X-ray on Curve DEX Dynamics

1. Introduction 🏁

Main Dashboard used to support this report: https://flipsidecrypto.xyz/Rodolfo-Lima/curve-trading-volume-dynamics-SL0ga8

Github Repo: https://github.com/LimaRods/Curve-Trading-Volume-Dynamics/blob/main/analytics.ipynb

1.1 Protocol Introduction

Curve Finance is a prominent popular automated market maker (AMM) platform focused on providing low fees and low slippage swaps by offering liquidity pools with similar behavior assets. In terms of Total Value Locked, Curve is the second-largest decentralized exchange (DEX) in DeFi, presented in 13 Blockchain Networks. Curve also incentivizes stakeholders by integrating other DeFi protocols and offering different ways to offer extra rewards like staking, locking its governance token CRV, and boosting users' earnings besides only providing liquidity.

1.2 Objective

The objective of this research is provide to readers with a comprehensive analysis of the Protocol when it comes to trading behaviour and liquidity dynamics, covering different faces of the protocol like vote-escrowed tokens (veCRV) and why Curve is popular and relevant in the DeFi field.

We will explore inherent risks and external impacts on the protocol's health, as well as, how to use its genius features and design to improve traditional finances.

At last, we would pick a pool Curve to build a predictive model about its volume.

2. Methodology 🧪

Keep in mind that most of the data collected was restricted to Curve DEX deployed on Ethereum, we won't mention that anymore. We only brought data from other chains to elucidate the distribution of TVL across the chains.

2.1 Data Sources and Tools

For this project, besides the two datasets provided by Ocean Protocol, we used FlipsideCrypto data and DeFiLlama. Interesting fact: Flipside platform allows us to use DeFilama datasets and API through Flipside Studio, in this way, we can use SQL language to manipulate its data and the no-code visualization tool in order to consolidate different sources of data into a single dashboard.

We used Python to read, prepare, and visualize the datasets provided by Ocean, as well as to build the ML algorithm. You can find this in our GitHub repository

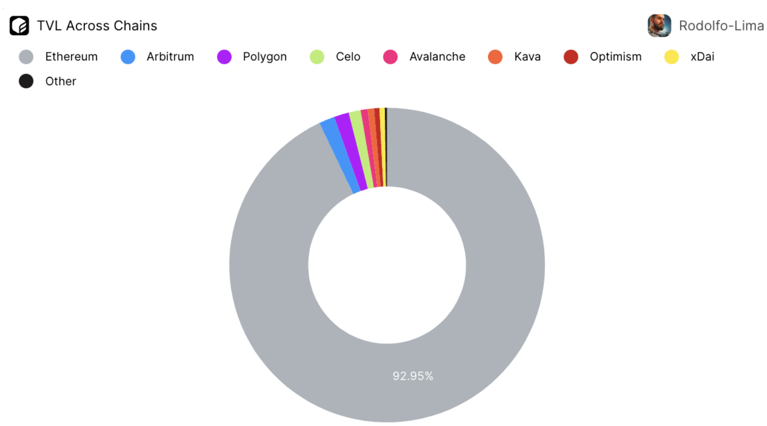
3. Discussion & Analyses 📈

3.1 Exploratory Data Analysis

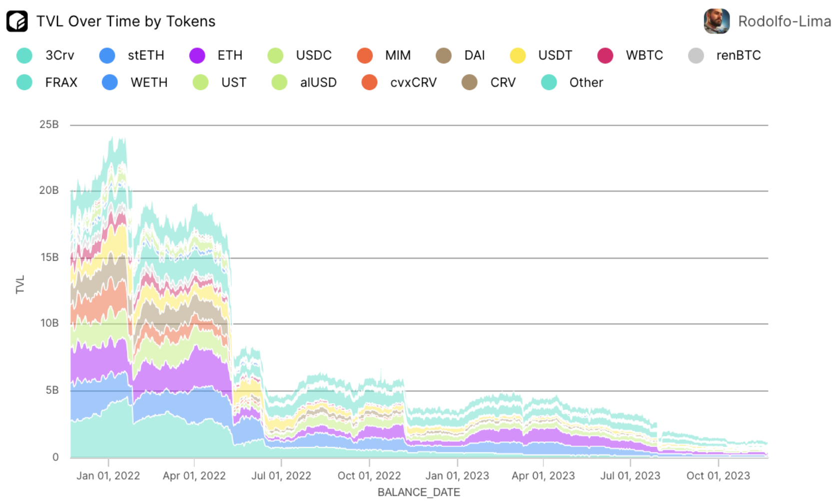
As mentioned in the article's introduction, it's highly recommended you take a look at the Curve Dashboard to zoom out our complete EDA, It would be boring to cover all charts built into the dashboard. In this section, we will present the Curve's facets in the last 2 years responsible for making it a DeFI's leading AMM.

TVL (Total Value Locked):

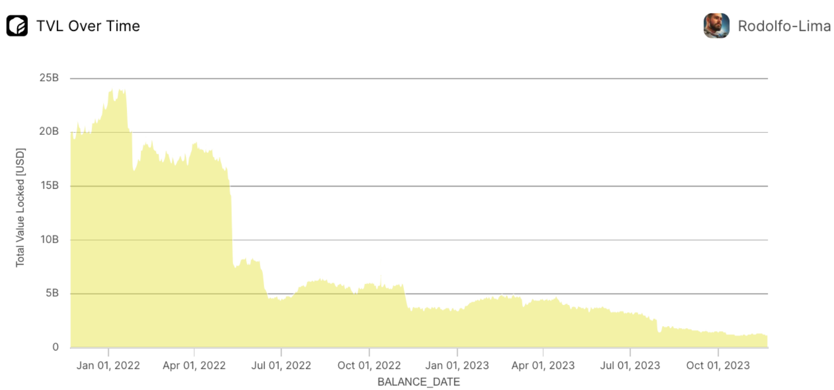
The current TVL of Curve DEX is **$1.752b** distributed across multiple Chains. Only Ethereum accounts for **93% ($1.57b)** of the value, following Arbitrum (1.61%) and Polygon (1.53%):



Observing the TVL of pools over time and breaking down the metric by tokens, we realize that the main tokens utilized to provide liquidity to the protocol over time are: **3CRV** (LP Token of USDT-USDC-DAI), **stETH, ETH, USDC, MIM, USDT, DAI, FRAX** and **WBTC:**



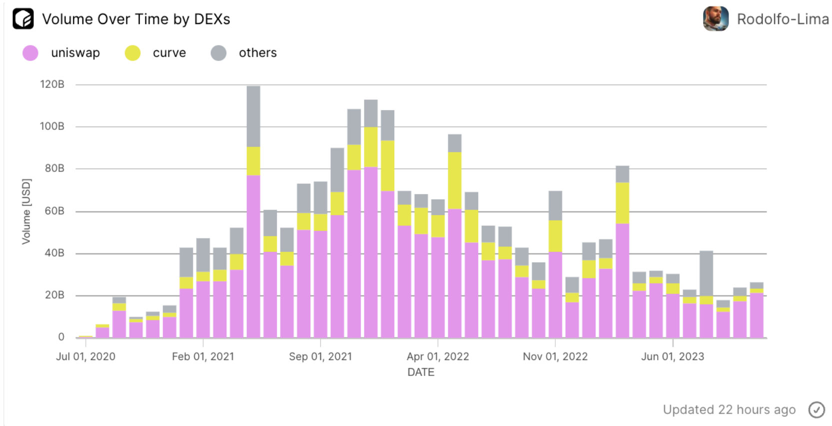
It's also interesting to look at some sharp declines in TVL at times. Most of them are directly linked to external events and we will talk more about them in the next sections. For now, notice them in the chart below:

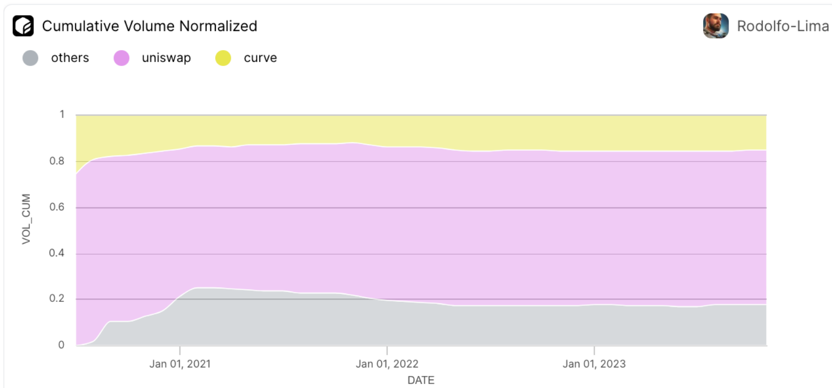


Volume, Revenues, and Slippage:

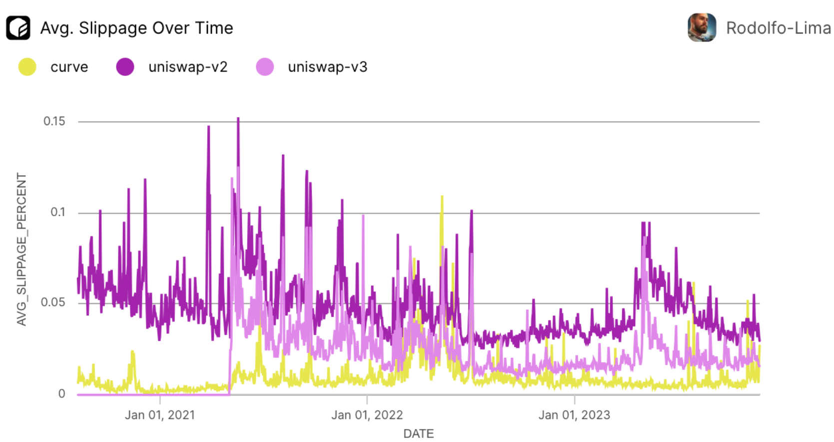
Since the protocol creation, besides market volatility and eventual impacts, Curve accumulated **$331.75b** of trading volume, collected more than **$230M** in trading fees, and generated a total revenue of **$113.63M** distributed according to its tokeneconomics design. Let's break down the trading volume from top to bottom.

Even though Uniswap is the DEX leader, Curve continues to keep its 2nd place and presents a relevant player in the realm, covering **18%** of all volume generated by DEXs on Ethereum since your creation, as shown below:



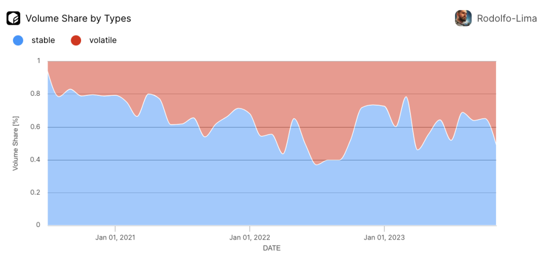


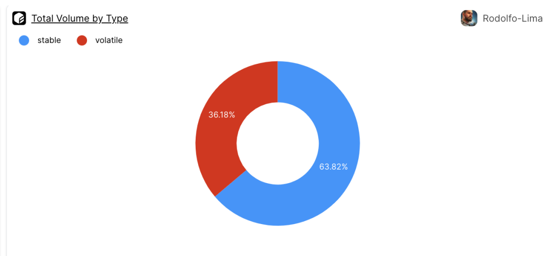
However, Curve has been showing itself the best protocol when the matter is **low slippage** and **fees**. The charts below elucidate this:



Pay attention to how the average slippage % (fees were not discounted from the slippage calculation) of Curve remains in general lower than the Uniswap pools. Even the v3 pools, in which their concentrated liquidity mechanism reduces the slippage, are shown to be higher than in Curve pools. There are some exceptions in the time series like slippage peaks.

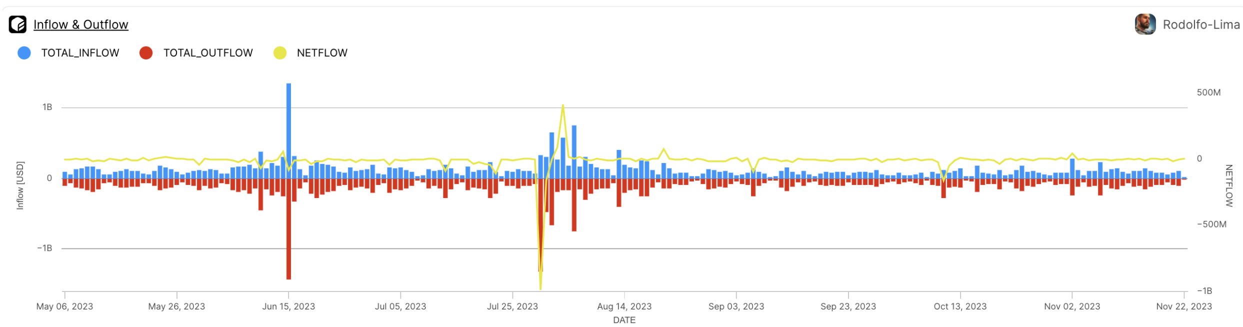
Another remarkable characteristic of the protocol is the **stability** over volatility and speculation. As we can see, **64%** of all volume in USD is derived from stablecoins :





USD Inflow and outflow:

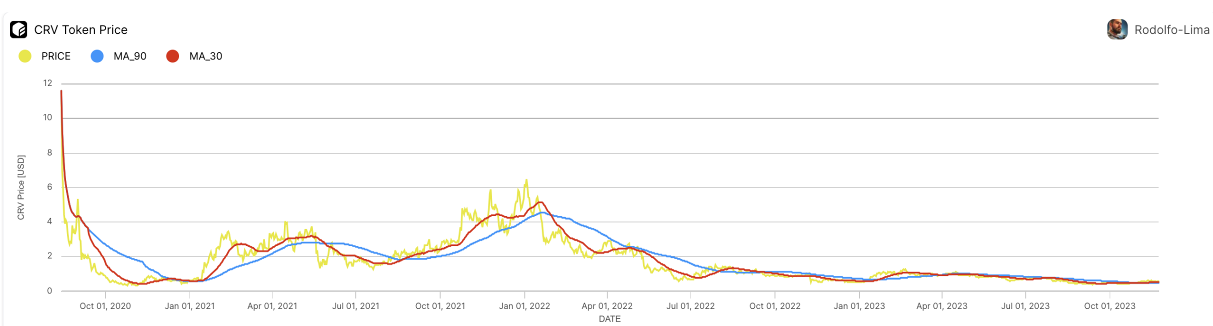
USD inflows and outflows are the directions that the money is following, whether it's getting into the protocol or getting out of the protocol. Those money movements are caused by **swaps**, **liquidity additions**, and **reductions** triggered by AMM's users. Additionally, it's interesting to measure the net value of those two metrics over time, as illustrated in the following chart:



Note the high negative netflow on **July 30th**. We will cover this phenomenon caused by external events in another section.

$CRV Token:

CRV token is the token governance of the protocol used to incentivize liquidity and involve its holders in the governance of the protocol. Its price lost much value since your launch. However, similar price movements happened to other DeFi tokens given the market conditions and the crypto winter. Follow the time series containing its price and a 30 and 90-day moving average:

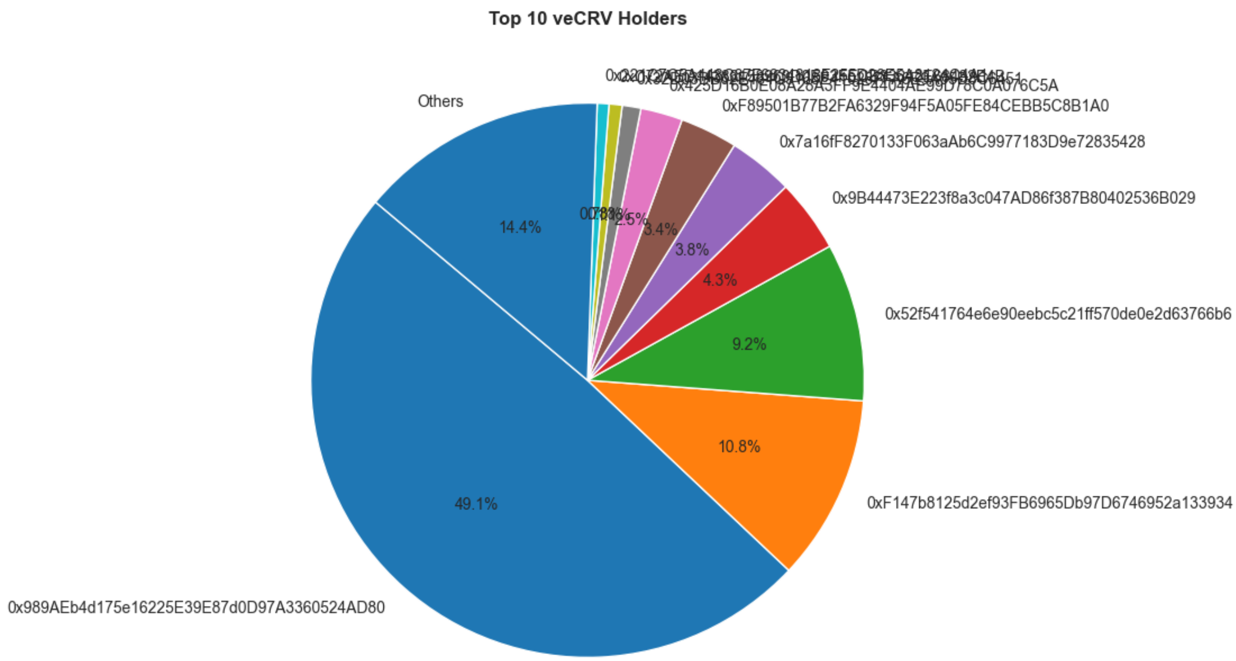


3.2 veCRV Impact

Regarding veCRV (vote-escrowed CRV token), it is a token that represents voting power in the Curve DAO governance system. Holders of veCRV tokens can vote on proposals related to the management and development of the Curve protocol, such as changes to fees, new pools, and upgrades to the platform. To hold veCRV, we should first lock CRVs.

As interpreted from the dashboard, we can't see a clear volume shift right after the launch of veCRV nor in the TVL or CRV token transfer. Of course, in the following weeks and months, we will notice growth in both metrics associated with other factors.

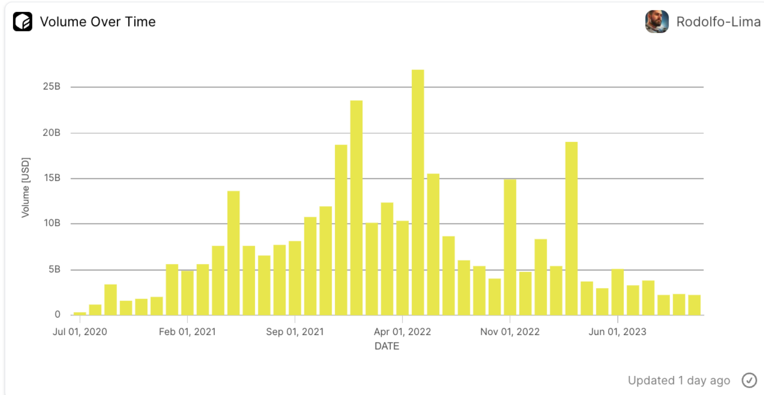
Still, we know that the impact of using veCRV to incentivize liquidity is in the long term. Given the dataset regarding CRV transfer provided by Ocean Protocol, the method `Add liquiquidity` is more representative than `Remove liquidity.` Other interesting fact: there are currently (Nov 1st, 2023) **9022 wallets holding** the token, even though the distribution is too far from being even, based on the next chart:

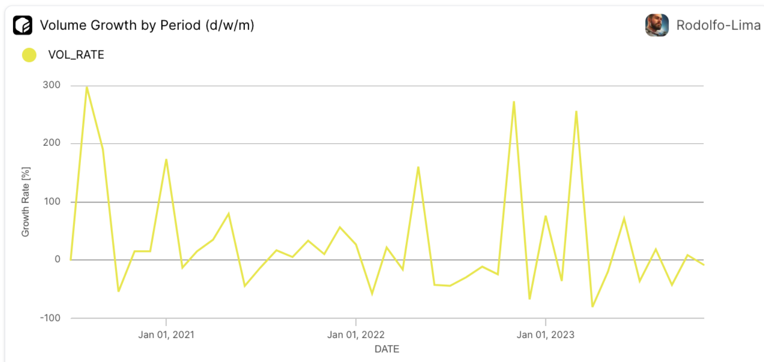


This concentration of veCRV and its scarcity has been causing noise in the DeFI community. I will talk more about that in the topic **Curve Wars and Bribes,** section 3.7 Unusual Activities.

3.3 Historical Revolution

Since the official launch of the protocol in Aug 2020, the volume trend has followed the same patterns as other DeFi protocols, increasing the volume until around the half of 2022 and following a downtrend. These events were recognized by the market as the crypto bull market and crypto winter. Watching the monthly volume growth rate, we can realize some shifts in trading volume, as seen below:

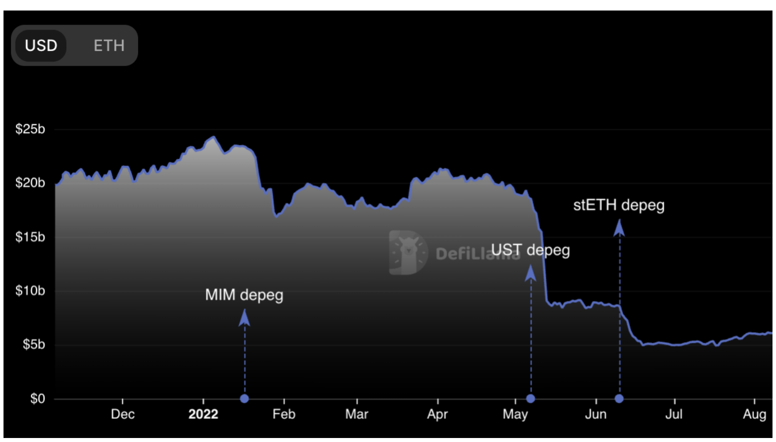




The 3 abrupt changes in volume (monthly) since Aug 2020 are:

- **Nov 2022**. The trading volume grew by **273%**
- **Mar 2023**. The trading volume grew by **256%**
- **Jan 2021**. The trading volume grew by **174%**

It's not clear which real-world events precipitated those positive shifts in volume besides the market volatility. Still, if we pick up other metrics like TVL and USD inflow we can get interesting insights such as:



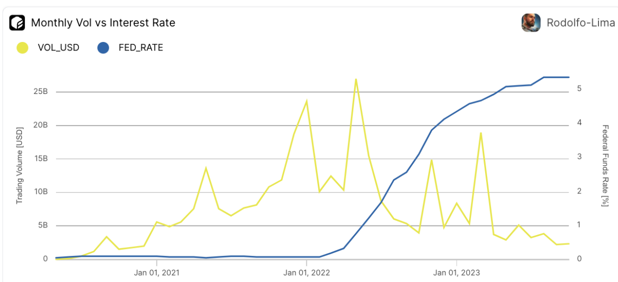
The 3 most notorious changes in the liquidity (TVL) of the protocol are directly correlated by the depeggings in crypto like:

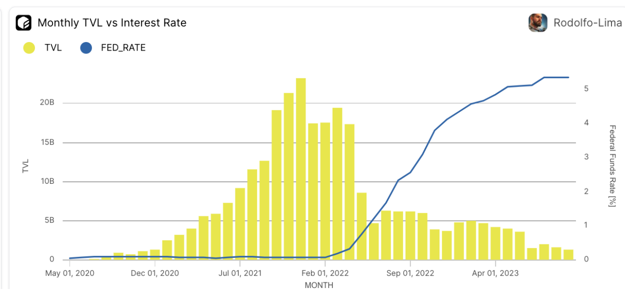
- **UST depeg** on **May 7, 2022** . One of the most critical events in crypto. The meteoric crash of UST and LUNA generated a pessimism wave in the entire field
- **MIM depeg** on **Jan 17, 2022**
- **stETH depeg** on **Jun 10, 2022**

As mentioned in the EDA section. If we use the USD inflow chart, we'll see a NetFlow of **-$989M** in Curve pools on Ethereum. Almost 1b of USD got out of the protocol on Jul 30, 2023. It's the date of the Reentrancy hack that Curve Finance suffered.

3.4 Trading Volume Fluctuations

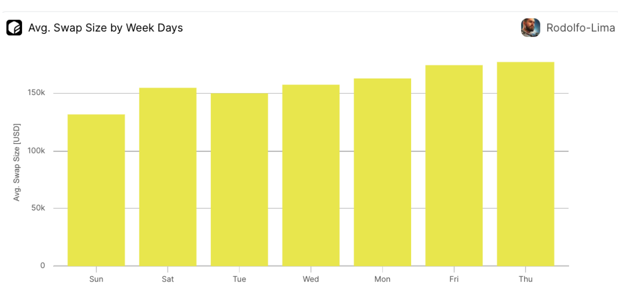
So far, we've already seen many times the trading volume over time chart from different angles. For now, let's zoom out and see how market and economic conditions can affect the volume fluctuation. We used the federal funds rate to analyze the interest rate and the trading volume together as shown in the chart below:

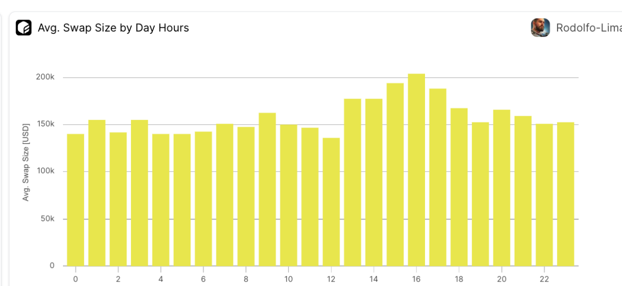




It's impressive to see the negative association between TVL/Volume and the interest rate. This could be because higher interest rates might mean less liquidity in the market or other financial instruments becoming more attractive than crypto trading.

Now, let's zoom in as much as we can and examine the trading volume along the days of weeks and within the days, as displayed in the charts below:

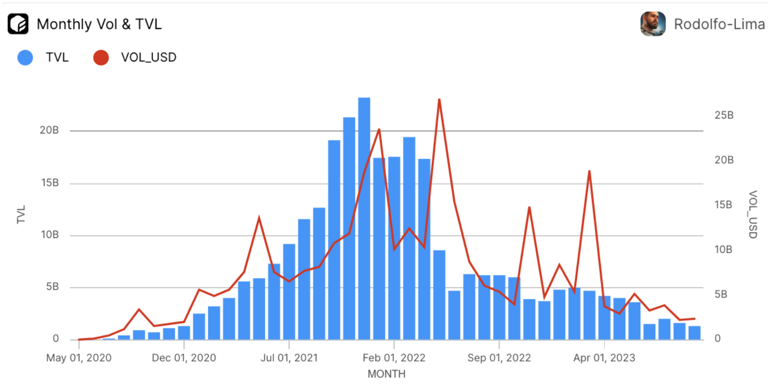




According to the charts, the highest swaps in terms of volume, on average happen on Thursday/Friday around 3 and 4 PM. The distribution of total trading volume instead of swap size follows the same pattern.

3.5 Trading Volume vs Liquidity

Now, let's examine the connection between trading volume and liquidity in Curve Finance. At first glance, we can see a positive association pattern between liquidity represented as TVL and trading volume, as understood from the next chart:



I've already written an article explaining how trading, liquidity, and slippage are all connected, you can access it at this link, in the section "4.1 Liquidity Overview on DEXs". First, the main factors that impact trading volume are:

- **Interest in certain tokens**: High trading volume often indicates a high level of interest or activity in a particular token. This can be a useful signal for traders and liquidity providers when deciding which tokens to trade or provide liquidity for, given the trades generate fees for liquidity providers.
- **Market Depth**: A deeper market can suggest high trading volume. In a deep market, large orders can be executed without causing a significant impact on the price. This is especially crucial in the context of DEXs and liquidity pools, where the size of trades relative to the pool can impact the token prices and create slippage.
- **Slippage and fees**: High slippages and fees discourage trades, particularly for larger trades. This is because larger trades are more easily absorbed in markets with high volumes, reducing the impact on the price.

Now, let's dive into more details about how **market depth (or liquidity) influences the trading volume**, and as a bonus, let's add the slippage effects to our explanation. Curve has its proprietary formulas to provide high volume but its foundation is based on follows the Uniswap V2's AMM algorithm. You can think of Curve as a specialized version of the Uniswap exchange so you will not cover the details of Curve's AMM algorithm, the following explanation is the fundamental concept of AMMs in DeFI.

The price in Liquidity Pools works based on a trading algorithm:

$$k = x \cdot y$$

Which:

- x is the balance of asset A, and y is the balance of asset B in a token-pair pool
- k is called the invariant and it's required to stay fixed over time

It's better explained with an example:

Given a Uniswap **DAI-USDC** AMM with **4 DAI and 4 USDC**. It sets the instantaneous exchange rate at **1 DAI: 1 USDC**

An Investor deposits 4 DAI to exchange for USDC, then:

$x \cdot y = 4 \times 4 = 16$

After the deposit, we'll have $x = 8 \rightarrow y = \frac{k}{8} = 2$

Thus, 2 USDC are withdrawn from the pool, and the new effective exchange rate in the AMM is set as **2 DAI: 1 USDC** because the investor deposited 4 DAI and withdrew 2 USDC. The slippage would result in a 50% decrease in the price of DAI

> *Note*: In the real world, arbitrageurs would seize this as an opportunity to deposit USDC and acquire DAI at a cheaper rate. This process would bring the AMM price back into equilibrium, thus aligning it with the market prices.

Let's implement a simple algorithm to visualize this in practice and simulate this behavior (again, you can find the resources in my article about liquidity in DEXs):

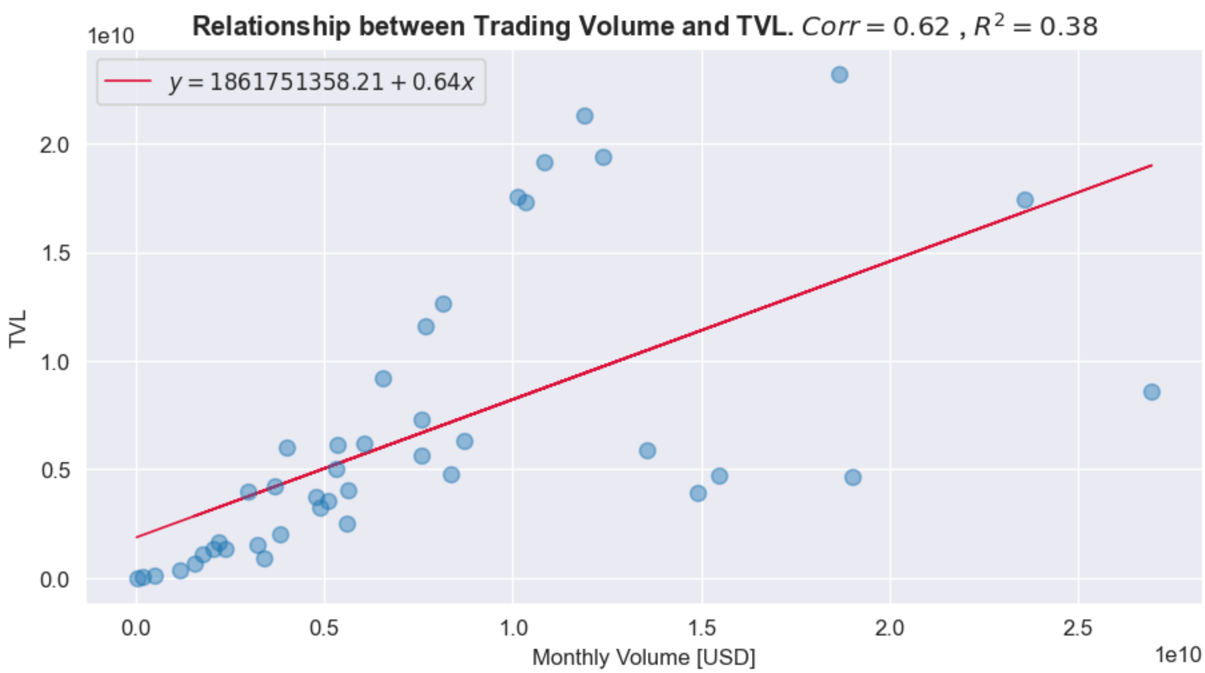
deposit_number	DAI	USDC	K	Price of USDC in DAI	Price of DAI in USDC	USDC Slippage percent
0	4	4	16	1	1	0
1	5	3.2	16	1.25	0.8	25
2	6	2.67	16	1.5	0.67	50
3	7	2.29	16	1.75	0.57	75
4	8	2	16	2	0.5	100

As we notice, the higher the deposit, the higher the impact. Let's try out a balance of **400 USDC** and **400 DAI**:

deposit_number	DAI	USDC	K	Price of USDC in DAI	Price of DAI in USDC	USDC Slippage percent
0	400	400	160000	1	1	0
1	401	399.0025	160000	1.0025	0.9975	0.2499
2	402	398.01	160000	1.005	0.995	0.4999
3	403	397.0223	160000	1.0075	0.9926	0.7500
4	404	396.0396	160000	1.01	0.9901	1.0000

Conclusion: The higher the pool liquidity in relation to the trade size, the lower the slippage. **For a deep pool as $x \to \infty$, slippage approaches 0** and consequently **attracts more trading volume.**

To put the theory into practice, look at the linear regression and correlation between Volume and TVL (Monthly) on Curve (Ethereum) since May 2020:



It's interesting to notice that the data looks more linear until reaching a certain trading volume level, after that, the correlation between TVL and volume is not very evident.

3.6 Trading Strategy

The more convenient way to make money on Curve is through **Yield Farming.** In a nutshell, provide liquidity to curve pools, stake your LP tokens, lock CRV tokens, and boost your rewards on provided liquidity.

However, once the user understands the basic mechanics of AMMs to generate liquidity, as presented in the last section, let's present a trading strategy of arbitrage explored for MEV (maximum value extracted) bots.

The main idea of this strategy is to start with asset A and sell it for asset B on the exchange where asset B is cheaper. Then, you would take asset B and sell it for asset A on the other exchange, receiving more of asset A in return than when you started.

Fictional Example:

Let's suppose that you want to explore the low slippage of Curve and capitalize on price differences between exchanges (**Curve** and **Sushiswap**)

- Initial situation:
  - The exchange rate between ETH and USDC on Curve is **1 ETH/2,060 USDC**.
  - Sushiwap DEX has a spot ask price of **1 ETH for 2,050 USDC** (i.e., the ETH is more expensive on Curve than Sushiswap).
- Flash Loan Method
  - Flash loan taken for **1M USDC** from Aave
  - Exchange 1M USDC for **487.80 ETH** on Sushiswap ((The instantaneous exchange rate would be different, due to slippage.))
  - Use **487.80 ETH** to trade for **1,004,878 USDC** on Curve
  - Repay the flash loan with **1M USDC** making a profit of **4,878 USDC**

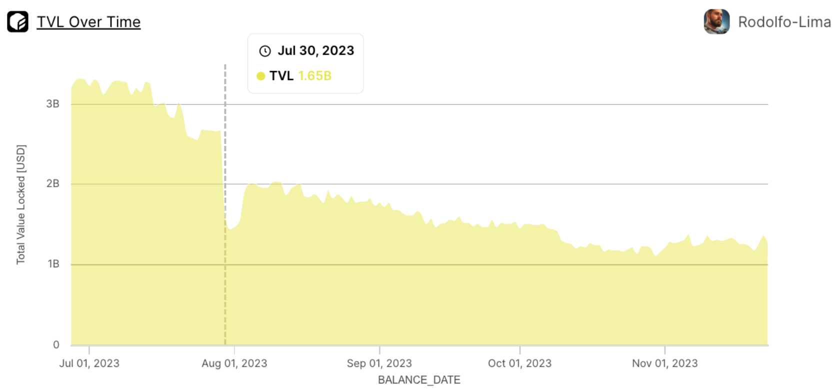
📖 A **Flash Loan** is a riskiness and no-duration loan where you don't need collateral. All this process happens almost instantaneously is can be automated by smart contracts and bots.

3.7 Unusual Activities

Given the charts discussed so far, we are able to discuss briefly two unusual activities:

- **Reentrancy Hack:**

  In the **USD Inflow chart** in the **3.1 section** or the dashboard, it's notorious the valley of **-$989M** in Netflow on J**uly 30.** The primary cause of the exploit was a malfunction in the reentrancy locks of specific versions of Vyper, a contract-oriented programming language that targets the Ethereum Virtual Machine (EVM). The protocol was exploited leading to losses surpassing **$47M.** We can see this impact from a different angle, observing the reduction in TVL on the protocol:



This type of event negatively affects the confidence of investors and users in Curve, it might hinder the inflow of capital into the protocol and their effects can be irreversible.

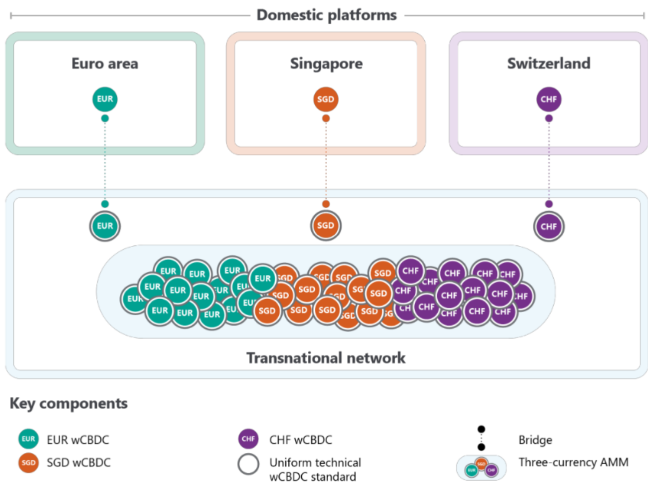
- **Curve Wars and Bribes**

  As illustrated in section **3.2.** One single holder accounts for nearly **50%** of all veCRV tokens held, and only 10 holders represent **86%** of all veCRV in wallet balance. As we know, veCRV (vote-locked Curve) gives voting power to users to participate in DAO governance when they lock their CRV.

  “Curve Wars" involve intense competition for limited Curve tokens (CRV), leading projects to engage in token bribes to influence CRV distribution across liquidity pools. Platforms like Convex facilitate this, controlling a significant share of CRV. This aggressive strategy, driven by decentralized governance and scarcity of resources, risks market manipulation and attracts regulatory scrutiny, potentially impacting the protocol's integrity and smaller participants.

3.8 BIS Use Cases

Project Mariana is a cross-border CBDC (Central Bank Digital Currency) pilot program that aims to explore the use of automated market makers (AMMs) for the exchange of wholesale CBDCs between different central banks. The main purpose of the project is to investigate how this technology can be used to improve the efficiency, speed, and cost-effectiveness of cross-border payments and settlements. The project is being led by the Bank for International Settlements (BIS), which is working with several central banks and other stakeholders to develop and test the technology. The ultimate goal of Project Mariana is to create a more seamless and integrated global financial system that can support the needs of businesses and consumers around the world. The main components and the high-level infrastructure are illustrated below:



Source: https://www.bis.org/publ/othp75.htm

Curve's AMM design is optimized for trading assets that are pegged to the same value, such as different stablecoins. This design is particularly well-suited for cross-border CBDC projects like Project Mariana, which involve the exchange of wholesale CBDCs between different central banks.

Using Curve's AMM technology can enhance the efficiency and effectiveness of cross-border CBDC transactions in several ways. First, the use of an AMM can help to ensure that the exchange rate between different CBDCs remains stable, which can reduce the risk of currency fluctuations and make cross-border transactions more predictable. Second, the use of an AMM can help to maintain liquidity in the underlying pools, which can ensure that transactions can be executed quickly and at a low cost. Finally, the use of a decentralized and transparent mechanism for managing liquidity and governance, the DAO used by Curve Finance, can help to ensure the long-term sustainability and success of the platform

By working together, these two sectors can leverage their respective strengths to create new and innovative solutions that can benefit the broader financial ecosystem. For example, the use of AMMs and other DeFi technologies can help to improve the efficiency, speed, and cost-effectiveness of cross-border transactions, while the involvement of central banks can help to ensure that these solutions are safe, secure, and compliant with relevant regulations. Additionally, the collaboration between BIS and Curve can help to promote greater understanding and awareness of DeFi among traditional financial institutions, which could help accelerate the adoption of these technologies and drive further innovation in the financial sector.

However, challenges associated with pre-funding, liquidity issues, and privacy concerns need to be addressed. The report of the project Marian suggests that collaboration between relevant stakeholders in existing FX (Forex) markets would be required to explore the commercial viability of AMMs for spot FX transactions using wCBDCs. Additionally, mechanisms to alleviate the costs of pre-funding, such as introducing short-term credit, could be investigated. Privacy-preserving mechanisms, such as stealth addresses, could also be put in place to support the privacy needs of users. Overall, the report suggests that while there are challenges associated with integrating Curve's AMM design into the CBDC ecosystem, these can

3.9 Prediction Model

In this comprehensive report, we show the concepts and real data that evident how trading volume, TVL, and market conditions are associated. Now, let's use this variable to forecast the future trading volume of 3pool (`0xbebc44782c7db0a1a60cb6fe97d0b483032ff1c7`).

We are considering in our prediction the monthly trading volume of the pool, its TVL, and the FED interest rate representing the market condition. You can find the script used in Flipside to get the data and the main Python code used to train and test the model in the Appendix section, but you can access the all code in our repository (file `analytics.ipynb`). Follow a summary of the application and results:

Model Choice

We chose to use the **Gradient Boosting Regressor** due to the following key reasons:

- Gradient Boosting is a powerful machine learning technique and particularly effective for regression tasks and can handle a variety of data types and distributions.
- Financial data often exhibit non-linear relationships, and Gradient Boosting can capture these effectively.
- It can evaluate the importance of different features, which is valuable for understanding the impact of factors like TVL, `FED_RATE`, etc.

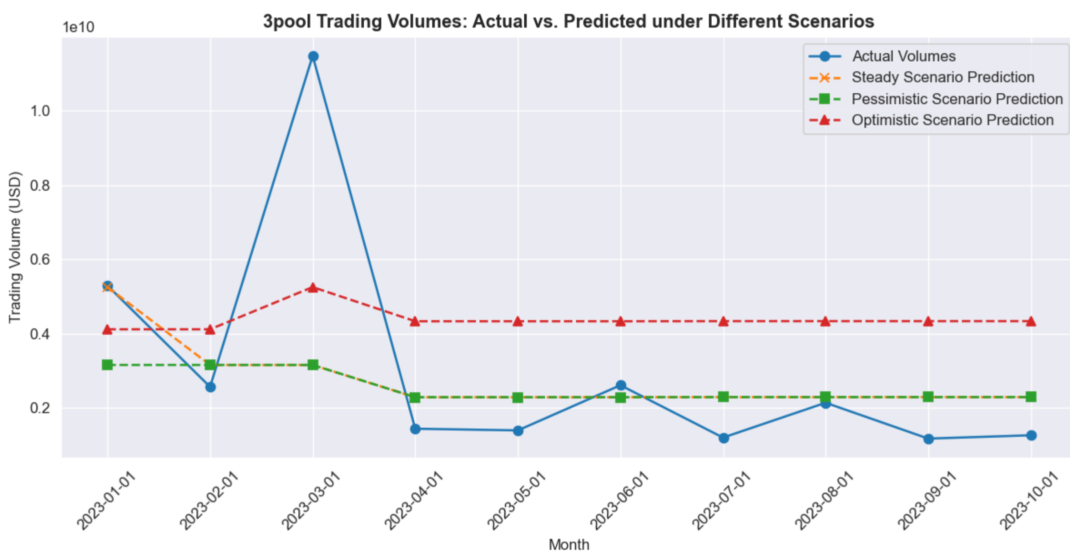
Features Used

The model incorporated the following features (variables) to predict the trading volumes for the Curve Finance 3pool (DAI-USDC-USDT):

1. **Total Value Locked (TVL)**: Represents the total amount of assets locked in the pool, a crucial indicator of liquidity and pool activity.
2. **Federal Funds Rate (FED_RATE)**: Serves as a proxy for broader market conditions, influencing investor behavior and market dynamics.
3. **Scenario-based Federal Funds Rates**: Adjusted versions of the federal funds rate were used to simulate different market conditions:
    - `FED_RATE`**:** In the **steady scenario**, the model used the unaltered federal funds rate to predict the volume.
    - `FED_RATE_PESSIMISTIC`**:** In the **pessimistic scenario**, a 15% increase rate was used.
    - `FED_RATE_OPTIMISTIC`**:** In the **optimistic scenario**, a 15% decrease rate was applied.

Results

The Gradient Boosting model's predictions for the trading volume were plotted against the actual volumes for the last 10 months of the dataset. The model was tested under three different scenarios: steady (using the original FED_RATE), pessimistic (with an increased FED_RATE), and optimistic (with a decreased FED_RATE). The visualization of these predictions alongside the actual data provided insights into the model's performance under varying market conditions, as indicated in the next chart:



Here are the evaluation metrics:

- **Mean Absolute Error (MAE)**: Approximately 1.65 billion
- **Mean Squared Error (MSE)**: Approximately 7.95 x 10^18
- **Root Mean Squared Error (RMSE)**: Approximately 2.82 billion

These metrics, particularly the MAE and RMSE, provide a sense of the average error in the model's predictions. The high values suggest that the model's predictions are, on average, billions of dollars off from the actual trading volumes. This could be due to various factors, including the inherent volatility and complexity of financial markets, the limited number of features used, or the need for more sophisticated modeling technique

This outcome aligns with the expectation that more substantial changes in market conditions (represented here by a larger adjustment in the federal funds rate) would have a more pronounced impact on trading volumes. It's important to note, however, that the real-world relationship between these factors can be complex and influenced by many other variables not included in this model.

‼ It's also important consider that unusual events like hacks in Curve DEX can affect our model, what open even more possible scenario like:

1. **Immediate Response**: A hack may cause a rapid decline in TVL and a spike in trading volumes due to urgent transactions. The model might not accurately predict these abrupt changes.
2. **Long-Term Outlook**: Persistent loss of trust could lead to continued low TVL and trading volumes. The model's historical data-based predictions may not reflect such sustained impacts.
3. **Recovery Scenario**: If Curve Finance swiftly mitigates the hack's effects and reassures users, the negative impact on trading volumes could be minimized. The model's accuracy would depend on its incorporation of recent recovery trends.

Apart from tha , we previously knew that the FED increased the interest rate in the last 10 months, then the steady and pessimistic scenario of our model fits better the prediction. Considering the market condition getting between in the next months, we can use the optimistic scenario to stipulate the future trading volume of Curve.

4. References 📚

- Project's Dashboard: https://flipsidecrypto.xyz/Rodolfo-Lima/curve-trading-volume-dynamics-SL0ga8
- Curve Docs: https://resources.curve.fi
- AMM Slippage: https://alvarofeito.com/articles/curve/#Slippage
- Liquidity in AMMs: https://rodolfo-projects.notion.site/Market-Challenge-a202a2a8513d48788c29078f6e057fc4
- Curve Wars: https://decrypt.co/90276/defi-bribes-are-on-the-rise
- Arbitrage: https://www.blocknative.com/blog/mev-and-creating-a-basic-arbitrage-bot-on-ethereum-mainnet
- Reentrance hack: https://mpost.io/the-aftermath-of-the-curve-finance-hack/

5. Appendix 📂

- **SQL script used to retrieve data from 3pool to forecast:**

```
-- Federal Funds Rate Series - ⚡LiveQuery
WITH interest_rate AS (
  WITH fred_data as (
  SELECT
    fred.get_series({
      'series_id': 'FEDFUNDS',
      'file_type': 'json',
      'observation_start': '2020-05-01',
```

```
      'frequency': 'm'
    }) as result
  )

  SELECT
    array.value:date::STRING AS month,
    array.value:value::FLOAT AS fed_rate
  FROM
    fred_data,
    LATERAL FLATTEN(input => TO_VARIANT(result:data:observations)) as array
),

-- Trading Volume
vol_tab AS (
  SELECT
    DATE_TRUNC('month', block_timestamp) AS month,
    SUM(COALESCE(AMOUNT_IN_USD,0)) AS vol_usd
  FROM
    ethereum.defi.ez_dex_swaps
  WHERE
    contract_address = '0xbebc44782c7db0a1a60cb6fe97d0b483032ff1c7' --3pool
    AND COALESCE(AMOUNT_IN_USD,0) > 0
  GROUP BY
    month
),

-- TVL
tvl_tab AS (
  WITH rank_balance AS (
    SELECT
      DATE_TRUNC('month', block_timestamp) AS balance_date,
      user_address,
      current_bal_usd,
      ROW_NUMBER() OVER(PARTITION BY user_address, contract_address, DATE_TRUNC('month', block_timestamp) ORDER BY block_timestamp DESC) as
    FROM
      ethereum.core.ez_balance_deltas
    WHERE
      (current_bal_usd > 0 OR current_bal_usd != NULL)
      AND symbol !=  ''
      AND user_address = '0xbebc44782c7db0a1a60cb6fe97d0b483032ff1c7' --3pool
  )

  SELECT
    balance_date,
    SUM(current_bal_usd) AS tvl
  FROM
    rank_balance
  WHERE
    row_id = 1
  GROUP BY balance_date
)

-- Final dataset
SELECT
  DATE(v.month) as month,
  vol_usd,
  tvl,
  fed_rate
FROM
  vol_tab v
  LEFT JOIN tvl_tab t ON v.month = t.balance_date
  INNER JOIN interest_rate i ON v.month = i.month
ORDER BY month ASC
```

- **Python Script utilized to prepare the data and build the model:**

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.ensemble import GradientBoostingRegressor
from sklearn.metrics import mean_absolute_error, mean_squared_error

# Feature Engineering for Market Condition Scenarios:
```

```
# 1. Steady Market Condition - Use the existing FED_RATE

# 2. Pessimistic Market Condition
coef = 0.15
df_pool['FED_RATE_PESSIMISTIC'] = df_pool['FED_RATE'] * ( 1 + coef)

# 3. Optimistic Market Condition
df_pool['FED_RATE_OPTIMISTIC'] = df_pool['FED_RATE'] * (1 - coef)
df_pool['FED_RATE_OPTIMISTIC'] = df_pool['FED_RATE_OPTIMISTIC'].clip(lower=0)

df_pool = df_pool.sort_values(by='MONTH')

# Splitting the data into train and test datasets
train_data = df_pool[:-10]  # First 32 months for training
test_data = df_pool[-10:]   # Last 10 months for testing

# Preparing features and target variable
X_train = train_data[['TVL', 'FED_RATE', 'FED_RATE_PESSIMISTIC', 'FED_RATE_OPTIMISTIC']]
y_train = train_data['VOL_USD']

X_test = test_data[['TVL', 'FED_RATE', 'FED_RATE_PESSIMISTIC', 'FED_RATE_OPTIMISTIC']]
y_test = test_data['VOL_USD']

# Train the Gradient Boosting Regressor
model = GradientBoostingRegressor(random_state=0)
model.fit(X_train, y_train)

# Predict and evaluate the model
predictions = model.predict(X_test)
mae = mean_absolute_error(y_test, predictions)
mse = mean_squared_error(y_test, predictions)
rmse = np.sqrt(mse)

# Evaluation metrics
mae, mse, rmse
```