71
OceanGPT#0740	4	64
alexcos20	5	69
bhavingala	13	238
bigimeyagazzz	15	66
birususama	20	90
blockchainlugano	19	120
denkobetona	4	50
doteth	23	53
dotunwilfred.eth	22	283
kreigdk	22	109
lukas85#6738	4	139
mickssp	2	63

Table 4: User Activity Statistics by Hour

4.1 Bot detection

In the bot detection process, the analysis began with the utilization of a machine learning model trained on a Twitter dataset found in Kaggle, distinguishing between bot and non-bot users based on various features. Subsequently, this trained model was applied to the Discord dataset, specifically focusing on the 'Content' column of messages. The code systematically transformed and vectorized the text data, enabling the model to predict whether a user's messages exhibited bot-like behavior. The unique bot authors, identified through this analysis, were stored in a CSV file. This file contains a list of distinct bot authors and their corresponding bot labels, contributing valuable insights into the presence of bots within the Discord dataset. The analysis offers a perspective on potential bot activity, allowing deeper and further investigation of the platform's user experience.
In the presented table, a subset of bot detection results is showcased, specifically highlighting ten users out of a total of 2,597 identified as potential bots. The "IsBot" column indicates whether each user is classified as a bot (1) or not (0). This comprehensive machine learning bot detection process involved analyzing content and user features, enabling the identification of accounts exhibiting automated or suspicious behavior within the given dataset.

Author	IsBot
mantisclone	1
white_rider_	1
AutoBot#1760	1
dudeamir	1
alexcos20	1
doteth	1
malgamoe	1
Deleted User	1
MEE6#4876	1
OceanDiffusion#4502	1

Table 5: Bot Detection Examples Results

4.2 Non Bot User Activity

The exclusion of bot users from the analysis contributes to a more precise understanding of genuine user interactions within the Ocean Discord community. For that a script was made to determine activity of non bot users. It groups the data by user ('Author') and counts the messages for each user, most active hour, most active month and type of user creating a DataFrame with the user's name and the corresponding message count. The following table represent the first 10 non bot users in the channel.

Author	MessageCount	YearMonth	Hour	Type
witchesrune	15	2022-08	8	Morning User
laurentony15#2825	8	2023-04	22	Night User
cryptoross_	6	2022-01	0	Night User
raiden_kev	5	2022-08	5	Morning User
Mason#5694	5	2020-06	12	Evening/Afternoon User
Neeeeira#9526	5	2022-08	5	Morning User
OthersideMeta#4783	5	2021-12	21	Night User
thesamap#7822	4	2023-08	23	Night User
anthonyhsiao.	4	2023-12	19	Evening/Afternoon User
irokesenigor	4	2021-05	20	Night User

Table 6: Non-Bot User Activity Results

To provide a more nuanced understanding of their engagement, we calculated the average daily message count for each user, considering only the days on which they participated in discussions. This approach helps to capture the regularity of user engagement while excluding days with no activity, resulting in a more accurate representation of their messaging habits. We observe that non-bot users typically participate in conversations by contributing in messages for just one day. The data indicates that these users tend to contribute messages on specific days rather than consistently over time.

Author	Avg Daily Message Count
witchesrune	15.0
Neeeeira#9526	5.0
raiden_kev	5.0
irokesenigor	4.0
thesamap#7822	4.0
Gru (Matt Wilson)#3202	4.0
Mason#5694	2.5
Mattericks#5344	2.0
rider_7978	2.0
anthonyhsiao.	2.0

Table 7: Top 10 Most Active non-Bot Users - Average Daily Message Count

In Figure 7, the visualization illustrates patterns of the top 30 non-bot users over time. Notably, there is a substantial surge in activity during the years 2021 and 2022, coinciding with the period of the Covid-19 pandemic and market volatility. This increased interaction suggests a high level of communication and participation, potentially driven by the unique circumstances and events during those years.

The table 8 shows key metrics for the top 5 authors in a Discord community, in terms of char count. It highlights their average daily message count, total message count, cumulative word count, and total character count. It underscores the importance of analyzing not only the most active users but also those who demonstrate dedication through meaningful contributions in terms of word and character count.

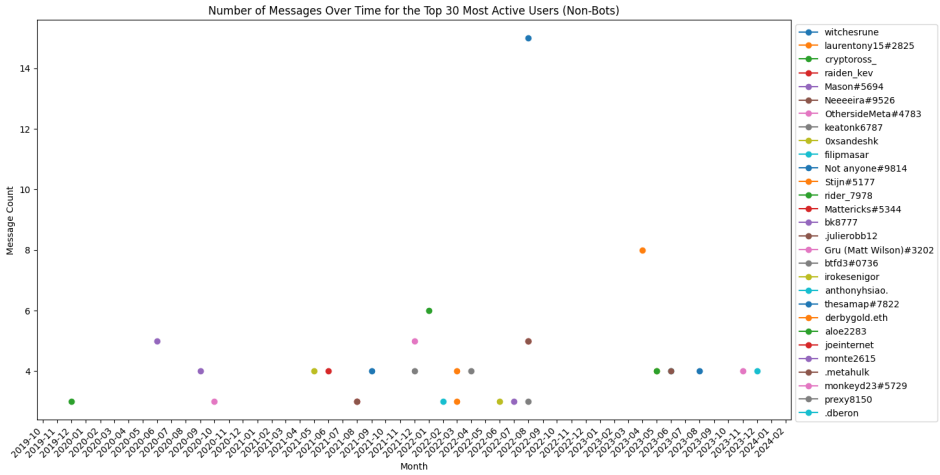


Figure 7: Non Bot users graph over time

Author	Avg Daily	Msg Count	Word Count	Char Count
poseidon2525	1.0	2	404	2485
Mason#5694	2.5	5	406	2350
awbvious	1.0	3	385	2224
bk8777	1.33	4	339	2056
Miriam RAZ#9864	1.5	3	244	1655

Table 8: Top 5 Authors with more characters count

5 Data Analysis : Community questions

To understand community questions we analysed the Ask AI channel and filtered out the questions. Then we look into the messages and pick out the words that come up the most. These words help us see what topics are more popular across the community and raise more questions.
We organize these words into themes like technical, price-related, and general information. A question, by default is set as a general information question however it needs to pass a “test” to determine if the question is technical or price-related before hand. This decision is based on whether certain words related to technology or prices are present.
Through this graph we can observe that general information questions are the more frequent with a frequency of around 160 messages followed by technical and price related questions respectively.

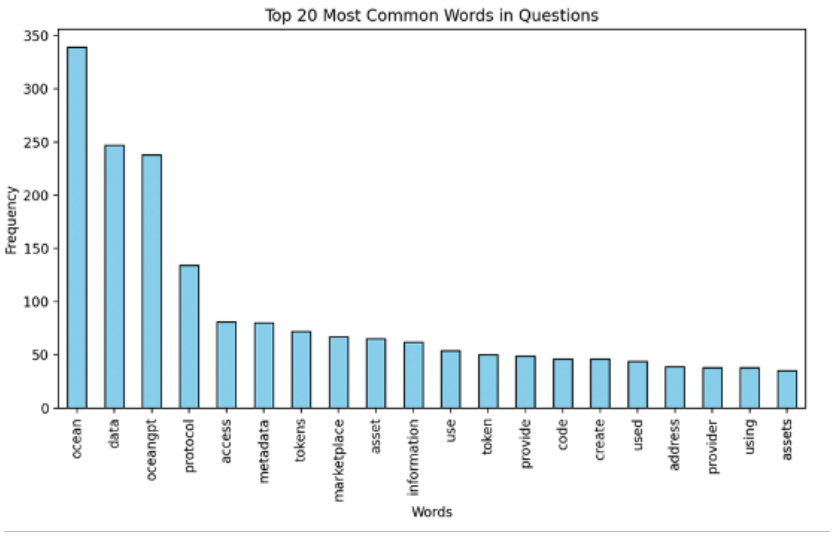


Figure 8: Top 20 words in questions

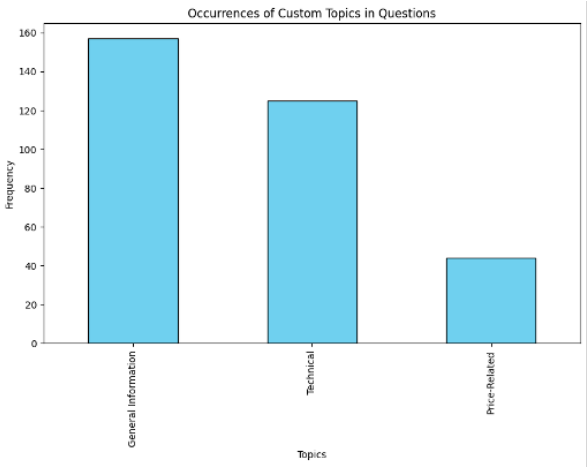


Figure 9: Occurrences of custom topics in questions

6 Data Analysis : Scam & Spam

In this spam detection analysis, we initially prepared and trained a machine learning model using a spam dataset containing SMS examples (5,574 messages) , where 'ham' represented non-spam messages and 'spam' indicated spam, this csv was extracted from Kaggle. After splitting the data and vectorizing the text content, a Multinomial Naive Bayes classifier was trained, achieving a certain accuracy on the test set. Choosing the Multinomial Naive Bayes (MNB) classifier for spam detection in the Discord dataset is appropriate for several reasons. Firstly, the initial training of the model on a well-established spam dataset provides a solid foundation. This dataset, sourced from Kaggle, offers diverse examples of spam and non-spam messages, enabling the MNB classifier to learn patterns effectively. Moreover, the efficiency of the MNB algorithm make it well-suited for text classification tasks, especially when dealing with a relatively large dataset of textual messages. Its ability to handle text data by considering word frequencies aligns with the nature of Discord messages. Applying this model to the actual Discord dataset, we predicted and classified messages as spam or non-spam. The results were saved in a CSV file. To gain insights into the nature of the misclassifications, we explored the most common words in messages labeled as spam, visualizing the top 20 in a bar chart. This analysis provides a fundamental understanding of spam detection on the Discord platform.

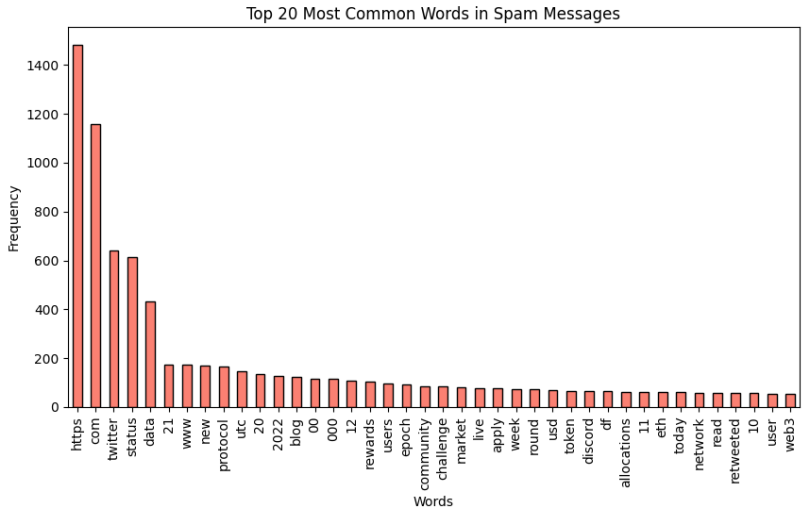


Figure 10: Top 20 most common words in spam messages

To refine the analysis, we excluded words containing "ocean," which may have affected results. Notably, spam often incorporates website references, signified by the presence of keywords like "http," and "com." Additionally, social media references, including terms like "twitter" are frequently encountered. Numerical elements further characterize spam, with a prevalence of numeric characters evident in the content. Furthermore, the use of appealing and attractive language, exemplified by words such as "rewards," "new," and "users," is also present.

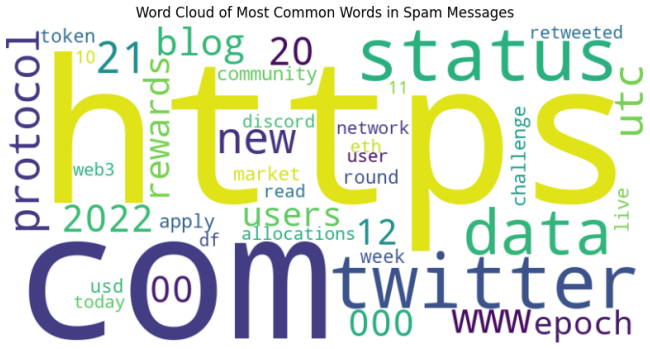


Figure 11: Word cloud in spam messages

Acknowledging the limitations of our spam detection analysis, it is important to recognize that the initial machine learning model was trained on a dataset composed of SMS messages, as sourced from the dataset on Kaggle. While this dataset provides a valuable foundation for training a spam detection model, it may not perfectly represent the format of messages exchanged within the Discord platform. Discord conversations often involve unique language, slang, and context that may differ from typical SMS interactions. As such, misclassifications in the model's predictions on the actual Discord dataset could be influenced by these differences. To improve model's accuracy in the context of Discord interactions, future iterations could benefit from training on a more Discord-specific dataset.

7 Data Analysis : Technical Issues

We refined the classification process by exploring technical questions. The search incorporate terms associated with pricing to form a "Price-Related" category, capturing discussions on costs, values, buying, and selling. The existing categories, including "User-Related," "System-Related," and "External Factors," were analysed with a more extensive set of keywords to ensure comprehensive identification of relevant terms. Despite these refinements, a general category labeled "Other" remains, spotting words and content that do not meet the specific criteria for the more defined categories.

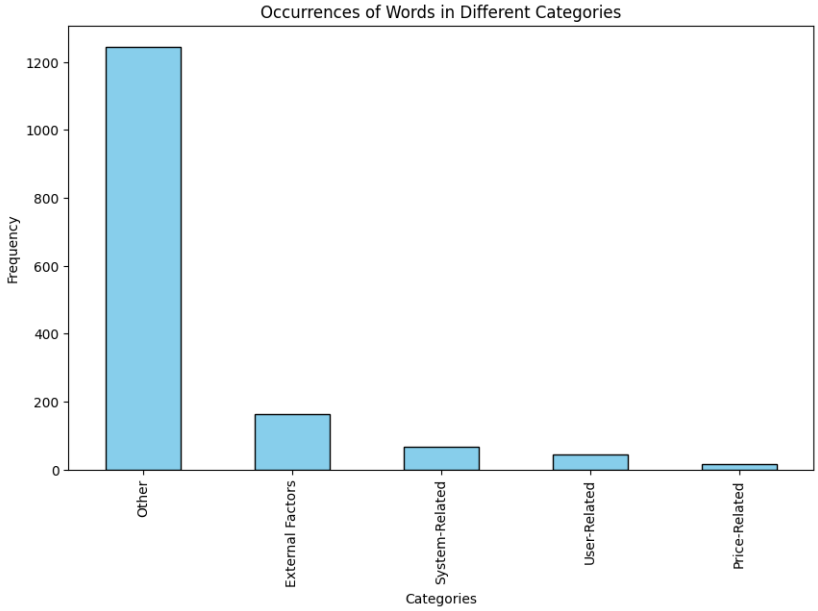


Figure 12: Categories for technical topics in questions

Integrating a CSV file containing data from the "tech-issues" channel in the Ocean Discord server would be highly beneficial for future tasks. This addition would provide valuable insights and facilitate a more comprehensive analysis in categorizing the technical topics.

8 Prediction model

Forecasting future server activity is crucial for efficient resource allocation, planning, and ensuring a enjoyable user experience. In this step, the aim is to develop robust forecasting models to predict server activity within the Ocean Protocol Discord server. By leveraging historical user activity data, our goal is to implement a model that not only adapts to unique characteristics of the Ocean Protocol Discord server but can also serve as a template for similar forecasting tasks in others online communities.
Forecasting server activity within a Discord community or any social platform focused on cryptocurrency discussions is a challenging task. The nature of user interactions, coupled with crypto market volatility, presents unique challenges such as handling fast peaks in activity, capturing trends influenced by market trends, and integrating external factors like regulatory changes. Therefore, the ability to predict user engagement patterns allows efficient moderation strategies.

8.1 NeuralProphet Forecasting Analysis

In this analysis, we employed the NeuralProphet library to construct a forecasting model for monthly message counts.
The choice of NeuralProphet for this analysis is justified by its ability to handle time-series data with complex patterns and nonlinear relationships, characteristics commonly found in Discord community dynamics. Additionally, its neural network architecture allows it to automatically adapt to the data without the need for extensive feature engineering.

After loading and preparing the dataset, which contains 'Date' and 'MessageCount'(monthly) columns, we initiated a NeuralProphet model and trained it on historical data. The initial visualization displayed the actual message count over time, figure 13. Subsequently, the model generated predictions for both historical (red line) and future periods (blue line), figure 14. The visualizations showed the accuracy of the model's predictions, with separate plots for actual historical data, predictions on historical data, and future predictions. The Mean Absolute Error (MAE) was calculated to quantify the accuracy of the model's predictions on historical data. Furthermore, the lengths of actual and forecasted data were checked for consistency, and if applicable, the MAE on future data was calculated. The results were the following:

```
Mean Absolute Error on Historical Data: 387.8026639071378
Length of Actual Data: 55
Length of Forecast Data: 55
Mean Absolute Error on Future Data: 1310.4529430042614
```

In this case, a lower MAE for historical data suggests that the model fits well to the observed historical values. However, a higher MAE for future data may indicate that the model struggles to generalize to unseen data points.

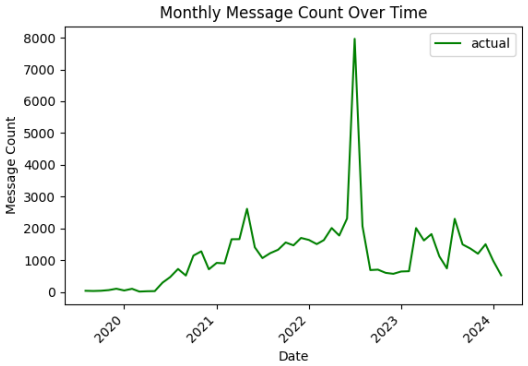


Figure 13: Actual Message count

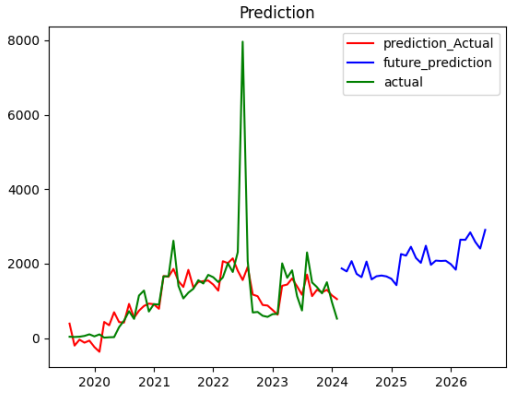


Figure 14: Prediction results

The predictive analysis suggests a gradual and periodic increase in message count over the years. The observed pattern indicates a positive trend.
Moreover we perform predictive analysis on monthly sentiment data. The 'actual' sentiment values are visualized initially, figure 15, showing the trend. The NeuralProphet model is then trained again on this historical sentiment data, and future predictions are made for the specified period.

```
Mean Absolute Error on Historical Data: 0.035874811831528755
Length of Actual Data: 55
Length of Forecast Data: 55
Mean Absolute Error on Future Data: 0.2614141053093013
```

For the historical data, the MAE is low at 0.036, indicating a close match between the actual sentiment values and the predictions. The length consistency between the actual and forecast data (both 55) adds confidence to the reliability of the model's performance on past sentiment trends.
Looking at future predictions, the MAE of 0.261 suggests a reasonable level of accuracy in predicting sentiment for the upcoming period.

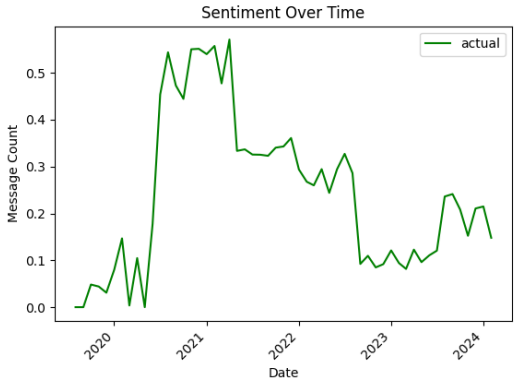


Figure 15: Actual Message count

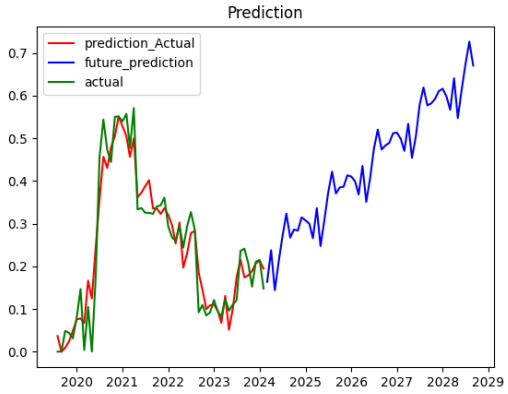


Figure 16: Prediction results

Based on the model, we can see a upcoming positive trend in sentiment across the Discord community. As established earlier, the correlation between sentiment and price is approximately 0.7, indicating a strong relationship. Therefore, this positive sentiment forecast can potentially serve as a valuable indicator for forecasting the future movements of the $OCEAN cryptocurrency, allowing stakeholders,for example, to make informed decisions based on the community sentiment prediction.

9 Conclusion

In conclusion, our analysis of the Ocean Protocol Discord community has revealed insights into user engagement and its correlations with various metrics. User messages, when examined alongside price trends, revealed interesting negative correlations. We successfully identified key patterns in user engagement, shedding light on the correlation between message count, sentiment, and the influx of new users. Additionally, our exploration into frequently discussed topics and prevalent technical questions has provided a deeper understanding of the community's dynamics. The predictive modeling further forecasts a positive trajectory in sentiment and a gradual increase in message count across the server. For future tasks, having more data and information is also important for data analysis tasks, providing up-to-date insights in users interaction. Data scraping from the Discord channel is essential for ensuring the relevance and efficacy of future analytical tasks. Also, incorporating bigrams in text analysis could be very usefull. Bigrams provide a accurate view, allowing the model to discover associations that contribute to a more accurate representation of language. Because combined meaning is different from the sum of their individual parts.It could gain a richer contextual understanding, leading to improved accuracy in tasks such as sentiment analysis, topic modeling, and language modeling. Furthermore, refining predictive models to achieve higher accuracy and lower means stands as a promising avenue for future work, ensuring more robust forecasting capabilities.

Github repository : https://github.com/hugomoura10/discord_analysis_ocean