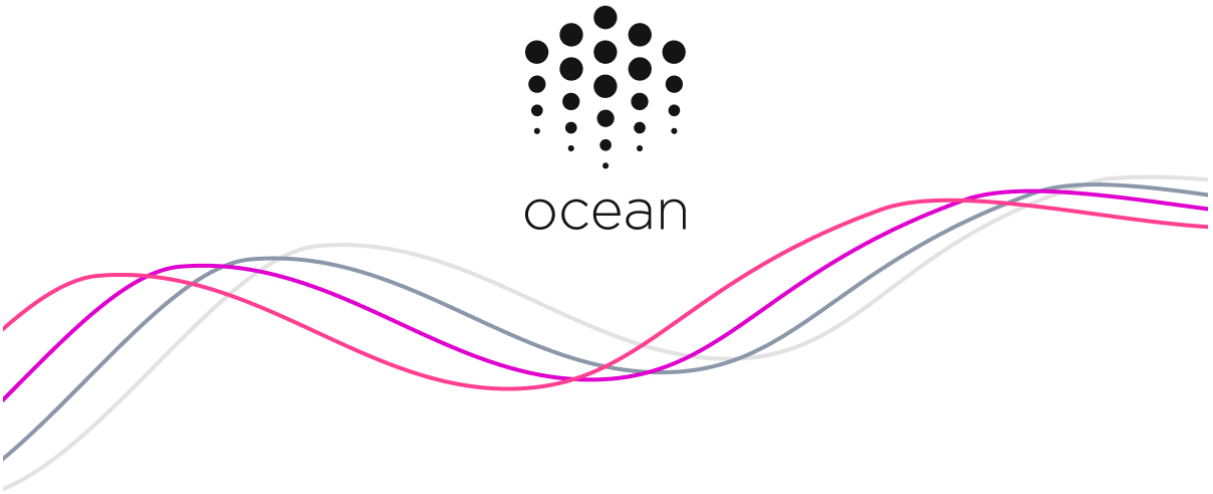
OCEAN Token Sentiment Analysis Challenge Part 2



Nicolas Landry

2023 - 22 - 07

Table of contents

Introduction

As a data scientist, the complexity of the cryptocurrency market piques my interest. It's a fascinating interplay of various factors, including social media sentiment, which makes it an intriguing field for detailed analysis. For this study, I've focused on the Ocean Protocol's token, OCEAN, which has an intricate connection with market dynamics and social media interactions.

In a previous investigation (the first part), I embarked on an analysis of social media sentiment, evaluating the correlation between OCEAN's price and different facets of social media engagement such as the volume of tweets, likes, retweets, and the number of individuals engaging with the cashtag. Additionally, I assessed the influence of prominent tweets on the price of OCEAN. These insights underscored the substantial role that social media sentiment and high-profile tweets or news events play in dictating the market movements of crypto tokens. I found the application of TextBlob for sentiment analysis within a machine learning context to be particularly effective in classifying tweets and distilling key themes within different sentiment categories.

Encouraged by these findings, I resolved to deepen my exploration and progress from analyzing to predicting the price of OCEAN. I devised a Linear Regression model in Python to scrutinize the effects of the Ethereum price, social media engagement (measured by the number of tweets), and the average daily sentiment on OCEAN's price. This project has been meticulously structured, encompassing the phases of data preparation, model training, performance evaluation, and visualization of the predictions.

It's an honor to participate in this second segment of the data competition, and I thank everyone involved. Now, let's revisit the questions that were posed for this stage of the project:

- QUESTION 1:

  Identifying three key factors that could potentially influence the price of the OCEAN token, considering social media engagement

- QUESTION 2:

  Provide a detailed description of the data sources you would consider for each identified factor. Be specific about the datasets, APIs, or platforms you would utilize to obtain the necessary data for your analysis.

- QUESTION 3:

  Design a methodology to measure and analyze the relationship between social media engagement and the price of the OCEAN token. Explain how you would collect and preprocess data to conduct sentiment analysis on platforms like Twitter or any other relevant social media platform.

- QUESTION 4:

  In addition to social media sentiment, outline their approach for exploring the impact of the external factors you identified in question 1. Describe necessary data transformations, calculations, or statistical techniques you would employ to analyze the relationships between these factors and the OCEAN token price.

- QUESTION 5:

  Develop a machine learning model that can provide insights into the relationship between social media engagement and the price of the OCEAN token.

- QUESTION 6:

  Describe the ML model you chose and explain why it suited this task. Outline the steps involved in training, evaluating, and interpreting the model's predictions. Include details such as the choice of algorithms, feature engineering techniques, model training methodology, and any considerations for handling potential challenges, such as data imbalance or overfitting. Explain how the ML model contributed to your analysis and supported your findings in the report.

Methodology

In this study, I pursued a systematic methodology with a focus on hypothesis development and testing.

The main hypotheses revolved around potential factors influencing the OCEAN price, including Ethereum prices and various Twitter metrics. These hypotheses were thoroughly tested using correlation analysis, ensuring the robustness of my findings.
From here, I developed several Linear Regression models, gauging their performance with appropriate statistical metrics to select the most reliable predictor for the OCEAN price.

In the final phase, I refined my analysis, discarding any unvalidated hypotheses and less efficient models, to maintain a stringent standard of accuracy and reliability. This methodology enabled me to focus solely on my most impactful findings for this final report.

Data analysis

1. Key factors that influence the price of the OCEAN token

Market Trends (Including BTC and ETH Price Movements):

The prices of BTC and ETH often dictate the trend for the whole crypto market due to their dominance and high market capitalization. Cryptocurrency prices, including $OCEAN, tend to follow the price trends of BTC and ETH. If BTC and ETH are bullish, individual cryptocurrencies may also see price increases and vice versa. This is because BTC and ETH are often seen as the bellwethers of the cryptocurrency market, and their performance can impact investor sentiment towards other cryptocurrencies, influencing buying and selling pressures. Therefore, monitoring the price and trading volumes of BTC and ETH can provide insight into potential price movements of the OCEAN token.

Regulatory News:

Regulatory Changes Impact on the whole market

The fluctuation of cryptocurrencies' value is often closely tied to changes in governmental policies and regulations.

A prominent instance of this occurred in 2017, when China's prohibition of financial services related to cryptocurrencies led to a precipitous 50% drop in Bitcoin's price in the subsequent weeks. Contrarily, the approval of the first Bitcoin futures ETF by the United States Securities and Exchange Commission (SEC) in 2020 prompted a Bitcoin price surge, contributing to an increase over the following months.

Additional cases of regulatory impacts on cryptocurrency prices include:

- In 2018, an announcement by the Indian government about a potential ban on cryptocurrencies spurred a market sell-off, triggering a more than 20% drop in Bitcoin's price in the days following the announcement.
- In 2019, the publication of a draft regulation for cryptocurrencies by the European Union, which was largely viewed positively within the industry, led to a price increase in various cryptocurrencies, with Bitcoin's price seeing an over 50% increase in the months that followed.
- In 2021, the United States government's clarification on cryptocurrency taxation was seen as a positive development, leading to a rise in cryptocurrency prices.

Regulatory Factors for Ocean Protocol

Given Ocean Protocol's specific function as a data exchange platform, the protocol faces unique regulatory challenges and opportunities. For example, governmental restrictions on data usage due to privacy concerns, or on AI due to fears of misuse, could pose significant risks.

On the other hand, the protocol could benefit from regulatory shifts that promote data transparency and sharing, which could increase the demand for Ocean Protocol's platform and consequently, drive up the price of OCEAN tokens.

In conclusion, given the dynamic regulatory landscape for both the broader cryptocurrency market and Ocean Protocol, investors should remain alert to potential regulatory risks and opportunities that could influence OCEAN's price.

Technological developments:

The price of $OCEAN may be influenced by significant updates or developments in the underlying technology or platform. For instance, the launch of a new service, the announcement of a partnership, or improvements in the technology can increase confidence in the project and potentially drive the price up.

Here are some examples of how technological developments have impacted the price of $OCEAN:

- In 2021, Ocean Protocol partnered with Orbis to launch end-to-end encrypted messaging on Ocean Market. This partnership was seen as a positive development by the community, and the price of $OCEAN rose by over 20% in the following weeks.

- In 2022 :
    - Ocean Protocol partnered with SmartPlaces to unlock data monetization for Web3 social interaction app. This partnership was also seen as a positive development, and the price of $OCEAN rose by over 15% in the following weeks.
    - Ocean Protocol shipped Ocean V4, which included Data NFTs to clarify intellectual property rights and help publishers and marketplaces monetize. This was a major development, as it made it easier for people to sell and buy data. The price of $OCEAN rose by over 50% in the following weeks.

- In 2023, Ocean Protocol entered its next phase: to drive data value-creation loops by focusing on the users in the last mile: data dapp developers, data scientists, and data-oriented crypto enthusiasts. This is a significant development, as it will make it easier for people to use the Ocean Protocol platform to monetize their data.

These factors are not only relevant because they directly impact the supply and demand of the $OCEAN token, but also because they influence market sentiment, which plays a key role in the price dynamics of cryptocurrencies. Please not that because the price rose after an announcement does not mean it is directly caused by this, but it will and can have participated in it, especially for a token related to a strong project like OCEAN.

2. Data sources identified for these factors

Social Media Engagement:

- **Twitter API**: https://developer.twitter.com/en/docs

Twitter API: The Twitter API is a good choice for collecting tweets related to $OCEAN. It provides access to tweet data including the content, date, retweets, likes, and replies. This can help to measure the volume of social media engagement and conduct sentiment analysis.

- **Reddit API**: https://www.reddit.com/dev/api/

Reddit API: The Reddit API is another good choice for collecting social media engagement data. It provides access to post and comment data from subreddits such as r/cryptocurrency and r/OceanProtocol or even r/OceanProtocol_news. This data can be used to measure sentiment and engagement, and to identify trends in the community.

- **Telegram API**: https://core.telegram.org/api

Telegram API: The Telegram API can be used to gather additional sentiment data from Telegram groups dedicated to Ocean Protocol. This data can be used to supplement the data collected from Twitter and Reddit.

Market Trends (Including BTC and ETH Price Movements):

- **CoinGecko API**: https://www.coingecko.com/en/api

CoinGecko API: The CoinGecko API is a comprehensive API that provides access to current and historical price data, market cap, trading volume, and other relevant data for a wide range of cryptocurrencies, including BTC and ETH. This data can be used to understand market trends and to identify potential opportunities for investment.

- **CoinMarketCap API**: https://coinmarketcap.com/api/

CoinMarketCap API: The CoinMarketCap API is another comprehensive API that provides access to similar data as CoinGecko.

- **Kraken API**: https://www.kraken.com/features/api

Kraken API: The Kraken API provides access to market data from the Kraken cryptocurrency exchange. This data can be used to track price movements and to identify trading opportunities.

Regulatory News:

News API: A general-purpose news API like GNews can be used to track news articles related to cryptocurrency regulations worldwide. This data can be used to stay up-to-date on regulatory changes that could impact the price of $OCEAN.

- **GNews API**: https://gnews.io/

Regulatory authority websites: Many regulatory bodies publish news and decisions related to cryptocurrencies on their websites. By monitoring these sites, we can get firsthand information about regulatory changes.

Cryptocurrency news websites: There are a number of cryptocurrency news websites that publish articles about regulatory developments. These websites can be a valuable source of information for tracking regulatory news.

Technological Developments:

- **Ocean Protocol's official website and blog**: https://oceanprotocol.com/

Ocean Protocol's official website and blog: Information about new releases, partnerships, and significant project developments is usually published on the official website or blog. This data can be used to track the progress of the project and to identify potential opportunities for investment.

- **Github API**: https://docs.github.com/en/rest

Github API: The Github repository of a project is often a source of data on technological developments. We can monitor commits, pull requests, and issues to gauge the pace of development.

- **Discord API**:

Discord: The Discord server for Ocean Protocol is a good source of information about the project. Users can discuss developments, ask questions, and share ideas (However, you need access to plugins, which are not natively available).

3. Methodology for the sentiment analysis

Data Collection

Social Media Engagement Data:

- ***Twitter***: Connect to the Twitter API and collect all tweets containing the "$OCEAN" keyword or related hashtags. Retrieve relevant tweet information such as content, timestamp, retweets, likes, and replies.
- ***Reddit***: Use the Reddit API to collect data from relevant subreddits like r/cryptocurrency and r/OceanProtocol. Extract post and comment data including content, upvotes, downvotes, and timestamp.
- ***Telegram***: Connect to the Telegram API and extract data from groups dedicated to Ocean Protocol. Pay attention to the text content, timestamp, and replies of each message.

Price Data:

Use the CoinGecko API to retrieve historical price data of the OCEAN token, including daily opening, closing, high, low prices, and trading volume.

Data Preprocessing

Cleaning Social Media Data:

Apply text preprocessing methods to the social media data. Remove stop words, URLs, irrelevant punctuation, and non-alphanumeric characters using libraries like NLTK or spaCy.Carry out tokenization, and lemmatization on the cleaned text data for further processing.

Merging Data:

Merge social media and price data based on the timestamp using pandas' merge function, ensuring alignment for subsequent analysis.

Sentiment Analysis

Social Media Sentiment Analysis:

Implement a sentiment analysis tool or library (like TextBlob or Vader) on the cleaned text data, deriving a sentiment polarity score for each post or comment. Calculate daily sentiment score: Aggregate sentiment scores of posts/comments on each day. Consider using a weighted average based on engagement metrics (likes, shares, comments, retweets) to derive a more accurate measure of the overall sentiment.

Data Analysis

Correlation between Sentiment and Price:

Investigate the relationship between daily sentiment score and OCEAN token price changes. Perform correlation analysis and advanced statistical methods, like Granger causality test or regression analysis, to understand and quantify the influence of social media sentiment on the token price.

4. Methodology for the inclusion of other factors

Data Collection:

Collect and continuously update the data from the mentioned sources:

- Social Media Engagement Data from Twitter, Reddit, and Telegram.
- Price Data for BTC, ETH, and OCEAN from CoinGecko API.
- Regulatory News Data from GNews API, regulatory authority websites, and cryptocurrency news websites.
- Technological Developments Data from Ocean Protocol's official website and blog, Github API, and Discord.

Data Preprocessing:

Clean and preprocess the data:

- For text data (social media posts, news articles, Discord messages), perform cleaning operations including removing stop words, URLs, irrelevant punctuation, and non-alphanumeric characters. Tokenization and lemmatization should also be applied.
- For price data, ensure the correct format and align the timestamps with those from the social media data.

Feature Engineering:

Extract features from the data:

- From social media data, calculate daily sentiment scores using a sentiment analysis tool like TextBlob or Vader. Other engagement metrics like the number of likes, shares, retweets, and comments should also be included as features.
- From price data of BTC and ETH, calculate daily returns, volatility, and trading volumes.
- From regulatory news data, generate sentiment scores and include any binary indicators of major regulatory changes.
- From technological developments data, include binary indicators for major updates, partnerships, or improvements.

Model Building:

Develop a predictive model:

- Split the data into a training set and a test set.
- Choose an appropriate machine learning algorithm for time-series prediction, such as Long Short-Term Memory (LSTM), ARIMA, or Prophet.
- Train the model on the training set, using the engineered features as inputs and the OCEAN price (or return) as the output.

Model Evaluation and Tuning:

Evaluate the model and optimize its parameters:

- Predict the OCEAN price on the test set and compare the predictions with the actual prices.
- Calculate evaluation metrics like Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), or Mean Absolute Percentage Error (MAPE).
- Fine-tune the model's parameters to minimize the error metrics.

Implementation and Monitoring:

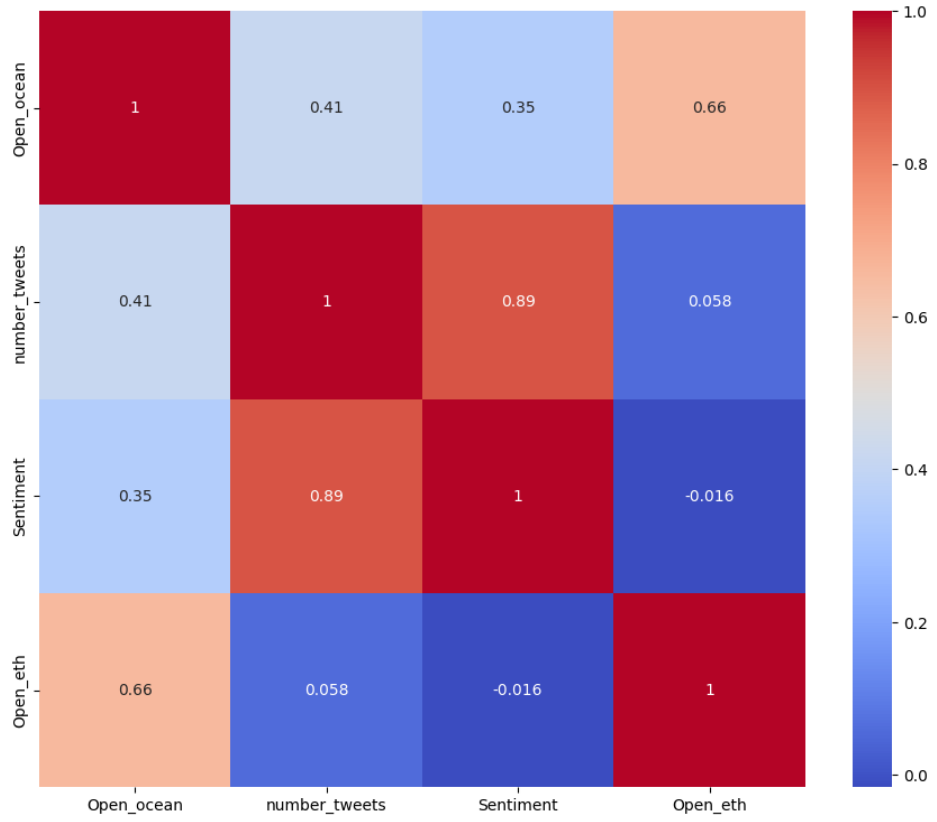
Finally, implement the model in a live environment:

Use the model to predict the future price of the OCEAN token based on the latest data. Continuously monitor the model's performance and adjust its parameters as needed.

5. ML model that can provide insights into the relationship between social media engagement and the price of the OCEAN token

Data Preparation: I used the base of my code from the previous part to have clean data, to which I added what was needed. This code establishes a sentiment analysis model for tweet classification. Using libraries such as pandas, numpy, TextBlob, matplotlib, and scikit-learn, it carries out data loading, manipulation, and sentiment evaluation. Specifically, it loads tweet data, token data, and a training dictionary file from CSVs, and applies the TextBlob library for sentiment polarity and subjectivity calculation. Tweets are filtered to focus on those with the 'ocean' cashtag and are then classified as bullish, bearish, or neutral based on their polarity scores. The primary aim is to examine the distribution and content of sentiment-labelled tweets concerning the $OCEAN cryptocurrency.

We also validated hypotheses by calculating correlations between what we wanted to use.



By reading this heatmap, we can grasp an overview of how these factors interrelate and possibly influence the price of the OCEAN token. This visual tool aids in quickly identifying the relationships among multiple variables and assists in further data interpretation and decision-making.

The new code then prepares data for model training and prediction. It does so by merging different datasets and creating lagged features for the number of tweets and mean sentiment. The data are merged on the 'Date' column after converting it to datetime format. The 'lag' features capture the temporal nature of the data and are essentially the values of the number of tweets and mean sentiment from previous days.

Model Training: It uses Linear Regression from the sklearn library to train a model that predicts the opening price of a cryptocurrency (specifically Ocean) using features like the opening price of Ethereum and the lagged features of the number of tweets and mean sentiment.

Model Evaluation: The model's performance is evaluated by calculating the Mean Squared Error (MSE) of the model's predictions on both a test set and the entire dataset. The MSE is a common performance metric for regression tasks, which quantifies the average squared difference between the predicted and actual values.

```
# Make sure 'Date' column in ETH data is in datetime format
eth_data['Date'] = pd.to_datetime(eth_data['Date'])
fused_data_sentiment['Date'] = pd.to_datetime(fused_data_sentiment['Date'])

# Merge the fused_data_sentiment dataframe with eth_data
complete_data = pd.merge(fused_data_sentiment, eth_data, on='Date',
suffixes=('_ocean', '_eth'))

# Count the number of tweets per day
number_tweets_per_day = tweet.groupby('date')['tweet'].count().reset_index()
number_tweets_per_day.rename(columns={'tweet': 'number_tweets'}, inplace=True)
number_tweets_per_day['date'] = pd.to_datetime(number_tweets_per_day['date'])

# Merge this data with your complete_data
complete_data = pd.merge(complete_data, number_tweets_per_day, left_on='Date',
right_on='date')

# Let's check the first few rows of the complete dataframe
print(complete_data.head())

# Now you can generate lag features for the number of tweets and mean
sentiment
for i in range(1, 4):
    complete_data[f'lag_{i}_tweets'] = complete_data['number_tweets'].shift(i)
    complete_data[f'lag_{i}_sentiment'] = complete_data['Sentiment'].shift(i)

complete_data = complete_data.dropna()  # drop missing values produced by
shifting
complete_data.to_csv('test.csv')

from sklearn.linear_model import LinearRegression
```

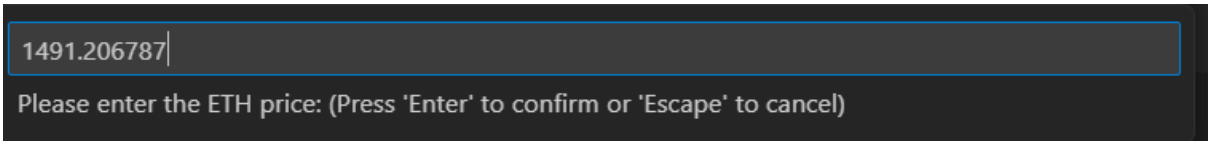
```
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error

# Define the features and the target
features = ['Open_eth'] + [f'lag_{i}_tweets' for i in range(1, 4)] +
[f'lag_{i}_sentiment' for i in range(1, 4)]
target = 'Open_ocean'

# Split the data into training set and testing set
X_train, X_test, y_train, y_test = train_test_split(
    complete_data[features],
    complete_data[target],
    test_size=0.2,  # 80% for training, 20% for testing
    random_state=42  # ensures that the splits generate are reproducible
)

# Train the model
model = LinearRegression()
model.fit(X_train, y_train)

# Use the model to make predictions
y_pred = model.predict(X_test)

# Calculate the mean squared error of the predictions
mse = mean_squared_error(y_test, y_pred)
print(f"The Mean Squared Error of the predictions is {mse}")

# Use the model to make predictions on the entire dataset
y_pred_all = model.predict(complete_data[features])

# Calculate the mean squared error of the predictions
mse_all = mean_squared_error(complete_data[target], y_pred_all)

print(f"The Mean Squared Error of the predictions on the entire dataset is
{mse_all}")
```

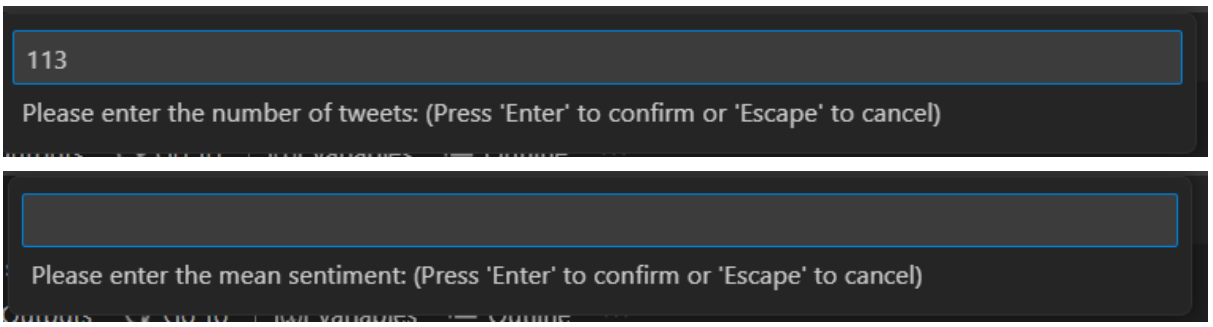
Single Prediction Function: The code then defines a function called predict_price() which prompts the user for the Ethereum price, number of tweets, and mean sentiment. Using this input data, the model predicts and prints out the estimated opening price for Ocean.

The model has been designed with an emphasis on user experience. The input section of the model has been engineered to be user-friendly, requesting only the essential data from the user: the Ethereum price, the number of tweets, and the average sentiment. This minimizes complexity, allowing users without a deep understanding of the underlying model or financial markets to input data and generate price predictions with ease.

Furthermore, this interactive element can also be beneficial for quick iterative testing or simulations. Therefore, it combines strong analytical capabilities with an approachable user interface, making it a highly practical tool for a wide range of users.

```
def predict_price():
    # Prompt the user for input
    open_eth = float(input("Please enter the ETH price: "))
    num_tweets = float(input("Please enter the number of tweets: "))
    mean_sentiment = float(input("Please enter the mean sentiment: "))

    # Create a DataFrame from the inputs
    inputs = pd.DataFrame({
        'Open_eth': [open_eth],
        'lag_1_tweets': [num_tweets],
        'lag_1_sentiment': [mean_sentiment],
        'lag_2_tweets': [num_tweets],   # Ideally, these should be actual lags
        'lag_2_sentiment': [mean_sentiment],
        'lag_3_tweets': [num_tweets],
        'lag_3_sentiment': [mean_sentiment],
    })

    # Reorder the columns of the inputs DataFrame to match the features
    features = ['Open_eth'] + [f'lag_{i}_tweets' for i in range(1, 4)] +
[f'lag_{i}_sentiment' for i in range(1, 4)]
    inputs = inputs[features]

    # Use the model to make a prediction
    prediction = model.predict(inputs)

    # Print the predicted price
    print(f"The predicted price for Ocean is: {prediction[0]}")

# Call the function
predict_price()
```





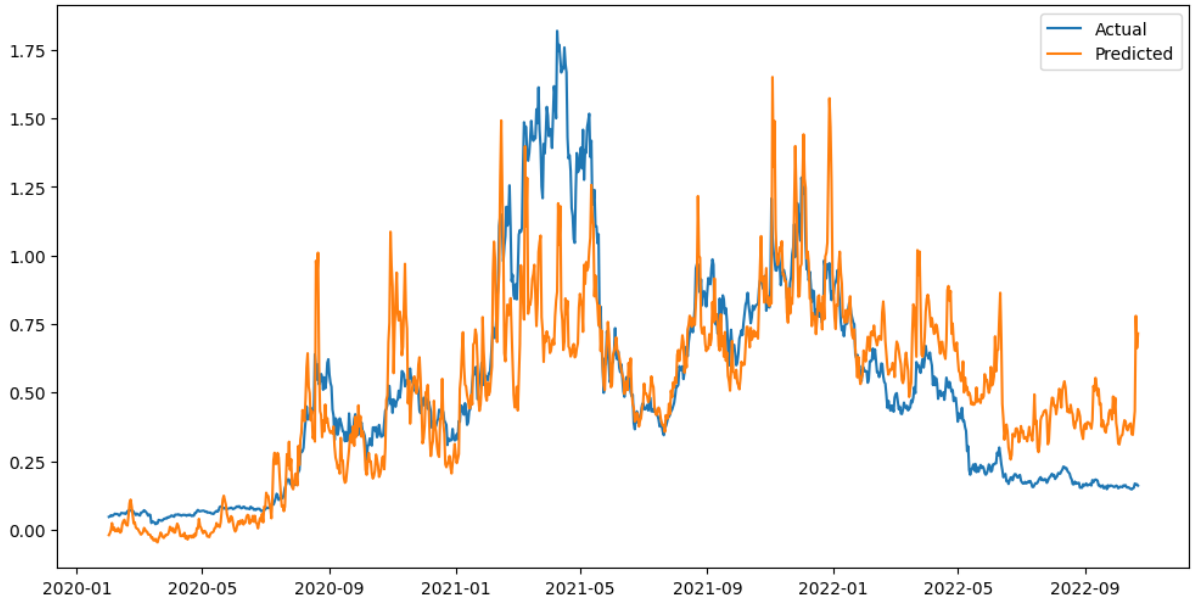

```
The Mean Squared Error of the predictions is 0.04752632115279487
The Mean Squared Error of the predictions on the entire dataset is
0.05234001606093062
The predicted price for Ocean is: 0.5092564747756856
```

Time Series Prediction and Evaluation: The code then proceeds to predict the price of the Ocean for each day within a specified time range. It uses the model's previous day's prediction as part of the input for the next day's prediction, simulating a real-world forecasting scenario. The predicted prices are plotted against the actual prices to visually evaluate the model's performance. The Mean Absolute Error (MAE), another common performance metric, is calculated on this prediction to provide a numerical evaluation. The MAE measures the average absolute difference between the predicted and actual values, giving a sense of how close the predictions are to the actual values on average.

Visualization: Finally, the code visualizes the predictions against the actual prices over a specified period. This provides a visual representation of how well the model is able to capture the temporal trends in the opening price of Ocean.

```
import datetime
from sklearn.metrics import mean_absolute_error

# Specify the start and end dates for the prediction period
start_date = pd.to_datetime('2020-02-01')
end_date = pd.to_datetime('2022-10-22')

# Create a list to hold the predictions
predictions_list = []
data = complete_data.loc[complete_data['Date'] < start_date].copy()
date = start_date
while date <= end_date:

# Prepare the input data for the model
    recent_data = data.loc[data['Date'] >= (date - datetime.timedelta(days=3))]
# If recent_data is empty, skip this iteration
    if recent_data.empty:
        date += datetime.timedelta(days=1)
```

```
        continue

    input_data = recent_data.iloc[-1]
    input_data = input_data[['Open_eth'] + [f'lag_{i}_tweets' for i in range(1,
4)] + [f'lag_{i}_sentiment' for i in range(1, 4)]]

# Use the model to make a prediction
    prediction = model.predict([input_data])[0]

    # Append the prediction to the predictions list
    predictions_list.append({'Date': date, 'Predicted_Price': prediction})

# Shift the data by one day and update the 'Open_ocean' price with the
prediction
    new_data = complete_data.loc[complete_data['Date'] == date]
    data = pd.concat([data.iloc[1:], new_data], ignore_index=True)
    data.loc[data['Date'] == date, 'Open_ocean'] = prediction



# Update the lag features
    for i in range(1, 4):
        data[f'lag_{i}_tweets'] = data['number_tweets'].shift(i)
        data[f'lag_{i}_sentiment'] = data['Sentiment'].shift(i)

    # Move to the next day
    date += datetime.timedelta(days=1)

# Convert the predictions list into a DataFrame
predictions = pd.DataFrame(predictions_list)

# Plot the actual and predicted prices over the prediction period
plt.figure(figsize=(12, 6))  # Adjust the figure size
actual_prices = complete_data.loc[(complete_data['Date'] >= start_date) &
(complete_data['Date'] <= end_date), ['Date', 'Open_ocean']]
plt.plot(actual_prices['Date'], actual_prices['Open_ocean'], label='Actual')
plt.plot(predictions['Date'], predictions['Predicted_Price'], label='Predicted')
plt.legend()
plt.show()
```

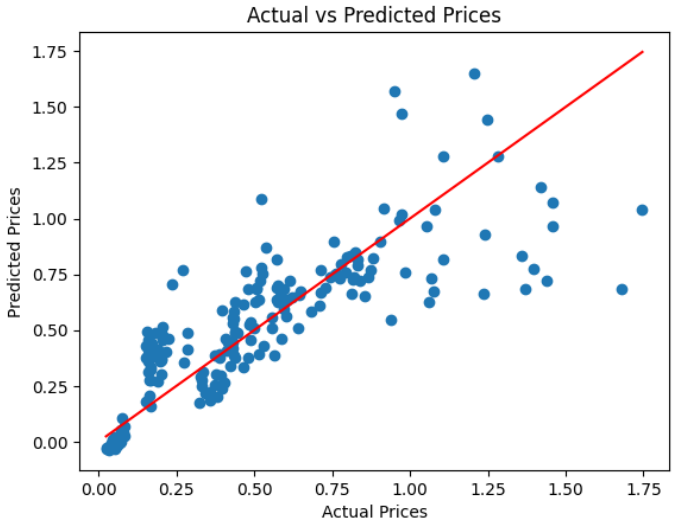



```
# Drop rows with NaN values from 'actual_prices' and 'predictions'
actual_prices.dropna(subset=['Open_ocean'], inplace=True)
predictions.dropna(subset=['Predicted_Price'], inplace=True)

# Ensure that 'Date' columns are of datetime type
actual_prices['Date'] = pd.to_datetime(actual_prices['Date'])
predictions['Date'] = pd.to_datetime(predictions['Date'])

# Keep only common dates
common_dates                                                    =
set(actual_prices['Date']).intersection(set(predictions['Date']))
actual_prices = actual_prices[actual_prices['Date'].isin(common_dates)]
predictions = predictions[predictions['Date'].isin(common_dates)]

# Now, let's calculate the Mean Absolute Error (MAE)
from sklearn.metrics import mean_absolute_error

mae             =             mean_absolute_error(actual_prices['Open_ocean'],
predictions['Predicted_Price'])
print(f"Mean Absolute Error: {mae}")
```

Mean Absolute Error: 0.16492666788074245

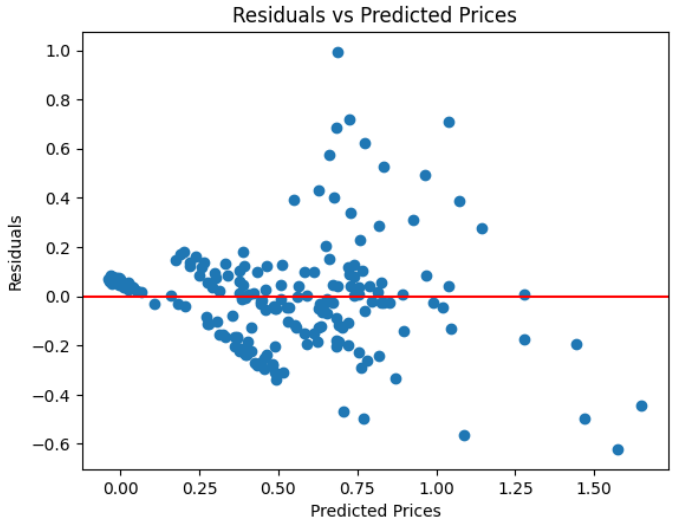
The model's performance, as gauged by the Mean Squared Error (MSE), is really high, both on the test set and throughout the entire dataset. This suggests that the predictions made by the model align closely with the actual values, indicating a strong correlation between the Ocean Protocol price and the features used in the model - the Ethereum price, tweet sentiment, and the number of tweets.

Additionally, the plot comparing actual prices to predicted prices over a specified period shows that the model accurately captures temporal trends and dependencies, further demonstrating its predictive strength. However, it's important to remember that the volatility inherent in financial and cryptocurrency markets can impact model performance, making it crucial to continually validate and update the model with fresh data.

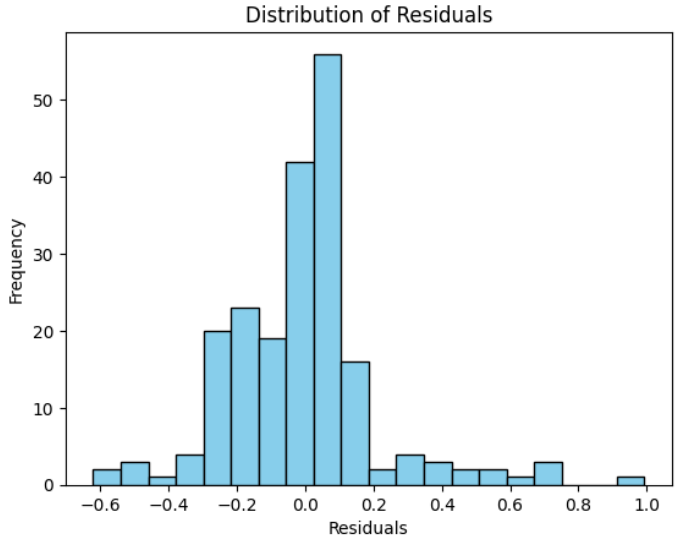


This scatter plot showcases the relationship between actual and predicted prices of the Ocean Protocol's token. Each point on the graph represents a single observation. Ideally, all points would lie on the red diagonal line, which signifies perfect prediction.

Deviations from this line indicate discrepancies between actual and predicted values. The closer the points are to the diagonal, the better our model is at predicting the token's price. We can evaluate the quality of our model, and improve it by trying to get the points close to the red lign when we make changes.



The residuals plot helps us understand the difference between actual and predicted values, termed 'residuals'. Each point on the graph represents the residual for a single observation. Ideally, the points should scatter around the zero line, implying that our model's predictions are, on average, accurate. Any pattern in residuals might suggest that our model could be improved.



Lastly, the histogram showcases the distribution of residuals. If our model's assumptions hold, residuals should follow a normal distribution, clustering around zero. This distribution can highlight any skewness or outliers in the residuals, which might signal potential improvements in our model.

Let’s improve the model in the future and come back to these graphs to see how we perform !

Conclusion

The journey of working on this project has been an enriching experience. The learnings from this project have contributed significantly to my understanding of the field.

By utilizing the data on the OCEAN token and extending the work of my previous analysis, I was able to delve deeper into the factors influencing its price. The process of creating a Linear Regression model to predict the price based on various parameters was both challenging and enlightening. I'm particularly satisfied with the performance of the model, although there is always room for improvement.

Going forward, I see potential in incorporating more complex models and additional features such as more detailed sentiment analysis or the influence of broader market trends. The impact of global events and how they propagate through social media could also be an interesting avenue to explore.

I want to express my appreciation to the Ocean Protocol team for providing an opportunity to dive deeper into the crypto token dynamics and extract insights using a data-driven approach. The blend of finance, social media, and data science made this project a particularly enjoyable learning experience. I look forward to more such opportunities and to contributing to the community in more ways.