The Gitcoin FDD Workstream and the Open Data Community Present

On-Chain Behavior Segmentation

A Practical Guidebook for Wallet Segmentation, Exploration and Insight Gathering

Introduction and Motivation

The Fraud Detection & Defense (FDD) workstream was open-source funding's first line of defense against malicious sybil attackers who attempt to manipulate Gitcoin's donor reward system. Consisting of high caliber developers, data scientists, and digital detectives, the team worked tirelessly to both proactively prevent and reactively thwart active threats to Web3's flagship public funding protocol, while continuously improving fraud defense detection processes and contributors' behavior and tendencies analyses. Through their experience defending the Gitcoin protocol, the team has developed advanced analytical approaches for the analysis and understanding of on-chain behavior of wallet accounts. To this end, the team proposed a wallet segmentation analysis that would help Aave better understand the behaviors of their user base & characterize platform utilization. To that end, this report and presentation are a culmination of those efforts.

Our goal is to enhance Aave's understanding of their current and future user base by performing a wallet segmentation analysis that will unearth insights into how groups of users borrow, lend, and interact with the Aave platform. Successful completion of this project will produce personas of wallet addresses that will:

1. Provide insight into Aave platform usage and group behavior.

2. Guide and direct marketing and product development.

3. Produce reproducible analysis that can be used to leverage and identify user behavioral pattern (usage, attitudes, loyalty)

At a high level, the process of developing on-chain behavioral personas first began with a transversal of the subgraphs of the relevant Aave contracts for all wallets that had made one or more event calls. This effectively defined our population sample as the subset of entities that used gas to successfully call any one of the functions enabled by the contracts deployed by Aave. This list of wallets would form a snapshot of all the Aave platform users up until that point in time. With this list, we collect the counts of their contract calls and the types. We then collected relevant platform agnostic characteristic data, such as the wallet balance, and active chains for each wallet, to serve as demographic data that could be utilized in the segmentation model or as exogenous, descriptive variables that could further highlight differences between the final user profiles. It is at this point the data was transformed and fed into an unsupervised machine learning model that best grouped the observations according to behavioral and descriptive similarity. These groups were then visualized and re-expressed as the easy to understand personas that are shared at the end of this document.

Data Details and Definitions

Before data collection could begin, a discovery step had to be initiated to identify and track each of the contracts deployed by Aave. Contracts for the platform have been well documented across multiple sources, where the most relevant can be found for V1, V2, and V3 here, here, and here. It should be noted that not every callable method corresponds to a relevant behavior of interest. To that end, we focused only on events that mapped directly to behaviors to be used within the segmentation model. These contract methods were:

- deposit()
- supply()
- borrow()
- repay()
- withdraw()
- redeemUnderlying()
- liquidationCall)
- flashLoan()

Savvy and experienced Aave developers will immediately notice that these contract calls contain events that have been depreciated across version changes. Most notably, the deposit() method was removed, leaving only the supply() call to indicate deposits into the platform. Likewise, the redeem() method from V1 was replaced with the withdraw() call to indicate when users successfully removed liquidity from the platform. These changes in methods were ultimately reflected in the analysis through the merging of these calls into their respective singular behaviors.

The most robust solution for querying these contracts would be to host a local Infura node and search across an indexed portion of blocks. Due to resource constraints, we opted to work around this requirement by utilizing the recently enabled data export feature by Dune Analytics. With their platform we were able to query each contract for emitted events and the wallets connected to them.

```sql
WITH
--------------------------------------------------------------------------------
-- AAVEV1 ~40k
--------------------------------------------------------------------------------
-- AAVEV1 Ethereum ~40k
aavev1_eth as
(
    SELECT DISTINCT "_user" as address FROM aave_ethereum.LendingPool_evt_Borrow
    UNION ALL
    SELECT DISTINCT "_user" as address FROM aave_ethereum.LendingPool_evt_Deposit
    UNION ALL
    SELECT DISTINCT "_target" as address FROM aave_ethereum.LendingPool_evt_FlashLoan
    UNION ALL
    SELECT DISTINCT "_user" as address FROM aave_ethereum.LendingPool_evt_LiquidationCall
    UNION ALL
    SELECT DISTINCT "_user" as address FROM aave_ethereum.LendingPool_evt_Repay
    UNION ALL
    SELECT DISTINCT "_user" as address FROM aave_ethereum.LendingPool_evt_RedeemUnderlying
),
aave_v1 as
(
    SELECT DISTINCT address from aavev1_eth
),
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------


--------------------------------------------------------------------------------
-- AAVEV2 ~250k
--------------------------------------------------------------------------------
-- AAVEV2 Avalanche ~25k
aavev2_avl as
(
    SELECT DISTINCT "user" as address FROM aave_v2_avalanche_c.LendingPool_evt_Borrow
    UNION ALL
    SELECT DISTINCT "user" as address FROM aave_v2_avalanche_c.LendingPool_evt_Deposit
    UNION ALL
    SELECT DISTINCT "initiator" as address FROM aave_v2_avalanche_c.LendingPool_evt_FlashLoan
    UNION ALL
    SELECT DISTINCT "user" as address FROM aave_v2_avalanche_c.LendingPool_evt_LiquidationCall
    UNION ALL
    SELECT DISTINCT "user" as address FROM aave_v2_avalanche_c.LendingPool_evt_Repay
    UNION ALL
    SELECT DISTINCT "user" as address FROM aave_v2_avalanche_c.LendingPool_evt_Withdraw
),

-- AAVEV2 Ethereum ~95k
aavev2_eth as
(
    SELECT DISTINCT "user" as address FROM aave_v2_ethereum.LendingPool_evt_Borrow
    UNION ALL
    SELECT DISTINCT "user" as address FROM aave_v2_ethereum.LendingPool_evt_Deposit
    UNION ALL
    SELECT DISTINCT "initiator" as address FROM aave_v2_ethereum.LendingPool_evt_FlashLoan
    UNION ALL
    SELECT DISTINCT "user" as address FROM aave_v2_ethereum.LendingPool_evt_LiquidationCall
    UNION ALL
    SELECT DISTINCT "user" as address FROM aave_v2_ethereum.LendingPool_evt_Repay
    UNION ALL
    SELECT DISTINCT "user" as address FROM aave_v2_ethereum.LendingPool_evt_Withdraw
),

-- AAVEV2 Polygon ~160k
aavev2_pol as
(
    SELECT DISTINCT "user" as address FROM aave_v2_polygon.LendingPool_evt_Borrow
    UNION ALL
    SELECT DISTINCT "user" as address FROM aave_v2_polygon.LendingPool_evt_Deposit
    UNION ALL
    SELECT DISTINCT "initiator" as address FROM aave_v2_polygon.LendingPool_evt_FlashLoan
    UNION ALL
    SELECT DISTINCT "user" as address FROM aave_v2_polygon.LendingPool_evt_LiquidationCall
    UNION ALL
    SELECT DISTINCT "user" as address FROM aave_v2_polygon.LendingPool_evt_Repay
    UNION ALL
    SELECT DISTINCT "user" as address FROM aave_v2_polygon.LendingPool_evt_Withdraw
),

-- AAVEV2 All Chains ~250k
aavev2_t as
(
    SELECT DISTINCT address from aavev2_avl
    UNION ALL
    SELECT DISTINCT address from aavev2_eth
    UNION ALL
    SELECT DISTINCT address from aavev2_pol
),
aave_v2 as
(
    SELECT DISTINCT address from aavev2_t
),
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------


--------------------------------------------------------------------------------
-- AAVE V1 + V2 ~283k
--------------------------------------------------------------------------------
aave_users_t as
(
    SELECT DISTINCT address from aave_v1
    UNION ALL
    SELECT DISTINCT address from aave_v2
),
aave_users as
(
    SELECT DISTINCT address from aave_users_t
)
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------

SELECT * from aave_users
```

Figure 1: SQL queries for searching for wallet addresses that interacted with the Aave platform through the Dune Analytics front end.

When querying across the six chains Aave has been deployed on, Optimisim, Arbitrum, Polygon, Ethereum, Avalanche, and Fantom, we found approximately 465,000 unique wallets. At this stage we now have, for every wallet address, a structured table of the contract events for each chain and version. In the data, the columns have been denoted with prefixes as: C for chain, E for events, V for versions.

This lends itself to the creation of seven unique combinations of where a standalone V_ indicates the version, VE_ means Version and Event Wise, and CVE_ denotes the Chain, Version and Event. As an example**, CVE_polygon_v3_Withdraw_calls** represents the number of withdraw() calls for wallets interacting with the v3 version of Aave on the Polygon network.

Demographic comparisons across profiles will be of interest and to support this task, additional data was collected from other on chain sources such as DeBank, the web3 social media platform Lens Protocol, off-chain governance platform Snapshot, Gitcoin's proof of personhood Passport and the experimental sybil score created by Trustalabs. Since most of these platforms exist only on the Ethereum mainnet, we were forced to constrain our analysis to only wallet addresses that interacted with Aave on that blockchain. This resulted in a final dataset of 114,915 wallet addresses with complete records suitable for segmentation through an unsupervised machine learning routine.

address	CV_polygon_v2_calls	CE_fantom_Repay_calls	VE_v2_Repay_calls	CVE_ethereum_v1_Borrow_calls	CVE_optimism_v3_LiquidationCall_calls	aave_debt_usd
C_arbitrum_calls	CV_polygon_v3_calls	CE_fantom_Supply_calls	VE_v2_Withdraw_calls	CVE_ethereum_v1_Deposit_calls	CVE_optimism_v3_Repay_calls	aave_net_usd
C_avalanche_calls	CE_arbitrum_Borrow_calls	CE_fantom_Withdraw_calls	VE_v3_Borrow_calls	CVE_ethereum_v1_FlashLoan_calls	CVE_optimism_v3_Supply_calls	if_lens_data
C_ethereum_calls	CE_arbitrum_FlashLoan_calls	CE_optimism_Borrow_calls	VE_v3_FlashLoan_calls	CVE_ethereum_v1_LiquidationCall_calls	CVE_optimism_v3_Withdraw_calls	lens_prof_count
C_fantom_calls	CE_arbitrum_LiquidationCall_calls	CE_optimism_FlashLoan_calls	VE_v3_LiquidationCall_calls	CVE_ethereum_v1_RedeemUnderlying_calls	CVE_polygon_v2_Borrow_calls	lens_id
C_optimism_calls	CE_arbitrum_Repay_calls	CE_optimism_LiquidationCall_calls	VE_v3_Repay_calls	CVE_ethereum_v1_Repay_calls	CVE_polygon_v2_Deposit_calls	lens_name
C_polygon_calls	CE_arbitrum_Supply_calls	CE_optimism_Repay_calls	VE_v3_Supply_calls	CVE_ethereum_v2_Borrow_calls	CVE_polygon_v2_FlashLoan_calls	lens_isDefault
E_Borrow_calls	CE_arbitrum_Withdraw_calls	CE_optimism_Supply_calls	VE_v3_Withdraw_calls	CVE_ethereum_v2_Deposit_calls	CVE_polygon_v2_LiquidationCall_calls	lens_followers
E_Deposit_calls	CE_avalanche_Borrow_calls	CE_optimism_Withdraw_calls	CVE_arbitrum_v3_Borrow_calls	CVE_ethereum_v2_FlashLoan_calls	CVE_polygon_v2_Repay_calls	lens_following
E_FlashLoan_calls	CE_avalanche_Deposit_calls	CE_polygon_Borrow_calls	CVE_arbitrum_v3_FlashLoan_calls	CVE_ethereum_v2_LiquidationCall_calls	CVE_polygon_v2_Withdraw_calls	lens_posts
E_LiquidationCall_calls	CE_avalanche_FlashLoan_calls	CE_polygon_Deposit_calls	CVE_arbitrum_v3_LiquidationCall_calls	CVE_ethereum_v2_Repay_calls	CVE_polygon_v3_Borrow_calls	lens_comments
E_RedeemUnderlying_calls	CE_avalanche_LiquidationCall_calls	CE_polygon_FlashLoan_calls	CVE_arbitrum_v3_Repay_calls	CVE_ethereum_v2_Withdraw_calls	CVE_polygon_v3_FlashLoan_calls	lens_mirrors
E_Repay_calls	CE_avalanche_Repay_calls	CE_polygon_LiquidationCall_calls	CVE_arbitrum_v3_Supply_calls	CVE_ethereum_v3_Borrow_calls	CVE_polygon_v3_LiquidationCall_calls	lens_publications
E_Supply_calls	CE_avalanche_Supply_calls	CE_polygon_Repay_calls	CVE_arbitrum_v3_Withdraw_calls	CVE_ethereum_v3_FlashLoan_calls	CVE_polygon_v3_Repay_calls	lens_collects
E_Withdraw_calls	CE_avalanche_Withdraw_calls	CE_polygon_Supply_calls	CVE_avalanche_v2_Borrow_calls	CVE_ethereum_v3_LiquidationCall_calls	CVE_polygon_v3_Supply_calls	snap_voted_aave
V_1_calls	CE_ethereum_Borrow_calls	CE_polygon_Withdraw_calls	CVE_avalanche_v2_Deposit_calls	CVE_ethereum_v3_Repay_calls	CVE_polygon_v3_Withdraw_calls	snap_voted_aave_numprop
V_2_calls	CE_ethereum_Deposit_calls	VE_v1_Borrow_calls	CVE_avalanche_v2_FlashLoan_calls	CVE_ethereum_v3_Supply_calls	if_debank_data	trustalabs_is
V_3_calls	CE_ethereum_FlashLoan_calls	VE_v1_Deposit_calls	CVE_avalanche_v2_LiquidationCall_calls	CVE_ethereum_v3_Withdraw_calls	agedays	trustalabs_score
CV_arbitrum_v3_calls	CE_ethereum_LiquidationCall_calls	VE_v1_FlashLoan_calls	CVE_avalanche_v2_Repay_calls	CVE_fantom_v3_Borrow_calls	iscontract	trustalabs_label
CV_avalanche_v2_calls	CE_ethereum_RedeemUnderlying_calls	VE_v1_LiquidationCall_calls	CVE_avalanche_v2_Withdraw_calls	CVE_fantom_v3_FlashLoan_calls	ismultisig	
CV_avalanche_v3_calls	CE_ethereum_Repay_calls	VE_v1_RedeemUnderlying_calls	CVE_avalanche_v3_Borrow_calls	CVE_fantom_v3_LiquidationCall_calls	num_chains_active	
CV_ethereum_v1_calls	CE_ethereum_Supply_calls	VE_v1_Repay_calls	CVE_avalanche_v3_FlashLoan_calls	CVE_fantom_v3_Repay_calls	num_tokens	
CV_ethereum_v2_calls	CE_ethereum_Withdraw_calls	VE_v2_Borrow_calls	CVE_avalanche_v3_LiquidationCall_calls	CVE_fantom_v3_Supply_calls	aave_amt	
CV_ethereum_v3_calls	CE_fantom_Borrow_calls	VE_v2_Deposit_calls	CVE_avalanche_v3_Repay_calls	CVE_fantom_v3_Withdraw_calls	bal_usd	
CV_fantom_v3_calls	CE_fantom_FlashLoan_calls	VE_v2_FlashLoan_calls	CVE_avalanche_v3_Supply_calls	CVE_optimism_v3_Borrow_calls	eth_bal_usd_	
CV_optimism_v3_calls	CE_fantom_LiquidationCall_calls	VE_v2_LiquidationCall_calls	CVE_avalanche_v3_Withdraw_calls	CVE_optimism_v3_FlashLoan_calls	aave_asset_usd	

Figure 2: Columns of the full superset of data collected across all data sources, blockchains and versions.

The final data dictionary for the dataset is as follows:

num_chains_active: Number of chains this wallet has interacted with.

num_tokens_eth: Number of tokens held on the Ethereum blockchain.

bal_usd_eth: Total balance, in USD held by that wallet on the Ethereum blockchain.

aave_asset_usd_eth: Total amount of assets, in USD, supplied to the Aave platform.

aave_debt_usd_eth: Total amount of debt, in USD, borrowed from the Aave platform.

CE_ethereum_Deposit_calls: Total number of deposit events recorded for this address on the Ethereum blockchain.

CE_ethereum_Withdraw_calls: Total number of withdraw and/or redeem events recorded for this address on the Ethereum blockchain.

CE_ethereum_Borrow_calls: Total number of borrow events recorded for this address on the Ethereum blockchain.

CE_ethereum_FlashLoan_calls: Total number of flash loan events recorded for this address on the Ethereum blockchain.

CE_ethereum_Repay_calls: Total number of repay events recorded for this address on the Ethereum blockchain.

CE_ethereum_LiquidationCall_calls: Total number of liquidations events recorded for this address on the Ethereum blockchain.

if_debank_data: An indicator variable for whether we have Debank data for that particular address.

if_lens_data: An indicator variable of whether an address has a lens profile. If not, an NA is recorded.

snap_voted_aave: An indicator variable if whether an address has voted on any Aave snapshot proposals.

snap_voted_aave_numprop: Total number of proposals voted on by that wallet address on the Snapshot platform. Zeros are represented as NAs.

trustalabs_is: An indicator of whether a wallet address has a TrustaLabs sybil score.

trustalabs_score: A quantitative variable recording the TrustaLabs sybil risk score. Scores range from 0 to 100.

passport_is: An indicator of whether a wallet address has a Gitcoin Passport.

passport_numstamps: A count variable recording the number of stamps associated with that user's Gitcoin Passport.

Exploratory Analysis of Aave Contracts and Wallets

An integral part of this segmentation study hinges on the exploratory analysis of the data to understand activity at the chain, contract, and function call level. It is at these different views can we begin to understand how usage differs, and it is these differences that begin to inform how individual wallet accounts may be clustered as we introduce more complexity into the partitioning of the data. At the highest level, across all chains, the average number of contract functions calls for any arbitrary wallet was 14. While the mean number of calls was in the double digits, from the median and up to the 3rd quartile there was only 1 contract call, suggesting that many of the wallets interacted once and only once. This is evident by the high skew of the distribution and is actually quite pronounced when we break up the number of calls by the chain.

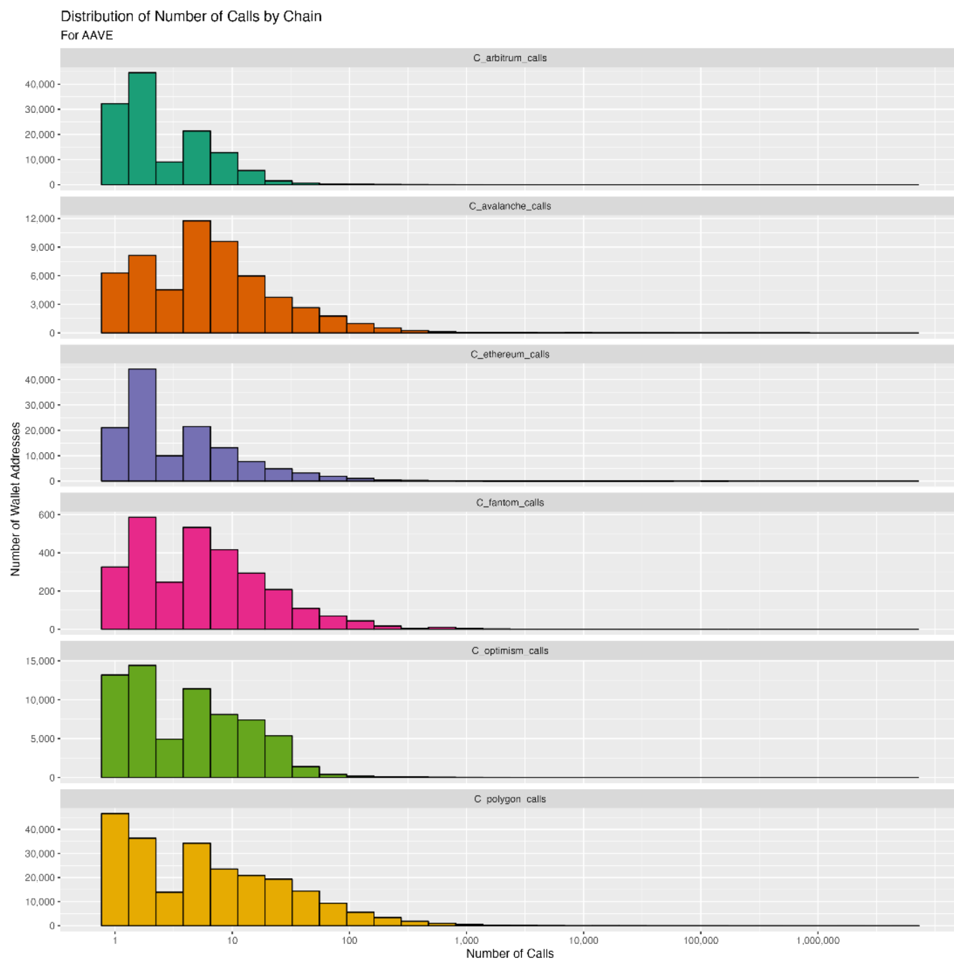

Figure 3: The distribution of the number of contract calls within each chain with an Aave deployment.

Despite having been established for the longest period of time, the Aave contracts deployed on the Ethereum blockchain have only the third largest number of contract calls with 1,766,897. When continuing to look at the total number of contract calls, the effect of lower fees on contract engagement is apparent as the Polygon deployment has a staggering 29,399,488 total event calls with one single address generating 5,577,529 of them. The revelation that a single wallet address can generate so many contract calls provides inspiration for our first deep dive into individual addresses, one that quickly exposes a point later particularly relevant for the segmentation of accounts.

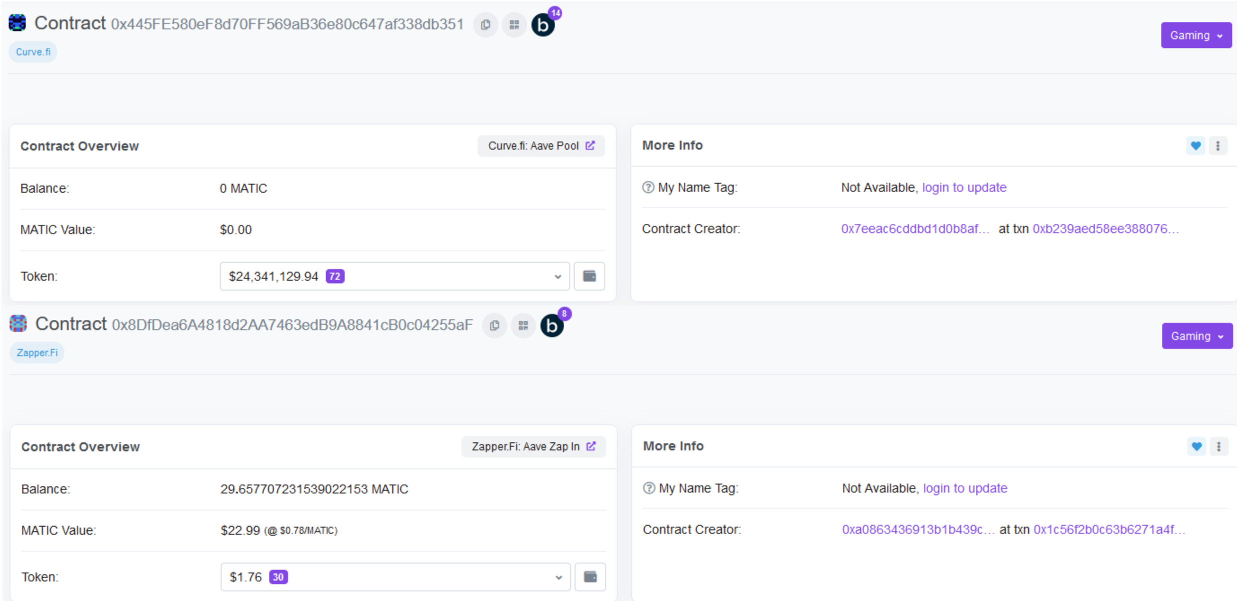

Figure 4: The wallet addresses with the most transactional calls to the Aave contracts on the Polygon Network happened to be contracts themselves.

Zapper.fi and Curve.fi offer users the ability to interact with the Aave platform in various ways from their own platform. The popularity of these projects produces a substantial amount of the transactional traffic with Aave on the Polygon blockchain, which is a useful insight to have at a high level, but will obfuscate the goal of our analysis; to categorize, understand and segment individual user behavior. With this knowledge on hand, knowing that contracts generate a disproportionate amount of traffic and act as a confounding force that aggregates the behavior of multiple users behind one point of contact with the Aave deployments, we will seek to both identify and remove them from the segmentation analysis. Another similar externally owned account style wallet that also exhibits this confounding effect are multi-sig wallets. These are specialized contract wallets that allow a group of individual wallets addresses to deposit assets into the contract without restriction, but requires all out going transactions to be approved by one or more of the owners. Possessing a similar confounding effect as standard contracts, but with alternative usage, these types of wallets will also be removed from the analysis.

One can also look at the contract calls at the version level. This aggregation looks across blockchain and focuses directly on the version of the contracts deployed by Aave irrespective of network.

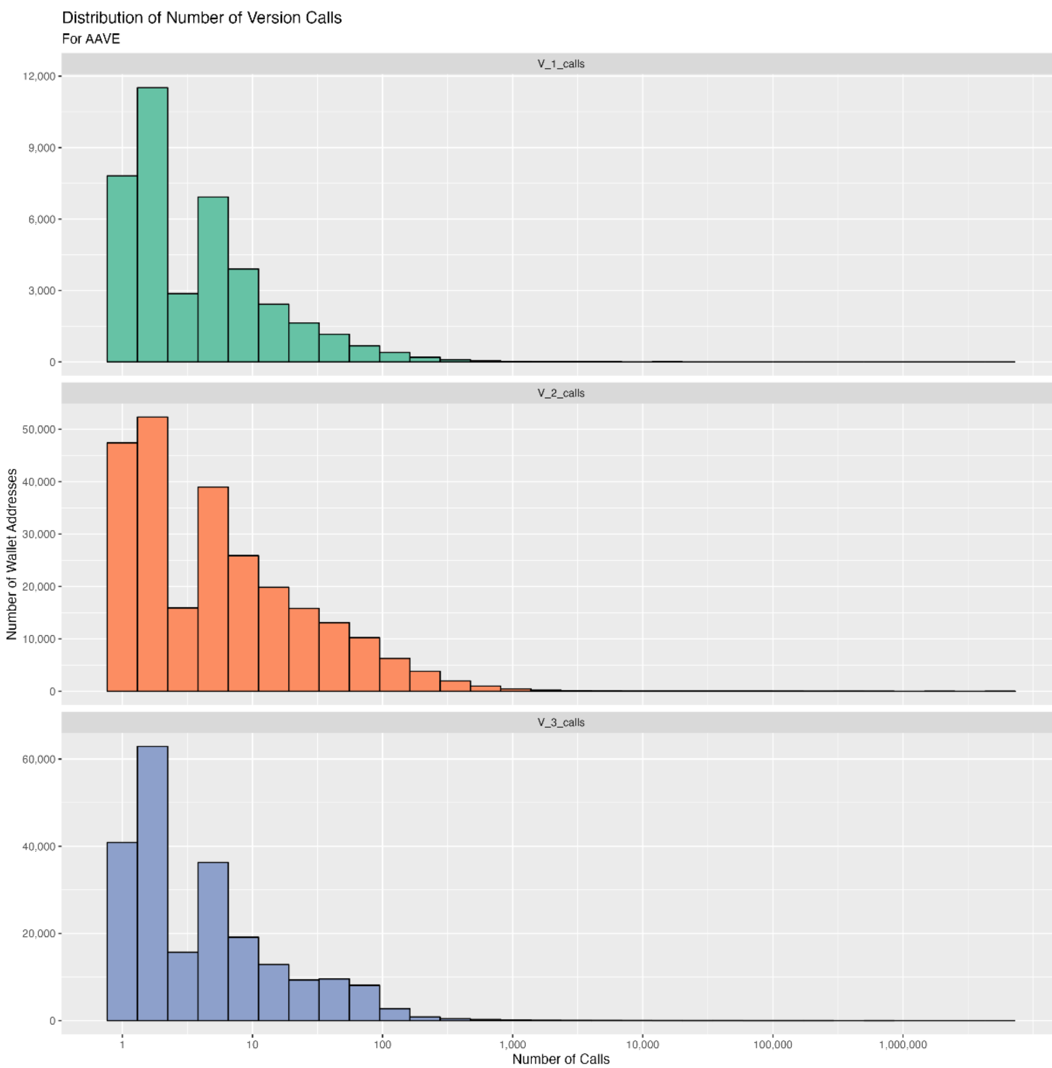

Figure 5: The Aave v2 deployment, which rests on Avalanche, Ethereum and Polygon, had the most contract interactions across all summary statistics.

Across nearly every number of calls tracked, there were more wallets interacting with the v2 version of the contract. This is because of the diversity of characteristics of the chains for which the v2 contract was deployed on. The Ethereum deployment is on the most reliable and oldest chain, while the Polygon deployment offers easy compatibility with Ethereum and has lower fees. The Avalanche deployment is on the most innovative chain of the three as it operates as an interconnected combination of three individual chains in one. At the time of this document's construction Aave v1 had approximately 500k contract calls, while v2 had 31 million with the v3 deployment across the same set of v2 chains has an up and coming number of 6.6 million calls.

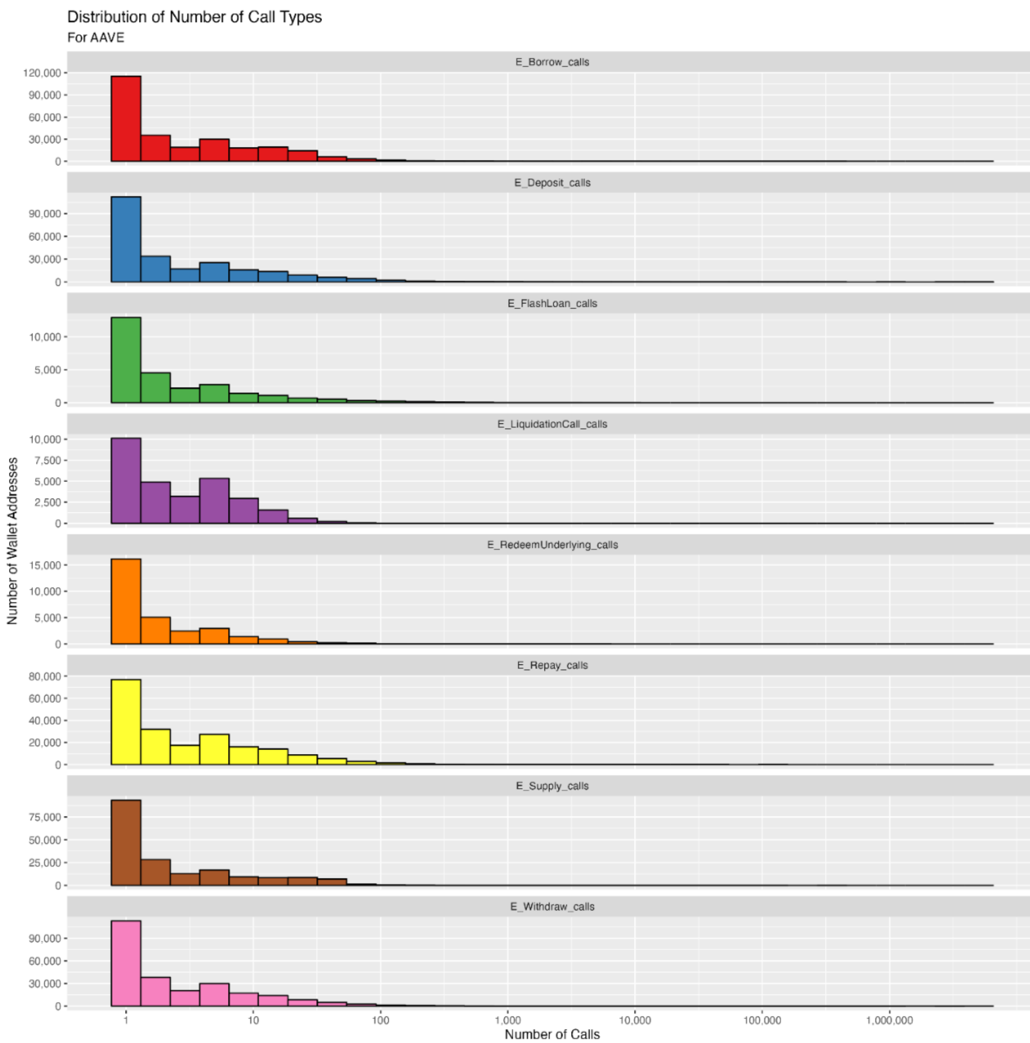

Figure 6: Distribution of events emitted from the Ethereum based mainnet Aave contracts.

As we drill down into the singular events that can be emitted from the Aave contracts, our intuition around user behavior immediately begins to form as we see which of the eight unique functions are called. Linearly, the comparison between the functional calls looks to be Deposits (16M) >> Withdraws (10M) >> Borrow (4.6M) > Supply (3M) > Repay (2.6M) >> Flashloan (356k) > RedeemUnderlying (145k) >> Liquidation (131k) for specific wallets having interacted with the Ethereum Aave contracts. Despite the histograms being heavily skewed, this disparity in events emitted is still obvious and apparent from the scales on the y-axis. Here is also where we begin our focus only on Ethereum based on wallet addresses because, as mentioned previously, data we are deeming "demographic" was only available for wallets on the Ethereum mainnet.

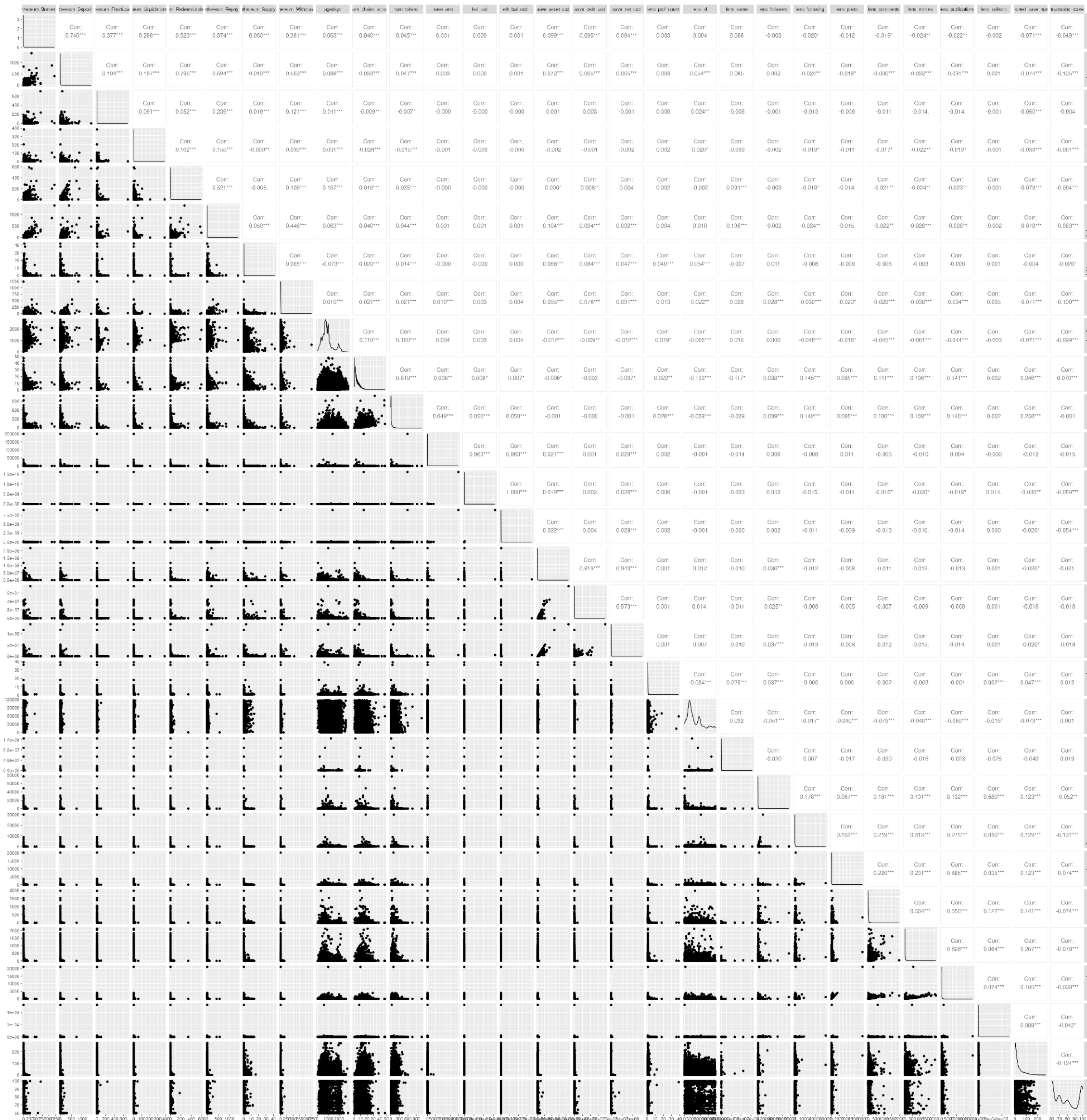

Figure 7: Zoomed out view of the scatterplot matrix calculated for all variables endogenous and exogenous to the segmentation model.

To continue our investigation, we took yet another high level view of the data, but this time with a focus on the linear pairwise correlations between the variables in the final set of data to be used in the segmentation analysis. The purpose of this task was to identify pairs of highly correlated features that could then have one of its members removed from the data to reduce complexity. Searching through the scatterplot matrix, a total of five sets of variables were found to have statistically significant correlations above .5. These were {aave_assets_usd, aave_debt_usd, .573}, {num_token, num_chains_active, .819}, {ethereum_repay_calls, ethereum_redeemunderlying_calls, .521}, {ethereum_repay_calls, ethereum_deposit_calls, .694}. These correlations are directly aligned with the intended use of the Aave platform, as users deposit then withdraw and take out flashloans then repay. Other relationships were intuitive, but not necessarily known, such as those who have a large number of tokens typically also are active on multiple chains. This isn't directly related to Aave in and of itself, but is an interesting insight of the population of wallets on the Ethereum blockchain that interacted with Aave. Through both the exercise of looking at correlations and partitioning traffic along the various versions did we realize that certain events had been renamed across different versions of the contracts, but retained their same functionality. This was true for the "Deposit", "Supply", "Redeem" and "Withdraw" contract calls. As an appeal to Occum's Razer, we merged the relevant calls according to the following rubric: **Supply** = Deposit (v1 and v2) and Supply (v3) while **Withdraw** = Redeem (v1) and Withdraw (v2 and v3).

Earlier in our exploration, the shape of the histograms exhibited a skew which motivated our search for outliers within the distribution. We more formally checked this through the use of the Mahalanobis distance metric, which is a multi-variate aggregation function that's unitless, scale invariant properties allow it to be used as a tool to assess the similarity of a single observation against a mean vector. In more common terms, the average across each variable in the data set is appended together in a vector and treated as the representative observation in the dataset. All other data points have their Mahalanobis distance calculated as the quantitative difference between that point and the mean vector. Observations with large distance values are deemed outliers. These distances were calculated for each wallet, but for the purposes of illustration, only the first 1000 values are plotted on the graph below.

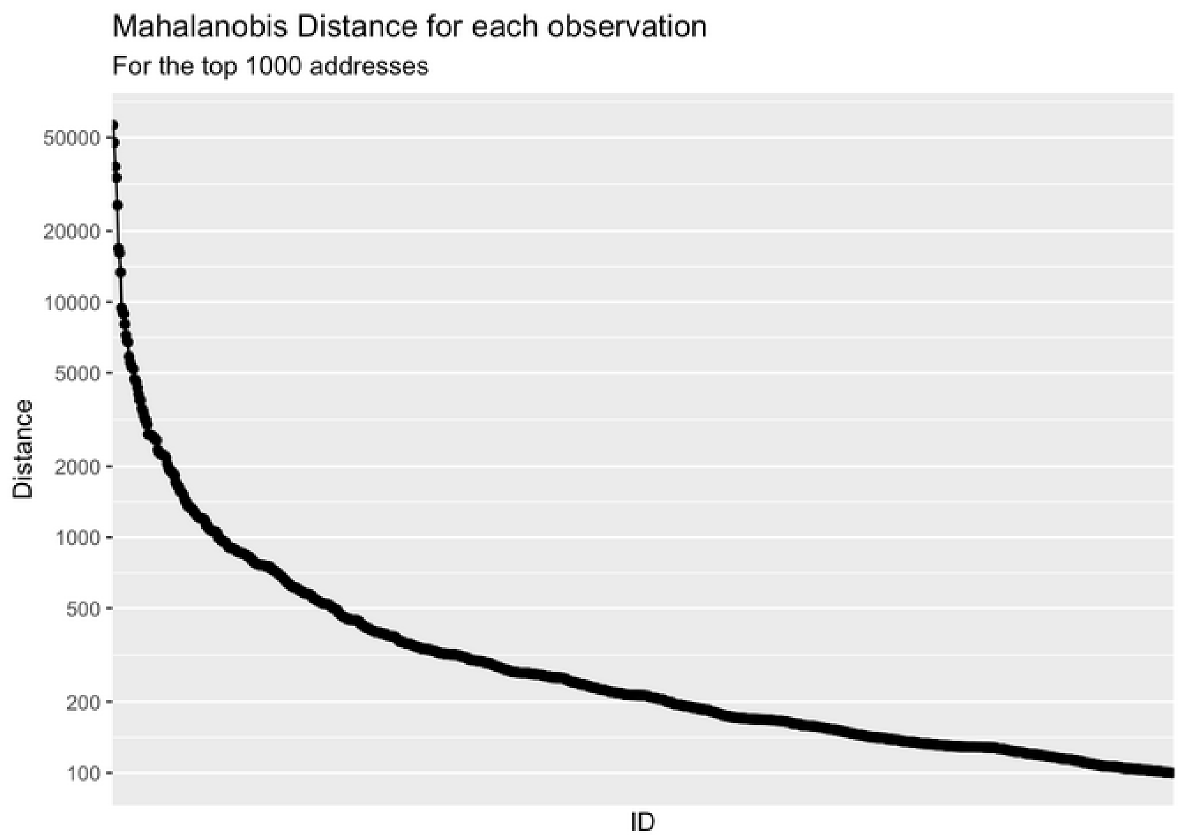

Figure 8: Malahanobis distances for the top 1000 addresses. The gap along the y-axis at 10,000 represents the cut-off where outliers exist.

For sorted dot plots like the one above, “elbows” and “gaps” are used to identify thresholds where outlying wallet addresses exist beyond the bounds. For this data, values above 10,000 appear to be outliers and when further investigated, these outliers are revealed to be institutional wallets, vault contracts and even degen whales. In this data, the top three outliers were one of **Binance's Hot Wallets** at 0xf977814e90da44bfa03b6295a0616a897441acec, **MakerDAO’s Vault** at 0x7d6149ad9a573a6e2ca6ebf7d4897c1b766841b4 and **patricioworthalter.eth** at 0x57757E3D981446D585Af0D9Ae4d7DF6D64647806. Given their networth’s are clear outliers and institutions are their own special class of users with usage likely to differ, we removed them from the dataset, yet kept individual whale accounts within the dataset.

The exploratory analysis journey culminated into a set of facts, justifications and verifications for a few of the assumptions and design choices for this study. Specifically, the analysis backs the removal of contracts and multi-signature wallets from the sample to reduce bias, the merging of function calls to reduce complexity, the removal of outliers to further reduce bias, and the intentional focus on Ethereum wallet addresses due to data availability.

Wallet Segmentation Methodology

With our population sample procured and the variable set fixed, we then initiated the wallet segmentation routine. As an unsupervised learning machine learning task, customer segmentation is a use case of cluster analysis applied to user characteristic and demographic data. The process involves a preliminary visualization to assess the linear and non-linear relationships between the observations, an ensemble clustering phase where multiple clustering algorithms are applied for varying number of clusters, the final solution is selected as the one with the least within group variance and most across group separation. The observations are then relabeled according to their cluster assignments and are then interpreted and explained using a combination of qualitative and visual references.

Quickly redefining our dataset, the 114,915 wallets we'll be developing profiles for have the majority of their transactional history on the Ethereum mainnet and will be characterized by the type of contract calls they make, the frequency they make them, and general demographic attributes such as the number of EVM compatible chains the wallet has interacted with. Before both visualization and the application of the clustering algorithms, the variables are to be normalized so that there is scale uniformity across each of the variables. We chose a simple three part procedure where, for each column, we first took the log(x+1) transformation, rescaled the values between 0 and 1 and then rounded to three digits.

Our first attempt at understanding the variance-covariance structure of the data as a whole will come from a scatterplot matrix of the pairwise principal components as constructed from a singular value decomposition of the data. The principal components immediately highlight the difficulty of the task as there is no clear separation of the wallet addresses into cohesive and compact clusters. The datum align themselves along the various axis, but do not form natural groupings, which suggests the relationships between the groups, if they exist, are non-linear in nature.

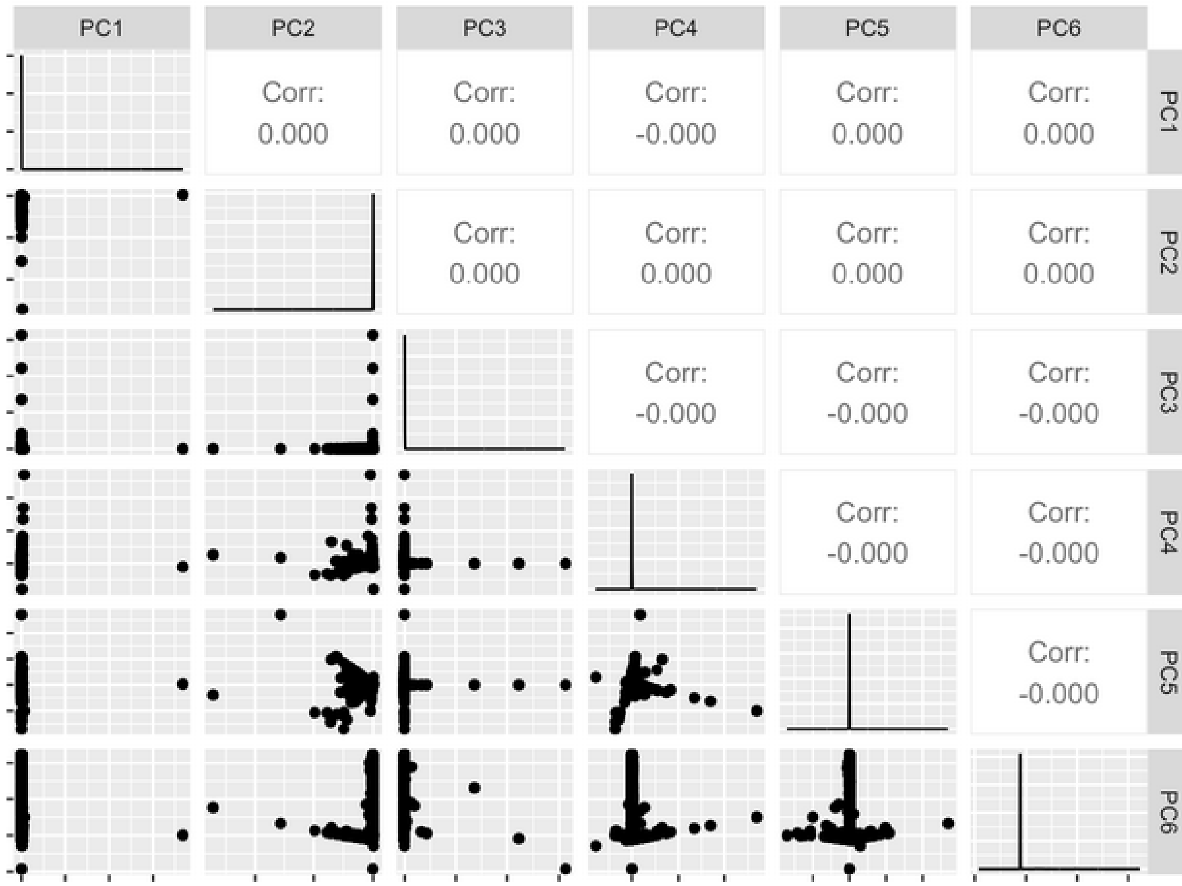

Figure 9: Scatterplot matrix of the first six principal components against each other.

When using traditional clustering methods such as Hierarchical Clustering, K-Means, Self Organizing Maps, K-Mediods, and DB Scan are applied they tend to create one large cluster, and for each subsequently chosen number of groups, they "peel" off a few observations and place them into a new group. This doesn't result in meaningful clusters, nor does the process for which the clusters are generated align with our intuition. The assumption is that all observations belong to one major group and every other group is just an off shoot of it, versus having their own distinct profile.

```
> apply(hclustWclusts,2,table)
[[1]]

     1      2
114809    106

[[2]]

     1      2      3
114809    103      3

[[3]]

     1      2      3      4
114753    103     56      3

[[4]]

     1      2      3      4      5
113725   1028    103     56      3

[[5]]

     1      2      3      4      5      6
103577  10148   1028    103     56      3

[[6]]

     1      2      3      4      5      6      7
103577  10148   1028    103     55      3      1

[[7]]

     1      2      3      4      5      6      7      8
103577  10148   1028    100     55      3      3      1

[[8]]

     1      2      3      4      5      6      7      8      9
103577  10148   1017    100     55      3     11      3      1

[[9]]

     1      2      3      4      5      6      7      8      9     10
103577  10148   1017    100     55      2     11      3      1      1
```

Figure 10: Sample clustering of the data that highlights the failures of traditional clustering based approaches.

In circumstances where linear distance based unsupervised techniques fail, non-linear manifold based techniques are leveraged. These methods come with a strong assumption, that there is a fundamental "shape" or "topology" that exists within the data, and it is this topology that the t-SNE and UMAP attempt to uncover.

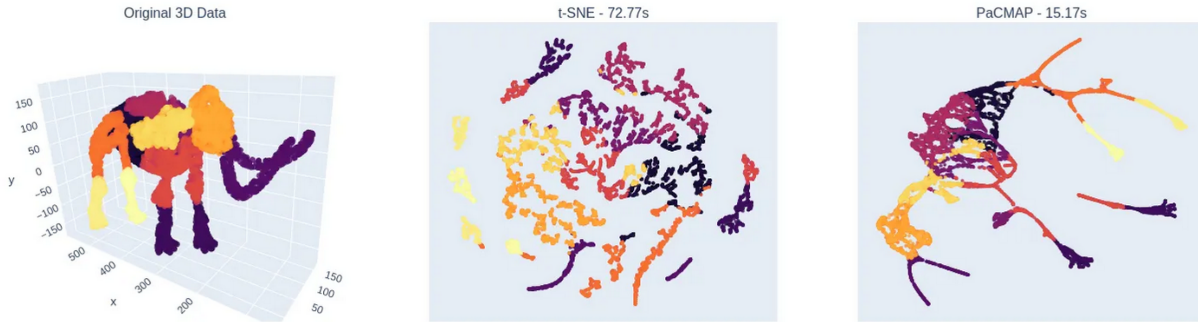

Figure 11: A sample demonstration of using non-linear manifold techniques to project multi-dimensional structures into 2D. Here the 3D elephant has its sections labelled and when projected using t-SNE and PaCMAP the components are maintained though rearranged.

When applied to the wallet addresses and their corresponding variable set, the 2D projection using standard t-SNE exhibits some structure that was not present during our first pass utilizing linear based techniques. Despite the clear circular nature of the data structure overall in the figure below, this pattern is not indicative of any intuitive high level relationship amongst all of the points. Instead, focus should be shifted to the grouped "peels" that appear almost as layers of an onion. Each one of these constructs would be considered a separate group; however, this is only one such configuration of the t-SNE algorithm.

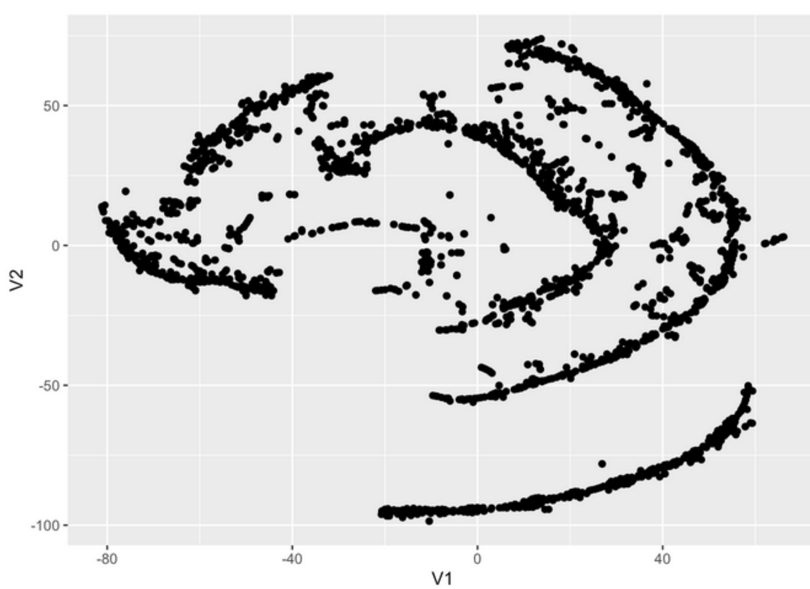

Figure 12: A 2D t-SNE generated projection of the wallet addresses within the feature space.

Like t-SNE, Uniform Manifold Approximation and Projection (UMAP) is a non-linear dimension reduction technique that must be parameterized during initialization. As part of the search routine, we looked across various parameter configurations to find 2D projections with meaningful separation. The following figure shows four such iterations where, going from left to right, top to bottom we see more distinct groups forming in the 2D space. As mentioned previously, the assumption is that groups of points are related and if those groups are sufficiently distant from other groups there will be some quantifiable difference in their values.

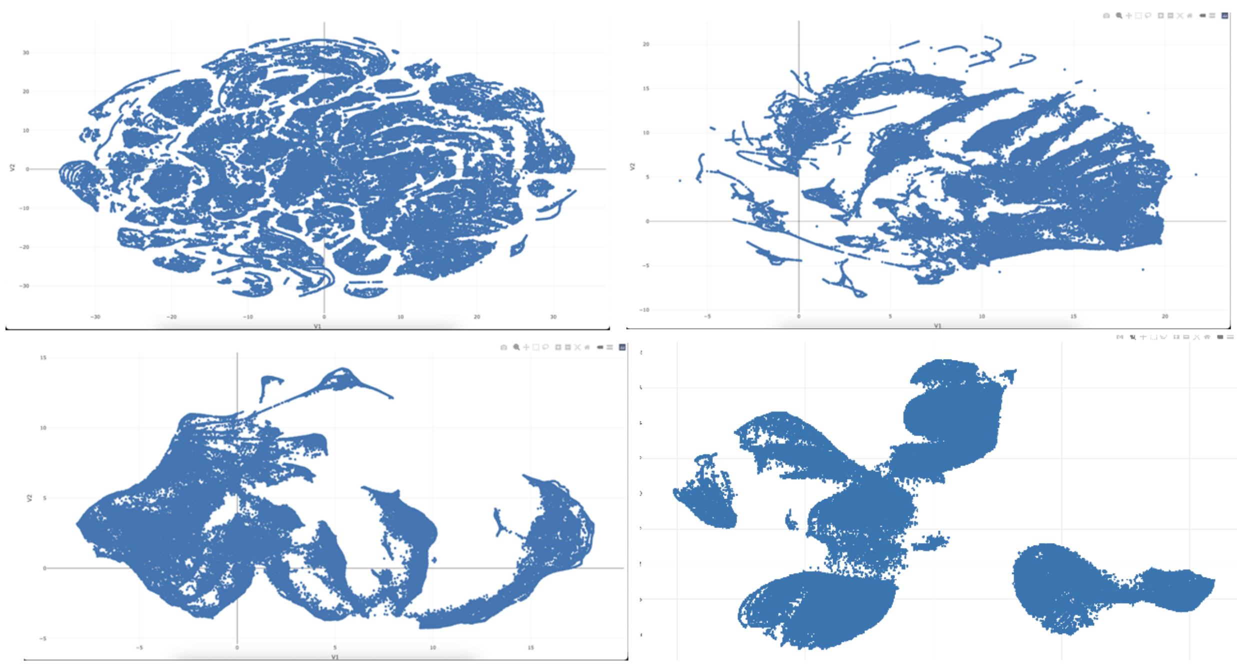

Figure 13: A visualization of the UMAP parameter search that culminated into the final solution on the bottom right.

Iterating through the parameter space concluded with a final visualization with three completely separable groups, each with several appendage like structures that, themselves, could be considered relatively separable. To further assess whether we were on the correct path, we then colored the individual wallets by one of the many k-means cluster solutions we found and then replotted all the points in the 2D UMAP space. Figure 15 indicates that the k-means solutions had clusters interwoven all across the structure without respecting the natural grouping of the UMAP projection. In practical terms, what we'd like to see are the colors indicated by the k-means solution wrapping around the natural structures in the 2D projection.

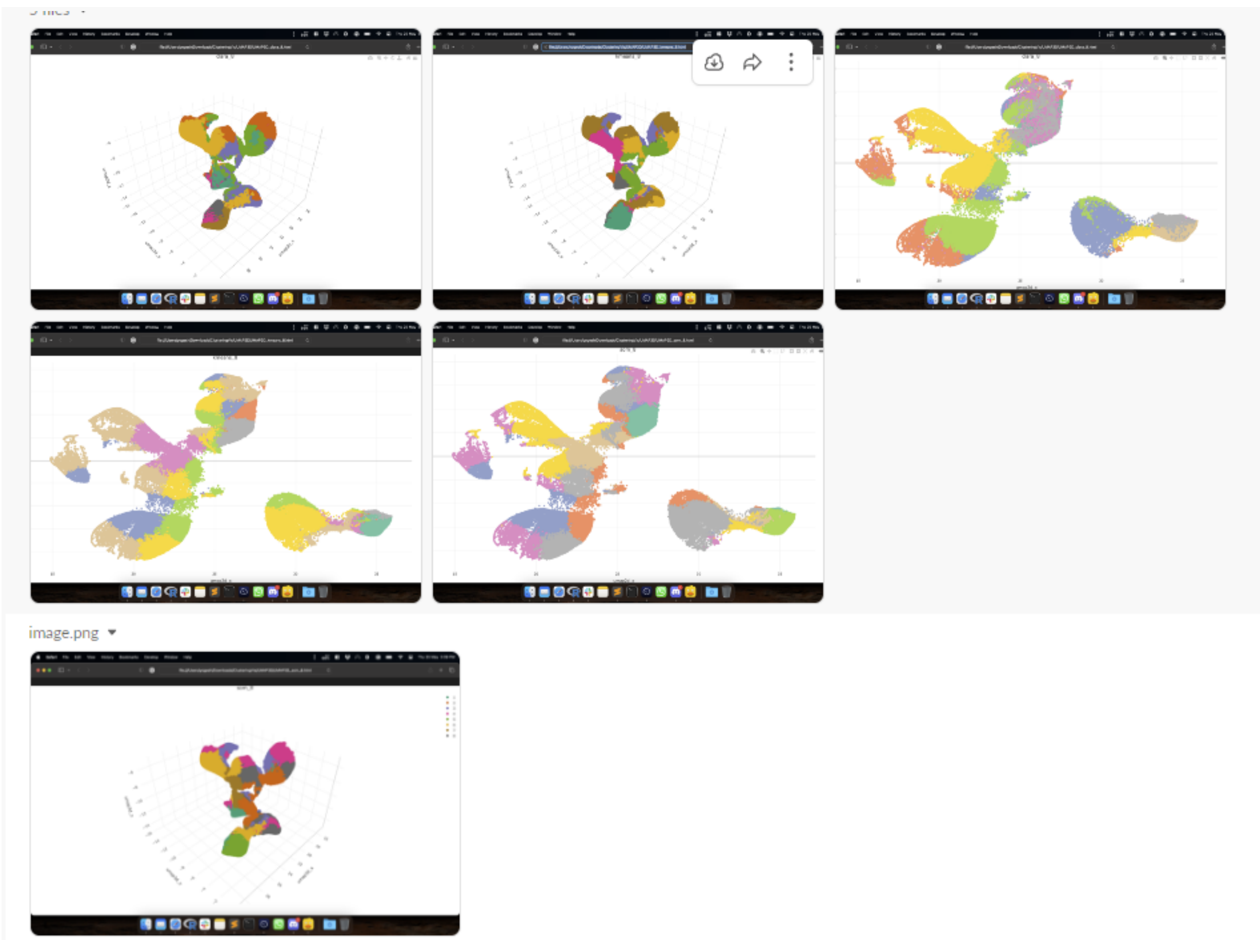

Figure 14: Example of the incompatibility between a distance based k-means solution and UMAP's manifold learning. Notice the lack of separation between the colors along the different "structures" within the data.

The lack of alignment suggests that an alternative approach may be needed, one that identifies groupings based on their arrangement in the 2D UMAP space. To do so, we employed a manual brushing technique that consists of using multiple projections of the UMAP derived coordinates, in both 2D and 3D space, to select groups of wallet addresses simultaneously. The selection was then color coded and given a cluster assignment. These assignments were then appended back to the dataset and investigated for similarities, contrasts, outliers and other distinctive features within an interactive table that tracked grouped means. This semi-automated process was then repeated leveraging human judgment to merge clusters with similar profiles into the final 13 cluster solution to be shared in the following results section.

In more cookbook fashion, the procedure for manually brushing was as follows:

Step 1. Plot the 3-Dimensional coordinate system of the UMAP projection and investigate the feature space visually.

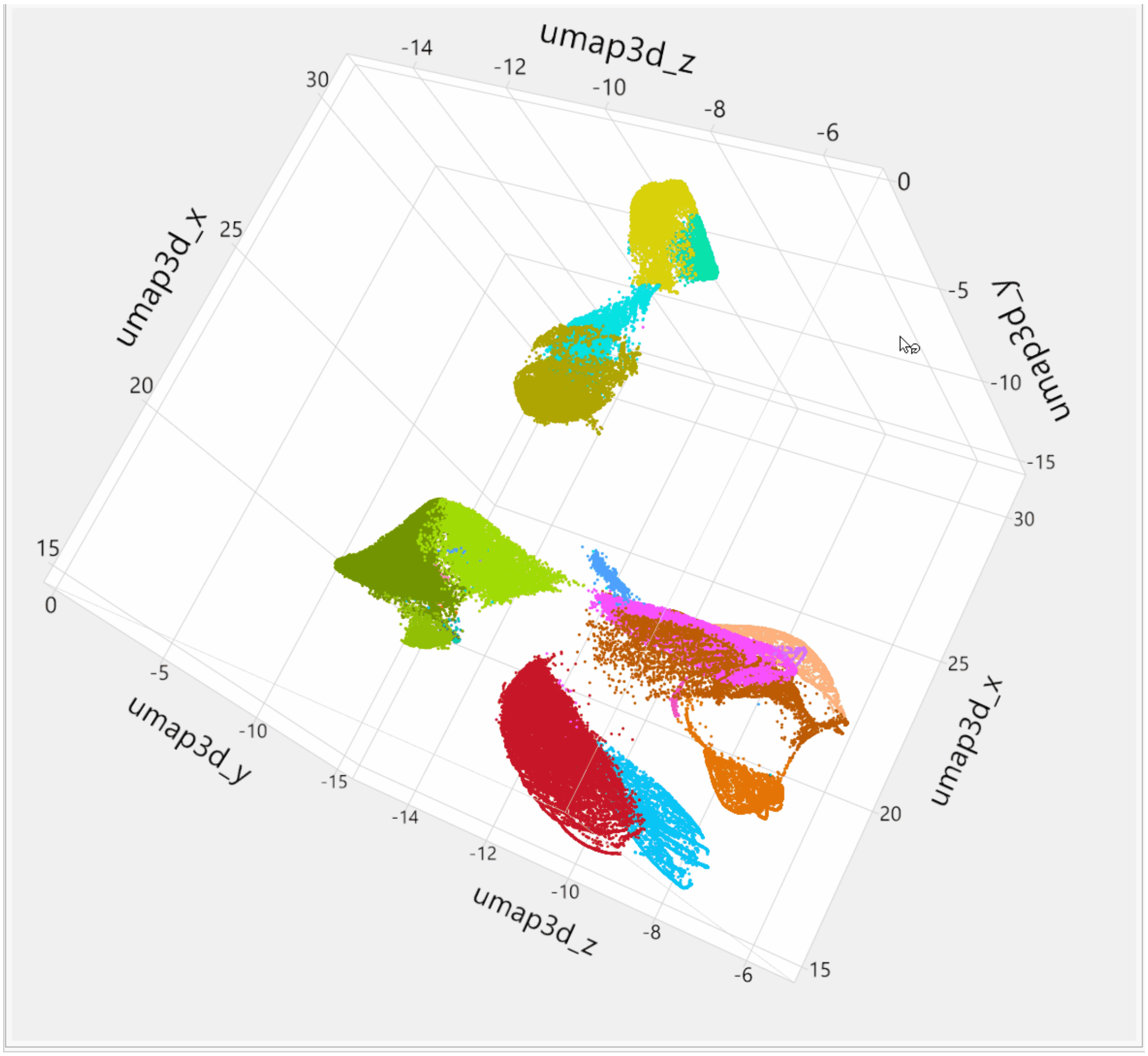

Figure 15: An alternative 3D view of the wallet addresses within the UMAP derived feature space. Notice that the distance between specific clusters becomes more pronounced at various perspectives.

Step 2. Plot multiple 2D projections of the UMAP space.

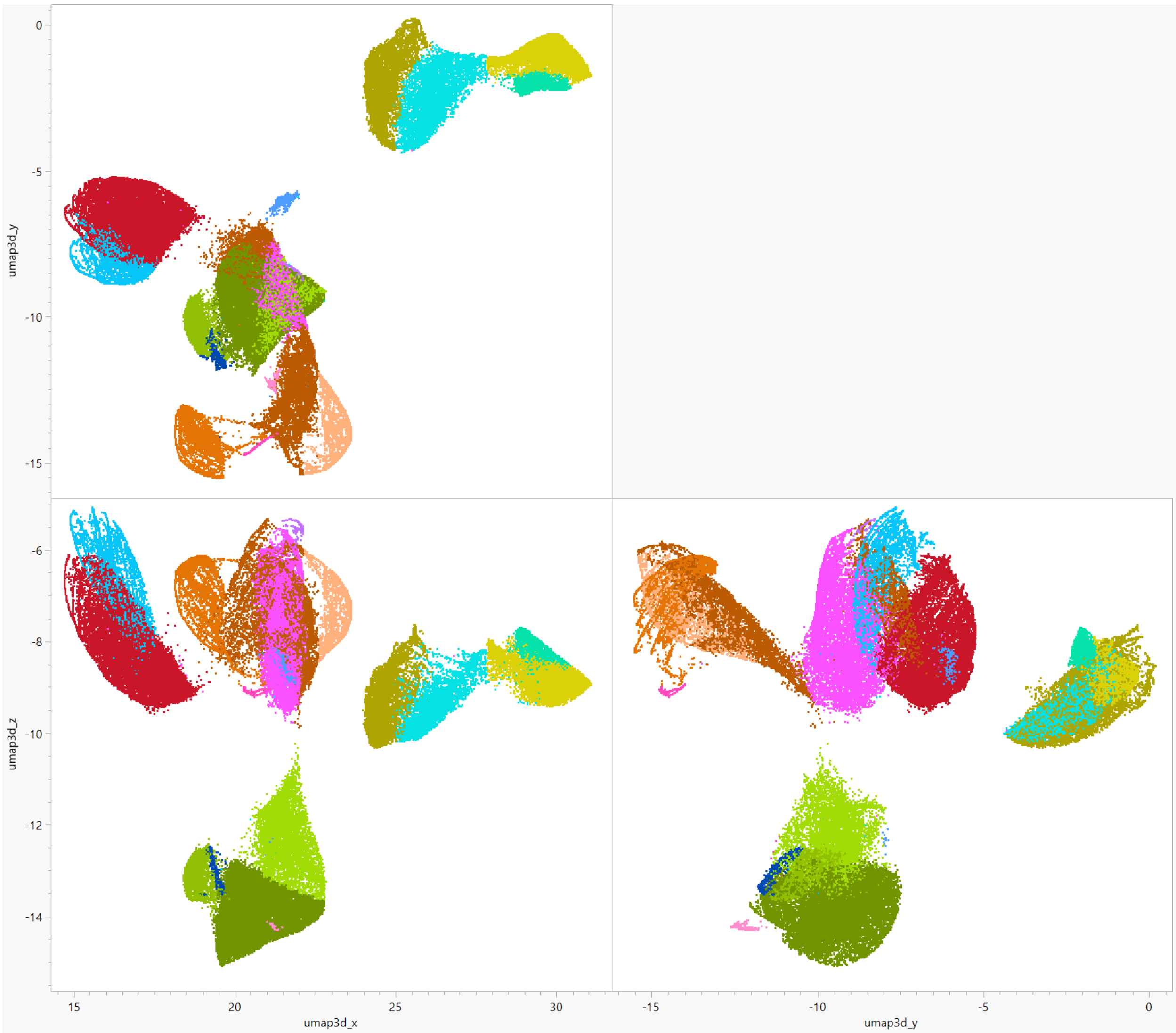

Figure 16: Multiple 2D projections of the UMAP solution used to manually select specific regions of the graph and those corresponding wallets.

Step 3. Link the graphs with an interactive table that tracks the cluster averages.

		OmniUMAP								
		6	9	12	19	22	25	28	37	39
num_chains_active	Mean	1.0001083306	3.4260262258	4.4912280702	5.777823169	2.6291020365	9.1807228916	5.0169079516	10.962562396	3.3687607573
num_tokens_eth	Mean	2.0892644351	6.0453249715	5.230387289	9.465013131	4.0683760684	6.4728915663	7.7578327571	17.496672213	6.3700516351
bal_usd_eth	Mean	25261.719419	74507.171695	37586.198069	40563.290721	25687.067471	283.42226454	45412.981217	2980.2797141	6611.8717602
aave_asset_usd_eth	Mean	0.8473344166	528765.3369	0.3246613704	0.1697130435	72554.513268	1.8810773785	1.139809745	16.497499168	2021.8195693
aave_debt_usd_eth	Mean	0.0012901094	233845.69232	0.0182601787	0.0020355413	0.0637357814	0.5904722684	0.1553795916	0.5293427621	941.52039415
CE_ethereum_Borrow_calls	Mean	0.0012999675	6.9680729761	1.9321416749	0.0025094835	0.0262741374	1.031262983	6.6136019879	0.2179700499	8.2267641997
CE_ethereum_Deposit_calls	Mean	1.6758747698	5.9017958951	0.0215160543	1.8320396849	1.5790862087	1.0628375571	6.9345289542	7.7911813644	5.7728055077
CE_ethereum_FlashLoan_calls	Mean	0.0111580544	0.7373147092	0.3174445548	0.008578932	0.0097077134	0.0001038637	0.6166810717	0.0008319468	1.2082616179
CE_ethereum_LiquidationCall_calls	Mean	0.0003249919	0.268386545	0.0642171466	0.0008170411	0.0013717421	0.0018695472	0.4244814175	0	7.571858864
CE_ethereum_Repay_calls	Mean	0.0146246344	4.7558437856	1.8828202582	0.0387510943	0.0167774612	0.021603656	5.6273228176	0.0291181364	3.0753012048
CE_ethereum_Withdraw_calls	Mean	1.4741631459	2.8021664766	0.0390599139	1.7168368836	0.4727234357	0.0165143332	5.0192307692	0.0424292845	1.7594664372
N		9231	7016	3021	17135	9477	9628	18512	1202	2324

		OmniUMAP								
		40	42	43	44	45	46	53	70	78
num_chains_active	Mean	2.4119318182	9.3173453123	4.2563875786	5.8509079811	2.0239529547	1	1.0398505604	1	0
num_tokens_eth	Mean	0	8.9113172004	8.5753734277	12.128643369	3.0556511761	0	1.8356164384	1.005409304	0
bal_usd_eth	Mean	0	2908.3042299	965264.12124	159242.1815	17280.292715	0	25097.357886	71.348640845	0
aave_asset_usd_eth	Mean	63.566417614	42.946320243	24767.369726	64404.148974	0.1328066552	160.47364691	0.1374694894	397.92230472	0
aave_debt_usd_eth	Mean	0	15.830370091	0.0467963836	2.469214253	0.0013304647	0.0115953608	0.0165716065	0.0076696718	0
CE_ethereum_Borrow_calls	Mean	0	1.0641935951	0.0045204403	7.1706088814	0.0014343087	0.0154639175	1.2826899128	0.0036062027	0.4968492487
CE_ethereum_Deposit_calls	Mean	1.2528409091	1.1407042458	1.8701847484	6.8173355715	1.6936316695	1.1610824742	0.0062266501	1.1831950956	1.7285506544
CE_ethereum_FlashLoan_calls	Mean	0.0056818182	0.0031879438	0.0119889937	0.6003357241	0.0159208262	0.0012886598	0.2540473225	0.0068517851	0.1342704799
CE_ethereum_LiquidationCall_calls	Mean	0	0.0134763078	0.0003930818	0.4181291012	0.0001434309	0	0.0361145704	0	0.0489578284
CE_ethereum_Repay_calls	Mean	0.0085227273	0.0427474279	0.014740566	6.8683045933	0.0240963855	0.0077319588	1.5417185554	0.0036062027	0.4847309743
CE_ethereum_Withdraw_calls	Mean	0.1079545455	0.0685407912	0.7468553459	4.9389592553	1.5149168101	0	0.0174346202	0.0003606203	1.4648570044
N		352	6901	10176	6553	6972	776	803	2773	2063

Figure 17: An interactive table of cluster means that is automatically updated when colored regions are added, removed or modified in the 2D map.

Step 4. Investigate pairs of clusters for cohesion, compactness and quantitative distinctness.

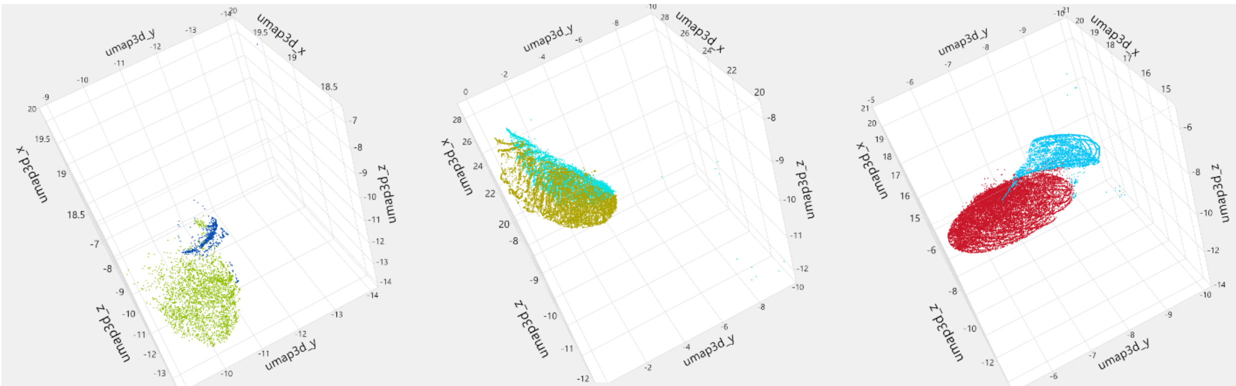

Figure 18: Sample visualizations of pairwise comparisons done to assess whether groups should be merged or continue to exist separated from one another.

Step 5. Review the final cluster solution in 2D space.

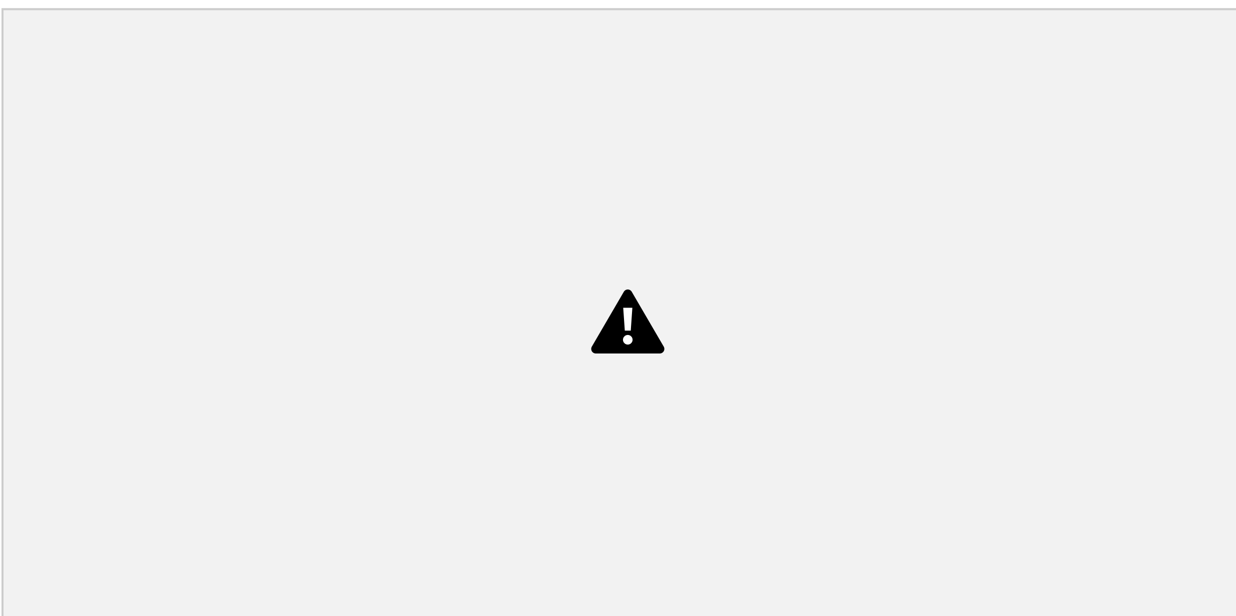

Figure 19: An animation showing the individual groups after the manual brushing process has been completed.

Aave User Segmentation Results

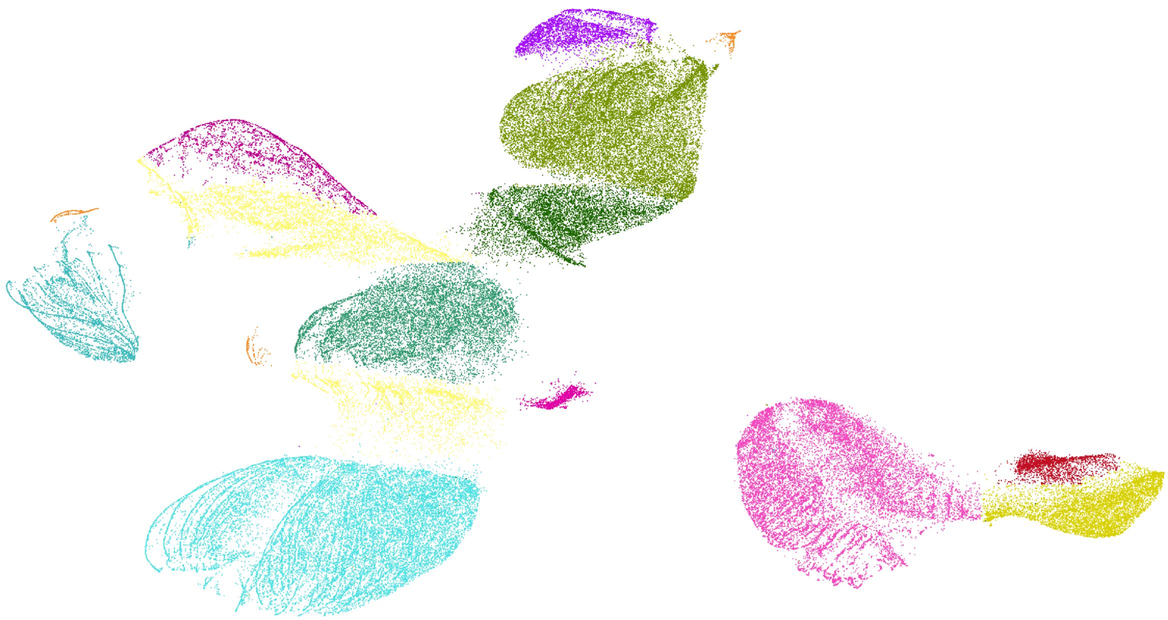

Figure 20: The final clustering solution after manually color brushing groups that were "interesting", relatively distinct and reasonably partitionable.

Utilizing a manual brushing technique within an interactive graphical framework to color code partitions inside of a 3-Dimensional non-linear feature space yielded a final segmentation solution consisting of 13 distinct groupings of wallet addresses. At the highest level, these users align themselves according to their wallet balances or income, how often they interact with the Aave contract, whether they use debt and if they possess a special combination of characteristics or behaviors that are distinctly unique. To develop this taxonomy, each profile was assessed quantitatively through the use of its mean vector, which is simply a collection of that groups average along each feature used in the segmentation. Started alternatively, a mean vector is equivalent to a row in a spreadsheet that corresponds to a single group and the individual columns are the group means for each variable. When concatenated on top of one another and color coded, it becomes relatively straight forward to identify unique features of a single profile, as well as, categorize the similarities and contrasts between them.

FinalClusters	Counts	Proportions	num_chains_active	num_tokens_eth	bal_usd_eth	aave_asset_usd_eth	aave_debt_usd_eth	CE_ethereum_Borrow_calls	CE_ethereum_Deposit_calls	CE_ethereum_FlashLoan_calls	CE_ethereum_LiquidationCall_calls	CE_ethereum_Repay_calls	CE_ethereum_Withdraw_calls
ETH Small Frys Depositooors	2773	2.41%	1.000	1.005	$71.35	$397.92	$0.01	0.004	1.183	0.007	0.000000	0.004	0.0004
ETH Testers	9231	8.03%	1.000	2.089	$25,261.72	$0.85	$0.00	0.001	1.676	0.011	0.000325	0.015	1.4742
High Rollers AD	6553	5.70%	5.851	12.129	$159,242.18	$64,404.15	$2.47	7.171	6.817	0.600	0.418129	6.868	4.9390
High Rollers AD w/Debt	7016	6.11%	3.426	6.045	$74,507.17	$528,765.34	$233,845.69	6.968	5.902	0.737	0.268387	4.756	2.8022
Middle-Class HC-LS	10174	8.85%	4.255	8.545	$67,959.50	$24,205.10	$0.05	0.005	1.869	0.012	0.000393	0.015	0.7441
Middle-Class LC-HS	9477	8.25%	2.629	4.068	$25,687.07	$72,554.51	$0.06	0.026	1.579	0.010	0.001372	0.017	0.4727
Multi-Chain Testers	24107	20.98%	4.692	7.611	$33,829.60	$0.16	$0.00	0.002	1.792	0.011	0.000622	0.035	1.6584
Potential Arbitragers	3824	3.33%	3.766	4.518	$34,963.67	$0.29	$0.02	1.796	0.018	0.304	0.058316	1.811	0.0345
Small Fry Degen Active Depositooors	1202	1.05%	10.963	17.497	$2,980.28	$16.50	$0.53	0.218	7.791	0.001	0.000000	0.029	0.0424
Small Fry Degen Debt Users	16529	14.38%	9.238	7.491	$1,379.33	$19.03	$6.95	1.045	1.095	0.001	0.006715	0.030	0.0382
The Good Guys	18512	16.11%	5.017	7.758	$45,412.98	$1.14	$0.16	6.614	6.935	0.617	0.424481	5.627	5.0192
The Liquidated	2324	2.02%	3.369	6.370	$6,611.87	$2,021.82	$941.52	8.227	5.773	1.208	7.571859	3.075	1.7595
Throw Away Accounts	3191	2.78%	0.509	0.000	$0.00	$46.04	$0.00	0.325	1.538	0.088	0.031652	0.316	0.9589

Figure 23: A quantitative representation of the group mean vectors also shown in the parallel coordinate plots. Values have been placed along a color gradient for easier examination.

Beginning at a bird's eye level, the table in Figure 23 is a quantitative representation of the wallet segment's mean vectors. Each row represents a different user profile, while each column is a separate characteristic of that group. The first two columns display the number of wallets within each specific profile and their proportion across the entire sample. For example, the user profile designated as "Multi-Chain Testers" has 24,107 wallets matching this profile and they make up 20.98% of the total addresses in the study. Making up only 1.05% of the total population of segmented wallets are the "Small Fry Degen Active Depositooors" with only 1,202 users exhibiting the behavior of this group. These names have all been generated from looking at the unique characteristics of the profile, particularly which combinations of variables does this group have high and low values for. A breakdown of the individual profiles is forthcoming in this document while in this section we focus solely on the interpretation of the mean vector table. The next set of columns correspond to our "demographic" variables and consist of features that track the number of chains a wallet has been active on, the number of tokens held, and the amount of Ethereum held in U.S. dollars. Similarly, the next two U.S. dollar denominated variables track the total assets deposited into Aave along with the total debt held by the wallet. For both these categories, the "High Roller AD w/Debt" profile has the highest average asset value and debt load at $528,242 and $233,845 respectfully. The last six columns of the table correspond to the average number of contract calls made of each type. Recall that the segmentation model includes tracking variables for the number of times a user borrowed funds, deposited funds, used a flash loan, repaid their loan, withdrew their funds and was liquidated. It is the patterns of high and low values amongst the users and between variables that are used to highlight the distinctions between the groups. For example, across all thirteen groups, twelve have an average number of liquidation calls less than 1, with two groups having no liquidations. This fact is contrasted with a group with an average number of calls exceeding 7 per wallet address. That suggests that members of this group are unique in that they've been liquidated at a frequency unlike any other cluster or profile. It is the distinctness of this trait that motivated the naming of this user profile as

"The Liquidated", one of the "Special Case" profiles that deserves its own categorization. As another example of interpreting the patterns within the mean vector table to extract meaningful insights about the profiles, consider the average number of deposits made by an account. One group had the highest mean number of deposited at 7.791. When looking across the other variable averages for this group, we see that these wallet addresses also stand out in the number of chains they are active on and the number of tokens held. Intuition and web3 experience clearly suggest that this group can be considered "degens", but what definitively sets them apart is the fact they also hold some of the lowest balances on or off the platform. When we aggregate all of these facts, it motivated the naming this profile "Small Fry Degen Active Depositooors" to categorize their full behavior. It is this process of traversing the mean vector data frame for patterns that inspired the naming of the profiles.

The task of making high level comparisons while also contrasting the differences between groups can be done more effectively using scale invariant visualization techniques specifically designed for multi-dimensional data. One such technique is the parallel coordinate plot, a visualization and descriptive tool that graphs mean vectors along a single dotted line plot where the highs and lows between groups can be more easily pinpointed. In Figure 22 we have a parallel coordinate plot visualization of the mean vectors for the 13 Aave user wallet profiles previously detected from our segmentation routine. Along the tick marks on the x-axis are indicators for the individual variables that comprise the mean vector. On the y-axis are the normalized values of each variable ranging from 0 to 1 through univariate scaling. Above each graph is the title indicating the name of the group. Within each plot is a dotted line graph where each point corresponds to the scaled value of that group for a specific variable. With this context, when analyzing the individual line graphs, the patterns become more pronounced. It is also immediately apparent that each of the 13 profiles are unique, with high-low patterns across differing variables.

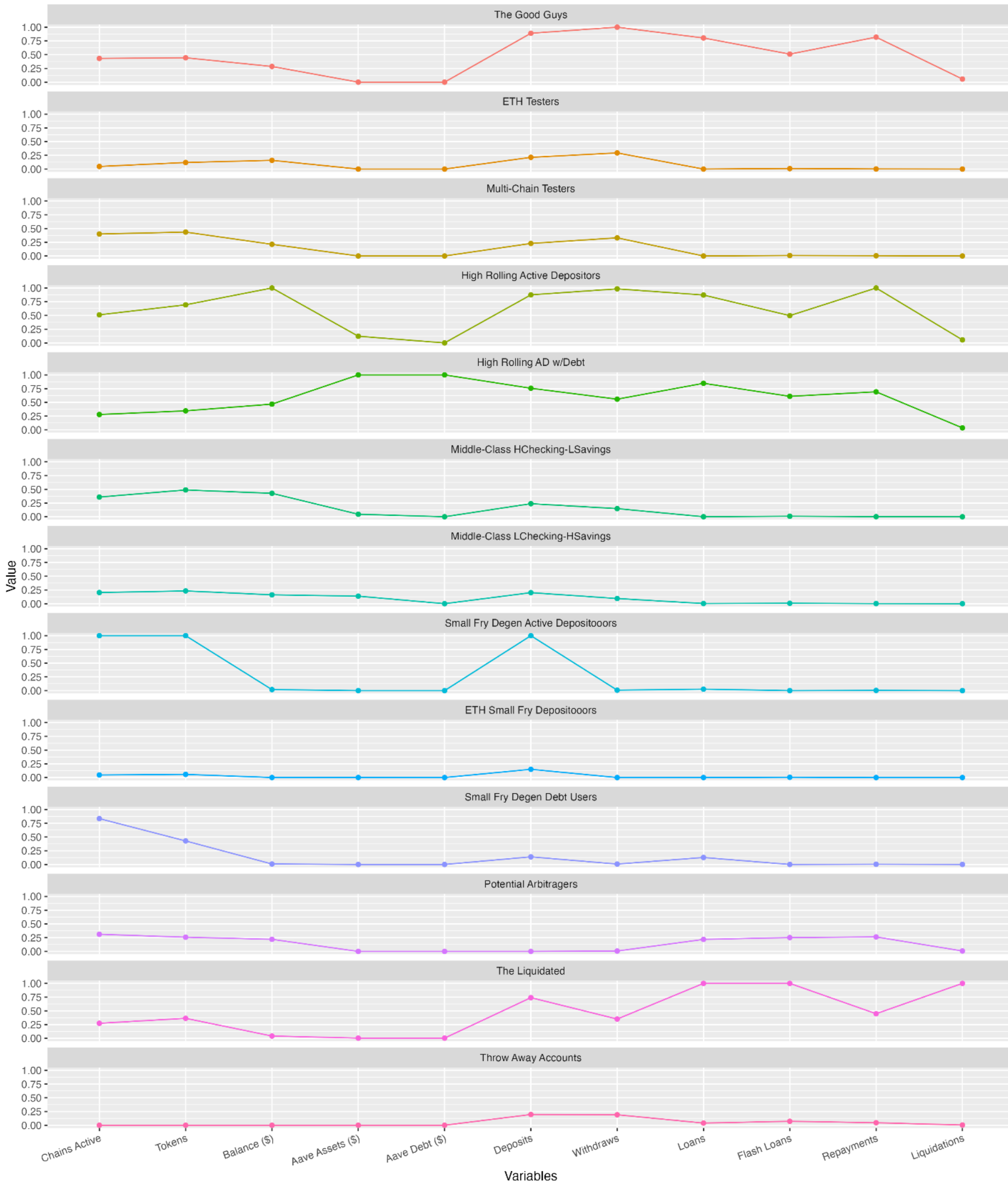

Figure 22: Parallel coordinate plots are the standard way user segments are visualized in multiple dimensions. The x-axis corresponds to a set of variables whose values are shown on the y-axis. The profile name in the title references the group.

In the following section we'll give an explanation of the individual profiles and provide intuition behind their naming, as well as, their relation to the other customer segments.

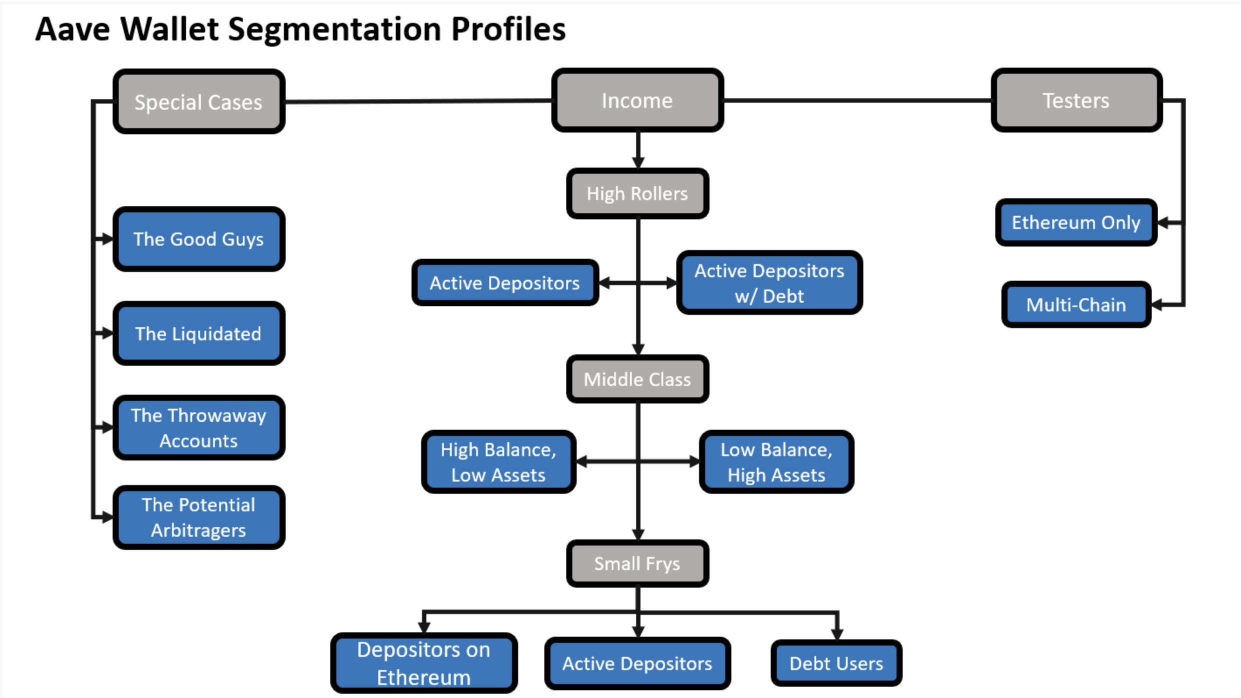

Figure 24: High level break down of one such categorization of the Aave wallet profiles.

ETH Small Fry Depositooors - These "small fries" are trying their best to survive in Ethereum's high gas fee environment. They are active exclusively on Ethereum, hold only Ethereum, deposit once and often just let their money sit on the platform. The value of their deposits is often higher than their wallet balance, which suggests they are just trying to save and hope for the best.

ETH Testers - Testers are those who use the platform once or twice then leave. They usually have one or two deposits that they subsequently withdraw. They usually don't borrow and have only tried the platform on ETH, as indicated by their low "Chains Active" value.

High Rollers AD - These “high rolling” accounts possess a six figure wallet balance, on average, and have assets deposited in Aave. These accounts regularly make deposits and withdraws as they manage and repay their debts.

High Rollers AD w/Debt - These “high rolling” accounts possess a six figure Aave deposit value, on average, all while still maintaining a meaningful wallet balance. These accounts regularly make deposits, but perform fewer withdraws and choose to keep some outstanding debt that they often repay. Interestingly enough, those who regularly keep debt in this cluster are liquidated less frequently than the other high rolling profile that chooses to not use debt.

Middle-Class High Checking – Low Savings (HC-LS) - This cluster of users keep a high wallet balance (checking), but a lower deposit balance (savings). Interestingly, neither of the middle class profiles use debt.

Middle-Class Low Checking – High Savings (LC-HS) - This cluster of users keep a low wallet balance (checking), but a higher deposit balance (savings). Interestingly, neither of the middle class profiles use debt.

Multi-Chain Testers - These wallets are active on more than one chain, yet have interacted with the ETH Aave contract in the same way as the other testers by refraining from borrowing while making and withdrawing only one or two deposits.

Potential Arbitragers - These savvy accounts are rarely liquidated but almost exclusively use flashloans and borrow without ever depositing. Despite that, they repay on time, then withdraw to keep their assets left on the platform very low.

Small Fry Degen Active Depositooors - This group of small fry accounts are active on the greatest number of chains and hold a large number of tokens in addition to making the largest number of deposit calls to the Aave ethereum contracts. They rarely withdraw, but also don't have large deposits. Airdrop farmers?

Small Fry Degen Debt Users - These “small fries” deposit, but also try out the platform's borrowing functionality with a small amount of funds that they keep outstanding.

The Good Guys - The credit industry maliciously calls these users "deadbeats" but they are shining examples of users that make deposits, borrow money, repay their loans and withdrawn when necessary. They are rarely liquidated or call for liquidations nor keep a large balance in their wallets.

The Liquidated - These are the active accounts that Supply, Lend, Borrow, use Flash loans, and generally do everything on the platform. Unfortunately they pay the cost for their experimentation because they are liquidated at a higher rate than any other profile.

Throw Away Accounts - Similar to the "tester" profile, these are accounts that were used to interact with the contract once, and then were drained or discarded. Despite having a positive asset balance on average, most accounts in this category have $0 on the platform.

Lastly, to further understand the differences in the groups, we created a table of proportions, counts and values comparing the profiles against a set of exogenous variables collected that were not included in the segmentation analysis. These represent how active the different profiles are across various platforms, how committed to Web3 governance and also how long they’ve been active on the Ethereum blockchain. Lens is a social media platform, currently invite only, that allows users to participate in a Web3 designed Twitter like experience. At the time of this writing, a substantial percentage of both “Small Fry” profiles had Lens profiles with the third highest group being the “High Rollers AD” with only 12.6% of their members having a profile. We have a similar result for the proof of personhood protocol, Gitcoin Passport where the same set of profiles have the most interaction with the platform. Curiously, these same set of profiles have the highest proportion of their members flagged by the Trustalabs anti-sybil detection platform, though the numbers are much lower as a percentage of the population. Here, at the highest, 24% of the “Small Fry Degen Debt Users” have been flagged as having been involved in sybil-like behavior. This shouldn't be too surprising because their profiles are similar to those of airdrop farmers who spread small amounts of activity across many different protocols and chains, where some of which do so in an automated manner. Activity within the governance platform, Snapshot, tells a similar story where the same three profiles are again the most active. It is only when we look at the last variable, the age of the wallet, do we see a break in this pattern. The "age" of a wallet in this context has been taken to be a measurement of time from its first transaction until this writing. Outliers aside, the "High Rollers AD" are the oldest group with the average age of a wallet in this group being approximately 2 years and 10 months old. The next oldest group is only a month younger and consists of the "Good Guys", the group who regularly borrow and pay back their loans. An interesting fact happens to be that the youngest profile actually are the hyper active "Small Fry Degen Debt Users", who are active across multiple chains, try out all the functionality of the platform all while doing so with limited resources. They are on average have only been around since the 2021, being approximately 1 year and 10 months old.

cluster	LensProfile	GitcoinPassport	TrustaLabSuspects	AtLeast1SnapshotVote	MeanSnapshotVotes	Mean(trustalabs_score)	Mean(agedays)
Binance Exchange	0.00%	0.00%	0.00%	0.00%	•	•	1414.4704256
Contract	0.00%	0.00%	0.42%	0.13%	5.2777777778	80.626968966	827.33285843
ETH Small Frys Depositooors	0.00%	0.00%	0.58%	0.04%	1	69.6875	731.84101404
ETH Testers	0.01%	0.00%	0.77%	0.03%	1	64.594360302	882.89772745
High Rollers AD	12.60%	6.61%	7.00%	7.22%	25.344608879	73.809582506	1036.7058826
High Rollers AD w/Debt	3.63%	1.23%	3.42%	2.01%	12.156028369	70.815874249	762.3040087
Maker Vault Owner	0.00%	0.00%	0.00%	0.00%	•	•	1281.449056
Middle-Class HC-LS	6.03%	3.01%	2.51%	3.81%	20.481958763	72.14735525	946.21552585
Middle-Class LC-HS	3.96%	4.69%	2.15%	2.81%	25.387218045	75.829781153	833.11375394
Multi-Chain Testers	4.65%	2.64%	2.98%	1.07%	20.953488372	72.899043721	1042.26986
MultiSig	0.00%	0.00%	0.26%	1.03%	5.75	79.5	557.34036914
Potential Arbitragers	2.85%	1.52%	3.43%	0.52%	16.95	78.591147548	799.98477062
Small Fry Degen Active Depositoors	55.82%	63.73%	19.80%	39.35%	37.179704017	75.537416926	816.97390059
Small Fry Degen Debt Users	58.75%	44.26%	23.72%	34.37%	34.417004049	81.114985828	683.94482931
The Good Guys	8.26%	5.44%	5.01%	3.21%	24.695798319	73.845126646	1000.1433297
The Liquidated	2.97%	1.03%	3.06%	1.08%	6.36	69.71455432	932.70157078
Throw Away Accounts	0.03%	0.00%	0.72%	0.00%	•	69.89828947	876.02434053

Figure 25: The cells within this table correspond to a numerical value for a specific variable or its proportion within that group. For example, 55.8% of "Small Fry DADs" had a Lens profile.

Conclusion

We embarked on this project to enhance Aave's understanding of their current and future user base by performing a wallet segmentation analysis that unearths insights into how groups of users borrow, lend, and interact with the Aave platform. Our three pronged goal was to provide insight into Aave platform usage and group behavior so that the Aave team could use these results to guide and direct marketing and product development while ensuring the results were externally reproducible. To that final point, in conjunction with this report a Github repository with the data and this report have been hosted online through the FDD Github. With these results we believe there are further opportunities to operationalize this research by:

- Creating targeted marketing campaigns that offer a customized UI/UX for wallet addresses of each personalized profile.
- Developing products and protocol features that align with the demand generated by each group.

From a research perspective, this work could be extended by:

- Including additional EVM compatible chains into the analysis.
- Performing more advanced exploratory analysis that assess the time to specific events such as the duration between deposits and withdraws, the time between funds being borrowed and repaid, or the average length of a loan before it results in a liquidation.
- Additional data could be collected on the customer path which could then be constructed into separate customer journeys for each of the 13 profiles.
- The interconnection between other Web3 platforms and Aave could be constructed from analyzing the number of users who've made contract calls to other platforms along with Aave. This would inform the Aave team of other communities that would be open to potential partnerships.

- From a technical perspective, Neural Network inspired Autoencoders could be used to offer an alternative approach to the dimensional reduction, possibly lending itself to additional new, unforeseen clusters.

Resources

1. https://www.aaveql.org/
2. https://docs.aave.com/developers/v/1.0/developing-on-aave/the-protocol/lendingpool
3. https://docs.aave.com/developers/v/2.0/the-core-protocol/lendingpool
4. https://docs.aave.com/developers/deployed-contracts/v3-mainnet/ethereum-mainnet
5. www.aaveql.com
6. Contracts of interest on Ethereum:
    a. https://etherscan.io/address/0x398eC7346DcD622eDc5ae82352F02bE94C62d119
    b. https://etherscan.io/address/0x7d2768de32b0b80b7a3454c06bdac94a69ddc7a9
    c. https://etherscan.io/address/0x87870Bca3F3fD6335C3F4ce8392D69350B4fA4E2
7. https://stats.stackexchange.com/questions/263539/clustering-on-the-output-of-t-sne

Appendix of Figures

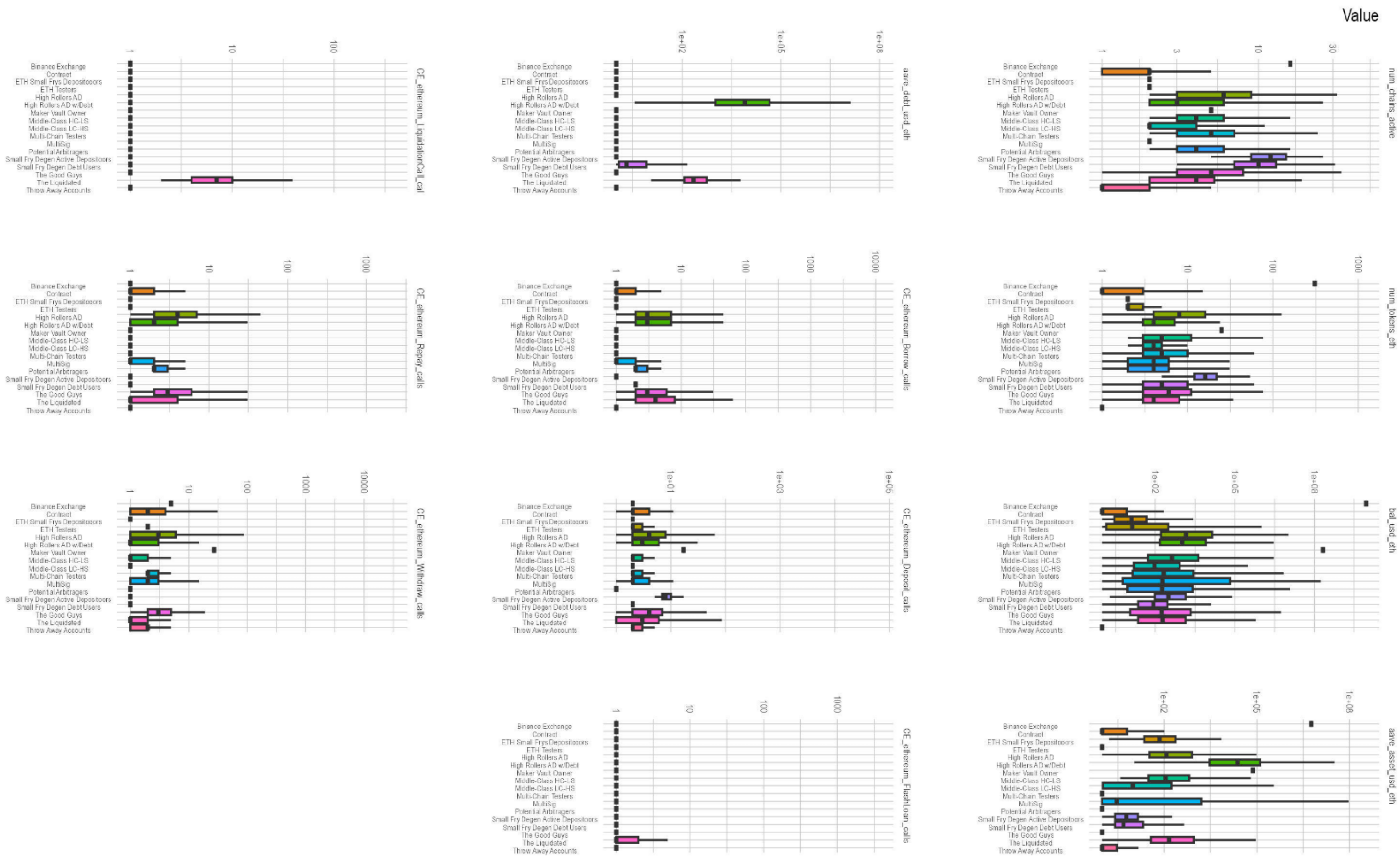

Figure 21: A stylized set of boxplots designed to highlight the profile differences along each feature in a univariate manner. Notice that Multi-Sigs, Contracts and both outliers have been included.