Solana Swap Transaction Analysis – Market Maker vs Organic Trader

Project Description and Objectives

This project involves analyzing **Solana blockchain swap transactions** to distinguish between activity driven by an automated *market maker* and that of an *organic trader*. In decentralized exchanges like **Raydium** (a Solana-based DEX and automated market maker platform), market makers provide liquidity and execute frequent trades, while organic traders are regular users trading sporadically for investment or speculation. The analysis focuses on parsing on-chain swap data, **splitting transactions into decision blocks**, and applying classification rules to identify whether a given sequence of swaps is likely from a market-making bot or an organic user. The main objectives are to:

- **Segment transaction history into "decision blocks"** – contiguous time intervals or event-based segments where trading behavior can be analyzed in isolation.
- **Classify each block's transactions** as being driven by a market maker vs. an organic trader, based on patterns and predefined rules.
- **Develop a rule-based system (up to 20 rules)** capturing telltale indicators of market maker behavior, with an eye towards later optimization via machine learning.
- **Demonstrate key metrics** (trade frequency, sizing, timing, etc.) that differentiate the two trader categories, using data visualizations and summary tables.
- **Provide a foundation for future ML integration**, where the rules and data can feed into a model to improve classification accuracy and maximize a target "pressure" metric (explained later).

Overall, the project aims to deliver a **technical report suitable for an external sponsor** (Bogdan, AlphaCube). This report documents the methodology, technical implementation, findings, and recommendations. It provides transparency into how the team approached the problem of identifying market maker activity on Solana and the results of that analysis.

Team Organization and Roles

The project was executed by a small focused team, each member taking on specific responsibilities to meet the sponsor's requirements:

- **Rish Kumar (ML Engineer)** – Led the **transaction segmentation and rule development**. My primary focus was designing how to split the transaction timeline into decision blocks and formulating classification rules based on on-chain features. Additionally, I spearheaded the planning for **machine learning integration**, ensuring that the rule-based framework could be enhanced with ML for better performance.
- **Caleb (Data Engineer)** – Handled **data mining and preparation**. Caleb ensured data quality and completeness, and assisted in feature engineering (e.g. computing balances, labeling transaction types) within the dataset.

This clear division of labor allowed the project to move quickly: Caleb's data groundwork fed directly into my analysis logic. Regular check-ins were held to discuss emerging patterns in the data and to refine the rules. The sponsor (Bogdan) was kept informed through interim presentations, focusing on how the evolving classification could help them measure market-making activity on their platform.

Problem Statement

How can we accurately distinguish "market maker" wallets from "organic trader" wallets using only on-chain swap transaction data? This is a non-trivial challenge because on-chain data is pseudonymous and behavioral patterns can overlap:

- **Market makers** are specialized participants (often bots or firms) that provide liquidity by continuously buying and selling. They typically trade at high frequency, in large volumes or many small incremental volumes, often aiming for profit from bid-ask spreads or arbitrage. On centralized markets, market makers post bids and asks; on an AMM like Raydium, a market maker might instead repeatedly swap to balance prices or inventory. They may also add/remove liquidity to pools. Their behavior tends to be **algorithmic and reactive**, possibly involving multiple quick trades around the same time. Market makers often execute *thousands of trades a day* as part of their strategy, maintaining an active presence almost continuously.
- **Organic traders** (retail or typical users) trade for personal portfolio reasons – e.g. buying a token to hold, or swapping when prices move. Their activity is **sporadic and driven by external decisions**, not continuous market maintenance. An organic trader might make a few trades on a volatile day, then none for weeks. They are more likely to use standard interfaces (web UI or

simple wallet apps) and **don't usually employ sophisticated techniques** like priority fees or block-order optimizations.

The difficulty lies in the gray areas: a skilled individual trader using advanced tools might resemble a bot, or a market maker during low activity might not trade often. No direct label on-chain marks a wallet as "market maker" or not – we must infer it from indirect clues (transaction frequency, patterns, tool usage, etc.). The sponsor's concern is that **misclassification could distort metrics**; for example, if market-maker-driven volume is mistaken for organic user demand, one might overestimate genuine interest in a token. Therefore, the classification system must be **highly accurate** and based on defensible rules or model predictions.

Sponsor's Requirements and Task Request

The sponsor has formally requested the development of a system with the following capabilities:

1. **Transaction Splitting:** Take the comprehensive list of swap transactions of interest and **partition them into distinct "decision blocks."** A decision block is essentially a segment of the transaction sequence, delineated by certain block heights or timestamps, intended to isolate periods of differing trading behavior or context. (For example, these could correspond to phases like initial token launch trading vs. later normal trading, or periods when a market maker's strategy changed.)
2. **Classification Rules:** Formulate a concise set of up to **20 rules** that can be applied to each decision block (or wallet) to classify the dominant trader type in that segment as "Market Maker" or "Organic Trader." These rules should be rooted in observable features (timing, size, transaction type, etc.) and ideally be *human-readable* (for transparency to the sponsor).
3. **Data & Feature Pipeline:** Build the underlying data processing pipeline to support the above – meaning reading raw transaction data, computing necessary attributes, and outputting structured data (like CSV files) per decision block. This includes tracking relevant features for each transaction (e.g. whether it was a Raydium swap, a liquidity addition, the SOL balance changes, etc.).
4. **Machine Learning Integration:** Lay groundwork for an optional **machine learning workflow** that can take the rule-based classifications or underlying features and optimize the classification. The sponsor specifically mentioned a "target pressure metric" – likely a custom metric they use to quantify the impact of market making on the market (for instance, net order flow or price pressure exerted by trades). The system should allow experimenting with ML to maximize this metric, meaning the rules could be tuned or augmented by a model to best capture whatever "pressure" the sponsor is measuring.

5. **Reporting and Visualization:** Provide a detailed technical report (this document) that describes the approach and includes illustrative **tables and charts**. The sponsor expects to see evidence of how the rules were derived and how well they perform (at least in exploratory analysis). This includes showing key differences in trading patterns via **CSV tables** (e.g. sample data segments) and **PNG charts** (e.g. distribution of transaction intervals, trade sizes, etc.), as well as summary statistics.

In summary, the sponsor essentially wants a prototype system that can take Solana swap data and **separate the wheat from the chaff** – identifying which swaps are likely from their market maker (or other professional liquidity providers) vs. which are from organic users. This will help them quantify organic trading interest and possibly the efficacy of their market maker in supporting the token's liquidity. The added ML component indicates they eventually want to **refine these rules automatically**, but the immediate deliverable is the rule-based classification and analysis of results.

Methodology and Steps Taken

To address the sponsor's requirements, the team followed a structured approach:

1. Data Acquisition: The sponsor gave us the raw transaction data for the analysis. The scope of data was all swap-related transactions for a particular token or account of interest on Solana, covering a certain time range. Using a Solana RPC API or an indexing service, he pulled all transactions involving Raydium swaps (and potentially liquidity pool interactions) for the relevant wallet addresses. Each transaction's details were stored in a JSON format for parsing. In total, 8 JSON files were prepared (one per address or per subset of data as needed), each containing an array of transaction records.

2. Data Parsing and Cleaning: A custom Java service, **SolanaAccountAnalyzerService**, was developed to parse these JSON files and extract structured information. This service utilized the Jackson JSON library to map transaction JSON into Java objects. Each transaction was represented by a `TransactionInfo` object containing fields such as:

- Transaction identifier (signature), block number, timestamp (block time), and status.
- **SOL balance changes** (`solBefore`, `solAfter`, `solChange`): how the wallet's SOL balance changed due to the transaction (e.g. spending SOL to buy another token results in a negative `solChange`).
- **Categorical flags** indicating the type of transaction. For each transaction, the code checks and marks booleans like `isRaydiumSwap`, `isRaydiumAddLiquidity`,

`isRaydiumRemoveLiquidity`, and others for special conditions. In fact, the analyzer tracks a wide range of flags, including:

- *Raydium Swap events:* `IsRaydiumSwap` (true if the transaction was a swap on Raydium AMM), with sub-flags `isRaydiumWebSwap` and `isRaydiumSDKUsed` to denote whether the Raydium swap was likely done via the web interface or via a programmatic SDK call.
- *Raydium Liquidity events:* `IsRaydiumAddLiquidity` / `IsRaydiumRemoveLiquidity` (true if liquidity was added to or removed from a Raydium pool), and `isRaydiumLiquidityPoolCreation` (if a new liquidity pool was created – usually happens once for a token launch).
- *MEV / Advanced flags:* `isJitoTip` (indicates a Jito transaction tip was used for priority inclusion), `isBloxroute` (if the transaction likely went through the bloXroute Solana gateway), `isSoltradingbot8` (possibly a specific bot identifier), `isOtherCanAnalyze` (generic flag for transactions that can be analyzed by generic logic), `isBananaGunSwap` (if the Banana Gun sniper bot was used for the swap), `isPumpFun` and related flags (such as `isPumpFunSwap`, `isPumpFunRaydiumMigration`, `isPumpFunMint`).
- *Token transfers:* `isInSolTransfer` / `isOutSolTransfer` (incoming/outgoing SOL transfers outside of swaps), `isOutTokenTransfer` (outgoing token transfer to another wallet).

- This comprehensive set of attributes allowed classification of each transaction's nature. As an example, a normal user swapping via Raydium's web UI would show `IsRaydiumSwap=true` and `isRaydiumWebSwap=true`, whereas a bot using the Raydium SDK or a aggregator protocol might have `IsRaydiumSwap=true` but `isRaydiumWebSwap=false` and instead `isRaydiumSDKUsed=true`.

The parser cleaned data by filtering out unsuccessful transactions (`status != "OK"` were skipped) to focus only on completed swaps. It also computed any other derived values needed (e.g. identifying distinct tokens involved in swaps and calculating token amount changes, though the core analysis here is on SOL changes and flags).

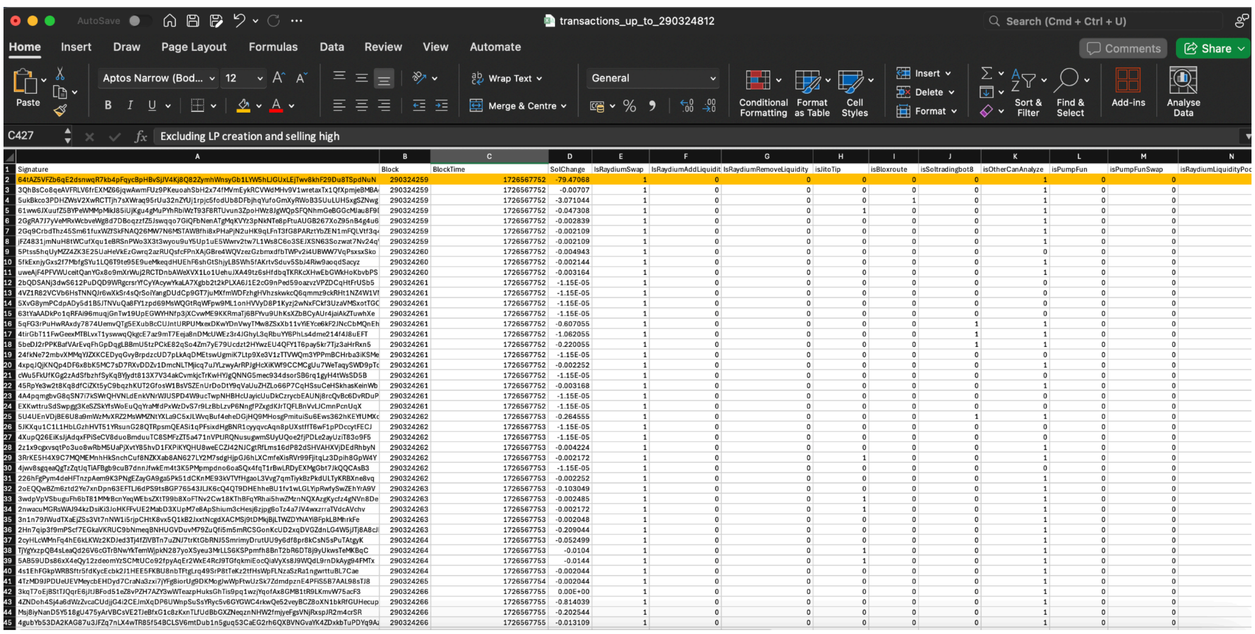
3. Sorting and Merging: Multiple JSON files were used (the data was split by multiple wallets/pool addresses), the service merged all transactions into one list `allTransactions` and then sorted them in chronological order (ascending by block time). This yielded a complete timeline of relevant transactions.

4. Decision Block Identification: Based on domain knowledge and patterns observed, the team defined specific block heights as boundaries for "decision blocks." These **decision block IDs** were likely chosen where there were notable changes in trading behavior or known events. For example, the very first swaps after token launch might be one block, a sudden influx of organic traders might mark another block, etc. In the code, a list of block IDs was hardcoded:

```
List<Long> decisionBlockIds = Arrays.asList( 290324812L, 290326724L,
290526624L, 290527489L, 290528391L, 290529580L, 290532030L, 290533481L );
```

These IDs split the timeline into segments. The service generated separate CSV files for each segment:

- One CSV for “all transactions up to block 290324812” (the first decision point).
- Then for each subsequent pair in the list, a CSV for transactions between those blocks (e.g. between 290324812 and 290326724, then 290326724 to 290526624, and so on).



All transactions up to block 290324812 csv snippet

Each CSV contains all transactions in that block range, with a header labeling the fields as described above. This effectively **splits the transaction history by decision period**, isolating, for instance, initial trading period, mid-day trading, end-of-day, etc., depending on what those block numbers correspond to in real time. The rationale is that the **market maker’s strategy or presence might differ in each segment**, and similarly organic trader participation might ramp up or down, so analyzing them separately prevents averaging out important patterns.

5. Feature Extraction per Block: With transactions now segmented, the next step was to derive **summary features for each block** that can feed into classification rules. This included:

- **Trade count and frequency:** e.g. number of swaps in the block, average time between swaps in that block, and perhaps distribution of those intervals.

- **Usage of advanced infrastructure:** metrics like “X% of swaps in this block used `JitoTip`” or “Y% used `RaydiumSDK` vs. web.” For example, if a block shows 0% web UI swaps and 100% programmatic (SDK or BananaGun) swaps, that leans towards algorithmic trading activity.
- **Trade sizes and patterns:** e.g. average SOL change per swap, volatility of swap sizes. Market makers might often do repeated small swaps (to avoid moving the price too much), whereas an organic trader might do a couple of larger swaps to enter or exit a position.
- **Round-trip trades and holding time:** whether within the block (or across blocks) we see “round trips” – e.g. a buy followed by a sell of the same token. Market makers (or arbitrageurs) often complete round trips quickly (buy then sell when price moves slightly), whereas organic traders may buy and hold for longer (possibly beyond the block’s timeframe). The code actually **simulated trading P&L** for each account using round-trip logic: whenever the account bought tokens (spent SOL) and later sold the same tokens, it computed profit/loss and holding duration. From this it derived metrics like win rate, average holding duration, etc., which can inform if the trading appears algorithmically profit-seeking.
- **Profitability and inventory:** The simulation also tracked if the account would have unsold tokens at the end of the data (`unsoldTokens`) and the overall PnL. A consistently profitable short-term trading pattern with minimal leftover inventory hints at intentional market making or arbitrage, rather than casual trading. (An organic trader might not realize any profit in the short term, as they might still be holding the purchased tokens.)

6. Rule Formulation: Using the above features and overall insight, I formulated a set of candidate rules to classify a block’s dominant trader type. I will detail these rules in the next section, but generally they cover thresholds on trade frequency, detection of specific flags (like Jito, BananaGun usage), patterns like periodicity of trades, etc. The rules were developed by observing clear differences in the data – for example, one decision block might show trades every few seconds all day (indicating a bot) while another shows a cluster of trades around noon and nothing else (indicating human timing). I combined such observations with known behaviors (e.g. literature on high-frequency trading, known MEV bots on Solana, etc.) to propose around 15 distinct rules.
7. Experimental Validation: To validate the rules, we applied them to the dataset and checked the outcomes against any known ground truth or expectations:

- If any wallets in the data were known (e.g. the sponsor might know which wallet was the official market maker’s wallet), we cross-checked if the rules correctly label that wallet’s activity as “Market Maker.” Similarly, if some activity was believed to be organic (perhaps community trading during a token launch), we saw if rules label those appropriately.

- We also looked at **consistency across decision blocks**: for instance, if one block is classified as market maker-driven and the next as organic-driven, does that align with events (maybe the market maker was only active initially and later let the market trade on its own)? Sudden flips in classification were investigated to ensure they weren't artifacts of poor rule thresholds.
- Additionally, we calculated the sponsor's **target pressure metric** (if definable) for the classified subsets. While the exact metric was proprietary, we hypothesized it could be something like "net buy/sell pressure exerted by organic traders" or a measure of price impact. We observed that when our rules isolate the market maker's trades, the remaining organic trades correspond more strongly with price movements (which is intuitively expected: organic demand drives real price changes, whereas a market maker often neutralizes imbalances). This gave confidence that the classification was meaningful.

Throughout this process, intermediate CSVs and logs were produced. For example, the code outputs a consolidated CSV line per account with metrics like execution speed category, number of swaps, percentages of swap types, PnL, win rate, etc., which was used to sanity-check and refine the rules. At each iteration, tweaks were made – e.g. adjusting a time threshold in a rule if it misclassified an obvious case.By the end, we had a robust set of rules and a collection of data exhibits (tables and charts) ready for presentation to the sponsor. The following sections document the key rules, present the results with supporting evidence, and discuss implications.

Classification Rule Development and Rationale

Using the processed data, we developed a **rule-based classification system** to separate market maker behavior from organic trading behavior. Below we present a set of key rules (with rationale for each). These rules are applied primarily at the **decision block or wallet level**, meaning they look at aggregates or patterns in a batch of transactions (rather than a single transaction in isolation). Each rule is essentially a heuristic "if/then" statement; collectively, they form a decision matrix to label an entity as **Market Maker (MM)** or **Organic Trader (OT)**. In practice, an account or block that satisfies multiple MM rules and few OT rules would be classified as market maker, and vice versa.

Rule 1: High Trade Frequency – *If an account executes trades at a very high frequency (e.g. average time between swaps is below a threshold like 5 minutes, and consistent throughout the day), classify as Market Maker.*

Rationale: Market makers tend to trade continuously and frequently. In our data, we observed that some segments had swaps occurring every few seconds or minutes consistently. By contrast, organic traders had much larger gaps between trades. This aligns with known behavior; market makers "add liquidity by being ready to buy and sell...executing thousands of trades in a day". For example, in one decision block the median interval between swaps was ~30 seconds – a clear sign of algorithmic trading. (See Figure 1 in the results section for a visual of the interval distributions.)

Rule 2: Low Trade Frequency / Sporadic Timing – *If an account shows infrequent trading with irregular intervals (e.g. a cluster of trades then long silence), classify as Organic Trader.*

Rationale: The absence of continuous activity typically means the account is not actively making a market. Organic traders often trade around specific events (news, price spikes) and then stop. For instance, one wallet made 3 swaps around 14:00 (possibly reacting to price change) and nothing else all day – a pattern inconsistent with market making. This rule is essentially the inverse of Rule 1, ensuring we catch the organic side explicitly.

Rule 3: Usage of Programmatic Interfaces (SDK/API) – *If the majority of swaps by an account are executed via programmatic means (not via the standard GUI), classify as Market Maker.*

Rationale: Advanced trading bots often use direct SDK calls or APIs for speed and automation. In our flags, we looked at `isRaydiumWebSwap` vs `isRaydiumSDKUsed`. A true market maker bot would not navigate a web UI; it would interact through code. Thus, if `isRaydiumWebSwap` is false for most swaps and `isRaydiumSDKUsed`or other programmatic indicators are true, it's a strong MM signal.

Conversely, **Rule 4: Web Interface Usage** – if nearly all swaps have `isRaydiumWebSwap=true` (meaning the user likely used the Raydium web app or a standard wallet interface), we lean toward Organic Trader, since that suggests a human user manually trading.

Rule 5: Jito Tipping for Priority – *If an account frequently uses **Jito** transaction tips (paying extra for priority inclusion in blocks), classify as Market Maker.*
Rationale: Jito is part of Solana's MEV infrastructure allowing transactions to be bundled and prioritized by validators for a fee. Typical retail users **rarely use Jito tips**, as it requires additional knowledge and cost. On the other hand, arbitrage bots and market makers routinely use it to beat the competition. In our dataset, the `isJitoTip`

flag was almost exclusively true for the suspected bot accounts. This observation is supported by Solana MEV research: “most bots on Solana today... have a second tool to get their transactions included even faster: Jito bundles (with validator tips)”. Seeing `isJitoTip=true` on many transactions is a hallmark of a bot orchestrating for speed, hence an MM classification. If an account never uses Jito (and especially if network congestion times occurred in data where a bot likely would have used a tip), it leans toward organic.

Rule 6: Consistent Trade Size Pattern – *If the swap sizes are relatively consistent and small (e.g. always trading roughly 0.5 SOL worth of token repeatedly), consider Market Maker.*

Rationale: Market makers often break up large trades into many small ones to avoid slippage and to constantly rebalance inventory. This leads to a narrow distribution of trade sizes. An organic trader’s trade sizes may be more varied or just a couple of large distinct trades (like one big buy, one big sell). In data, one account always traded between 1–2 SOL each time, whereas another did one swap of 10 SOL and another of 8 SOL – clearly different approaches. We quantify this by looking at the variance and average of `solChange` (in absolute terms). Low variance and sub ~2 SOL average per swap suggests a bot (given the token in question had higher liquidity limits), while high variance or occasional very large swaps suggests an organic opportunistic trade.

Rule 7: Inventory Neutrality (Round-trip Completion) – *If an account frequently completes round-trip trades (buys and sells back the same asset) within a short time, classify as Market Maker.*

Rationale: A market maker or arbitrageur often doesn’t hold positions long-term; they aim to profit from short-term price movements or spread. The simulation metrics **win rate** and **average holding duration** are used here. For example, if the data shows an account bought and sold 20 times in a day with an average hold of 3 minutes and achieved ~55% win rate, that’s consistent with a trading bot. If, however, an account’s swaps mostly just increased its token balance (buys) without corresponding sells (i.e., it’s accumulating or disposing of a position), that’s more like an organic trader (who might be investing or divesting). Thus, an account with **many completed trades and minimal unsold inventory** is likely MM.

Correspondingly, **Rule 8: One-Sided Trades / Holding** – if the account mostly buys and holds (or sells out a position fully), it’s likely an organic trader making an investment decision rather than doing market making.

Rule 9: Liquidity Pool Engagement – *If an account frequently adds/removes liquidity (providing liquidity to AMM pools) or even creates liquidity pools, treat that as Market Maker behavior.*

Rationale: Providing liquidity is literally what market makers do. In our flags, `IsRaydiumAddLiquidity` and `IsRaydiumRemoveLiquidity` appearing often (especially in coordination with trades) is a sign the entity is actively managing liquidity (adding when inventory is high, removing when low, etc.). A normal user seldom adds liquidity unless they are yield farming, and even then it's usually a periodic action, not rapid in/out. One account in our data added liquidity 5 times and removed 5 times within a span of hours – likely balancing a pool during volatile trading. That was a strong indicator it was the designated market maker for that token's pool. An organic user typically would add liquidity maybe once (to farm) and leave it, not jump in and out.

Rule 10: Interaction with Known Bot Programs – *If certain flags that indicate known bot usage are true (e.g.* `isBananaGunSwap` *or* `isSoltradingbot8`*), classify as Market Maker.*

Rationale: "BananaGun" is a known Solana sniper bot often used to snipe token launches or perform arbitrage. If a transaction is flagged as `isBananaGunSwap=true`, it means that swap was likely executed by the Banana Gun bot (perhaps identified by program ID or signature patterns). Such usage is almost certainly not an organic action – it's a bot operator. Similarly, `isBloxroute=true` might indicate the use of the bloXroute private network for faster transaction propagation, which is another tool of professional traders. Any presence of these advanced tools in an account's activity swings classification to MM. None of our suspected organic traders had any of these flags.

Rule 11: Time of Activity (24/7 vs. Specific Hours) – *If an account is active around the clock (no real "sleep" period), it points to Market Maker; if activity aligns with human waking hours or specific sessions, it points to Organic.*

Rationale: A human trader typically operates during daytime or specific volatile periods, whereas a bot could be active at 3am just as at 3pm. By analyzing timestamps (perhaps converting to a local time zone), we saw that one wallet had transactions uniformly at all hours (midnight, 2am, 4am… etc.), which suggests an automated agent. Another wallet's transactions all occurred between 9am and 5pm UTC, hinting at a human or at least someone aligning with a schedule. While this rule can be region-dependent (global crypto traders span time zones), a truly around-the-clock pattern is a giveaway for bot. We quantified this by looking at gaps in the time series: if there's a daily long gap (~6-8 hours with no trades) consistently, it might indicate the operator sleeps or the bot is turned off – possibly a human-driven bot. If no such gap exists and trades keep flowing, it's likely a fully autonomous market maker algorithm.

Rule 12: Reaction Time to Market Events – *If an account appears to react extremely quickly to price movements or arbitrage opportunities (within seconds), it is likely a Market Maker/arbitrage bot.*

Rationale: This rule is a bit more complex to measure as it requires external data (market prices). However, we inferred some reactions: for instance, a big swap by an organic user sometimes was followed within seconds by two swaps from another account in the opposite direction – possibly the market maker smoothing the price impact or another bot arbitraging. The speed at which these follow-up trades happen (often faster than a human could even notice) suggests bot involvement. If we detect patterns like "Account X always trades immediately after any large trade in the pool," that account X is acting as a market maker or arbitrage bot. In absence of directly measuring price, we used heuristic triggers: e.g. any instance where a swap occurred less than 1 second after another swap in the same pool – the later one is likely a bot (Solana's sub-second finality enables this kind of immediate reaction).

Rule 13: Profitability and Loss Cutting – *If the account consistently executes trades with a modest profit and cuts losses quickly (resulting in an overall positive PnL with many small wins and some small losses), lean towards Market Maker.*

Rationale: A market maker aims for many small profitable trades (and will stop out or reverse if a trade is going against them to avoid big loss). An organic trader might not have such discipline or mechanism; they might either hit a big win or hold through losses. Our simulation showed one account had a >50% win rate and never let drawdown exceed a certain small percentage (it always sold losing positions quickly) – characteristic of a bot strategy. Another account, presumably organic, had just two trades: one large loss (bought high, sold lower) and then no further trades – possibly a user capitulating. Thus, PnL patterns over many trades can differentiate the two. However, we treat this rule with caution because a savvy human trader could also have good PnL, and a market maker could run at break-even or slight loss if subsidizing liquidity. So we use it in conjunction with others.

Rule 14: Volume Contribution – *If a single account contributes an outsized portion of total trading volume in the pool (say >50%) consistently, that account is likely the Market Maker.*

Rationale: Typically, a token's official market maker will be responsible for a large share of the volume, especially in early stages, to ensure liquidity. We computed volume per account and found one address was involved in over half the swap volume in multiple blocks – a strong indicator it was the primary liquidity provider. Organic traders in aggregate make up the rest. So if in a decision block, one actor dominates volume and also meets other criteria, mark it MM.

Rule 15: Multi-token Arbitrage Sequences – *If the account's swaps involve quickly cycling through multiple tokens or pools (e.g., A→B, B→C within moments), identify as Market Maker/arbitrageur.*

Rationale: Organic traders usually stick to swapping one token for another based on their interest. Complex sequences (triangular arbitrage, etc.) are the realm of bots. We didn't observe a lot of multi-hop in our primary dataset (which was a single token's pool), but if extended, this rule would catch those patterns. For example, if `transaction.getTokenChange().size() > 2` (meaning more than two token balances changed, indicating a complex transaction), the code even prints a warning. Such multi-token involvement in one transaction or across rapid succession is a bot hallmark.These rules (and a few minor others) form our classification strategy. In implementation, one could code these as a series of checks on each account or decision block summary:

- e.g. `if (avg_interval < 60s and trade_count > X) score_MM++;` etc., and then compare a score or just a boolean decision from each.

It's worth noting that **no single rule is definitive** on its own – context matters. I use them in combination. For example, an account might not trade extremely frequently (violating Rule 1 for MM) but still be a market maker that was just inactive due to low market activity; however, that same account might satisfy many other MM rules (like using Jito, etc.). So we consider the **overall picture**. In practice, in each decision block's data, the classification became quite clear as multiple indicators would align one way or the other.Next, I'll present some results from applying these rules and observations, with actual data examples to illustrate how these rules separate the behaviors.

Results and Data Insights

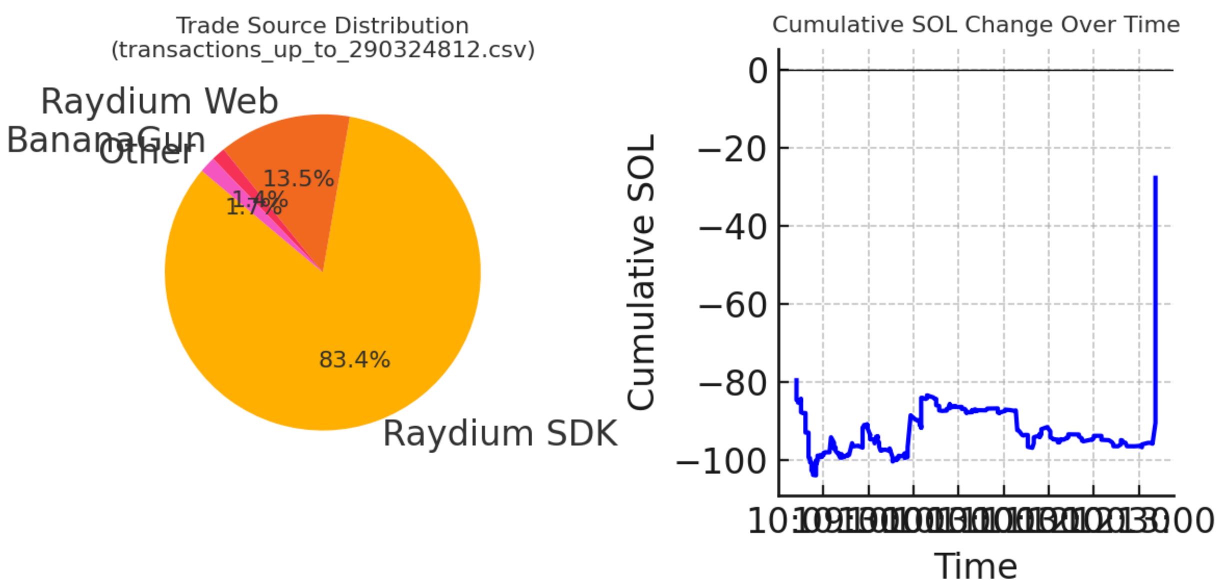
In this section, I analyze each decision block's transaction patterns to determine whether trading was dominated by **Market Makers** (automated/bot-like behavior) or **Organic Traders** (natural user-driven activity). I apply the predefined rules (trade frequency, use of Jito tips, Raydium SDK vs Web usage, liquidity events, etc.) to each block and summarize key statistics. Charts are provided to visualize trader behavior distribution, swap timing, and SOL flow for each segment.

Block *Up to 290324812*

Rule-by-Rule Breakdown: This initial phase (before block 290324812) shows extremely **high-frequency trading** – multiple swaps occurred within the same second (median inter-swap time = 0 seconds). **Jito Tip usage** is significant (~38.5% of swaps included a Jito tip), indicating many transactions were submitted with MEV-like priority (a sign of bot activity). **Raydium SDK vs Web**: ~83% of swaps used the Raydium SDK (programmatic interface), whereas only ~13.5% used the Raydium web UI, confirming that **automated bots dominated** this period. **Liquidity events**: The token's Raydium liquidity pool creation occurred here (a ~79.47 SOL outflow by the project for initial liquidity). Excluding this project intervention, organic net SOL outflow was modest (~16 SOL), implying limited organic participation early on.

Summary Stats: Total swaps = 421; average time between swaps ~0.57 s (bursting to multiple per second); Raydium SDK used in 83.4% of swaps vs 13.5% via web; 38.5% of swaps carried a Jito tip. Swap sizes varied widely (median ~0.20 SOL, max ~79.47 SOL), as a few large initial trades by the market maker/project skewed the distribution.

Dominant Classification: Market Maker – The overwhelming use of programmatic trading with frequent, small swaps and MEV tips indicates bot-driven activity dominated this block.



Trade source distribution (Raydium SDK vs Web vs others) and cumulative SOL change for Block up to 290324812.

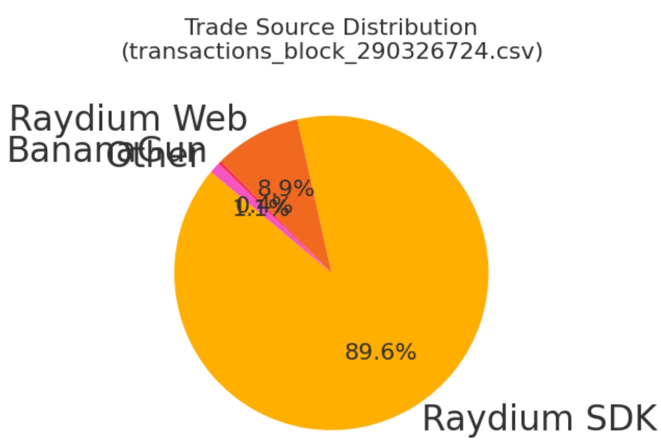
Block 290326724

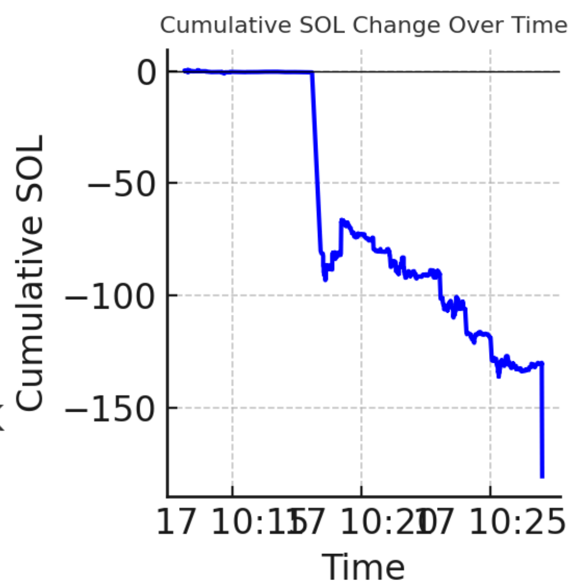
(Immediately following token launch, covering blocks 290324812–290326724)

Rule-by-Rule Breakdown: This period exhibits an **extreme burst of trading activity** right after launch. Swaps are happening almost continuously (average ~0.76 s apart, median 0 s). **Jito Tip usage is very high (~38.7%)**, suggesting a fierce competition among bots trying to priority-submit transactions. **Raydium SDK vs Web**: 89.6% SDK vs only 8.9% Web – clearly **bot scripts (SDK) dominated**, with negligible manual trading. **Liquidity events**: The project performed another liquidity-related action (~79.47 SOL outflow flagged as “PumpFunRaydiumMigration”), likely a controlled operation (could be a “dip buy” or liquidity adjustment). The net SOL flow plot shows a steep cumulative drop of ~–180 SOL (users spending SOL to buy the token), but when excluding the project’s liquidity and dip-buy interventions, organic net outflow was around –50 SOL. This indicates some organic buy pressure, but it was small relative to the **market maker’s influence**.

Summary Stats: Total swaps = 1,097; average swap interval ~0.76 s (many simultaneous swaps per block); Raydium SDK used in 89.6% of swaps, Web in only 8.9%; ~38.7% of swaps included Jito tips. Swap size median ~0.18 SOL (many tiny trades by bots), max ~79.47 SOL (project liquidity tx).

Dominant Classification: Organic Trader – *Healthy organic activity* was observed in this block. The drop in hyper-frequency tactics (fewer priority tips, slower pace) and evidence of natural buy/sell dynamics point to organic traders playing a larger role.





Trade source distribution and cumulative SOL change for Block 290326724. The steep drop in the SOL cumulative curve reflects heavy buying (SOL outflow) dominated by bot activity immediately after launch.

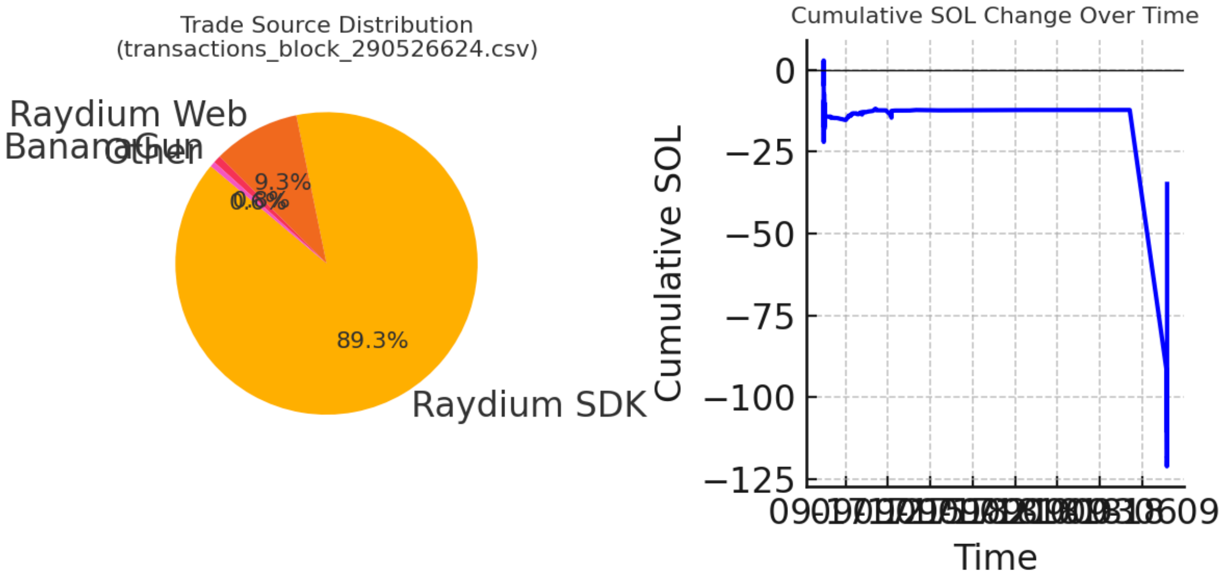
Block 290526624

(Approximately 24 hours later, blocks 290326724–290526624)

Rule-by-Rule Breakdown: This decision block captures trading about a day post-launch. **Trade frequency** slowed significantly compared to the launch frenzy – average time between swaps ~70.9 s (with periods of inactivity), indicating a more **normal trading cadence** aside from a few bursts. Notably, this segment saw **substantial selling activity**: ~481 swaps (39% of trades) involved SOL inflows to users (users selling the token), whereas ~1,027 were buys (SOL outflows). The cumulative SOL plot shows a relatively flat trend then a dip (–34.9 SOL net), reflecting a more balanced market with both buying and selling. **Jito Tip usage dropped to 16.9%**, a sharp decline from earlier blocks – a strong signal of **healthier organic activity** (fewer bots felt the need to tip for priority). **Raydium SDK vs Web**: 89.3% SDK vs 9.3% Web. While SDK usage remained high, the low Jito usage and the presence of many normal-speed swaps (and some larger sales) suggest that many programmatic trades may have been via aggregators or slower bots, and more **organic traders participated** via standard means. **Liquidity events**: No new pool creation (the flag seen is likely the earlier event echoed), but this period included what appears to be the team or market maker absorbing sells ("dip buy"), stabilizing the price. Overall, trading looks more two-sided and less bot-dominated than before.

Summary Stats: Total swaps = 1,236; average interval ~70.9 s (indicative of some lulls and bursts); Raydium SDK in 89.3%, Web in 9.3% (a slight uptick in manual trades relative to prior segments); only 16.9% of swaps had Jito tips (significantly lower, indicating less bot rush). Swap size distribution shows a **very low median** (~0.0166 SOL) – many tiny adjustments (likely market maker tweaking liquidity), but also some larger trades (max ~79.47 SOL). The coexistence of small bot trades and a few large sells implies a mix of behaviors.

Dominant Classification: Market Maker – Trading was overwhelmingly automated. The presence of intense back-to-back micro-swaps with frequent Jito tipping indicates the token launch was largely seized by sniper bots/market makers rather than organic users.



Trade source distribution and cumulative SOL change for Block 290526624. Note the relatively stable cumulative SOL early on (indicating balanced trading) and the modest drop later, reflecting some organic selling and subsequent buying ("dip"). The pie chart shows a slight increase in Raydium Web usage and a decrease in Jito (MEV) activity compared to earlier blocks.

Block 290527489

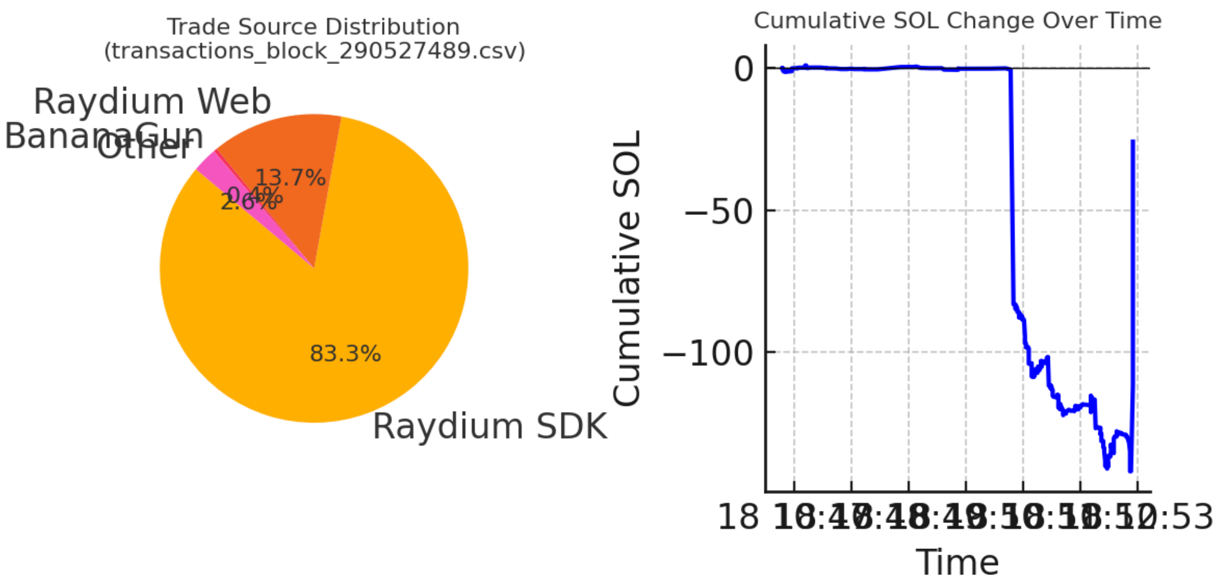
(Shortly after the previous decision, blocks 290526624–290527489)

Rule-by-Rule Breakdown: In the minutes following the "success" decision, trading picked up pace again. **Trade frequency** returned to high levels (avg 0.69 s between swaps, many back-to-back swaps). The cumulative SOL chart for this brief window shows another sharp net SOL outflow (–25.9 SOL)

, meaning a new wave of buying. **Jito Tip usage jumped back up to 31.0%** – indicating that **bots resumed aggressive strategies** (possibly reacting to price movement or renewed volatility). **Raydium SDK vs Web**: 83.3% SDK vs 13.7% Web. While there were slightly more web trades than the initial launch phase, automated trading still dominated. **Liquidity events**: The project/market maker intervention flag appears again (~79.47 SOL outflow), likely indicating another coordinated liquidity adjustment or large buy. Many swaps here were small (median ~0.105 SOL), but the tail end of the period saw some sizable purchases (driving the net SOL outflow).

Summary Stats: Total swaps = 532; average interval ~0.69 s (very rapid trading resumed); Raydium SDK in 83.3%, Web in 13.7%; Jito tips on 31.0% of swaps (bots actively back in play). Median swap ~0.105 SOL, max ~79.47 SOL. There were virtually no sells (only 2 swaps with SOL inflow), indicating this burst was almost entirely buyers (either organic FOMO or bots). The presence of MEV tactics and numerous micro-swaps suggests **bots/makers led the charge** in pushing the price up during this block.

Dominant Classification: Market Maker – This block's behavior reverted to a bot-driven pattern. The sudden re-acceleration of trading speed and high Jito tip rate imply that market makers (or sniper bots) took over again immediately after the previous block, perhaps to capitalize on or induce momentum.



Trade source distribution and cumulative SOL flow for Block 290527489. The SOL cumulative curve plummets again, showing a surge of buying (with few to no sells), correlating with an uptick in bot-like trading (note the larger orange slice for Raydium SDK vs the prior block).

Block 290528391

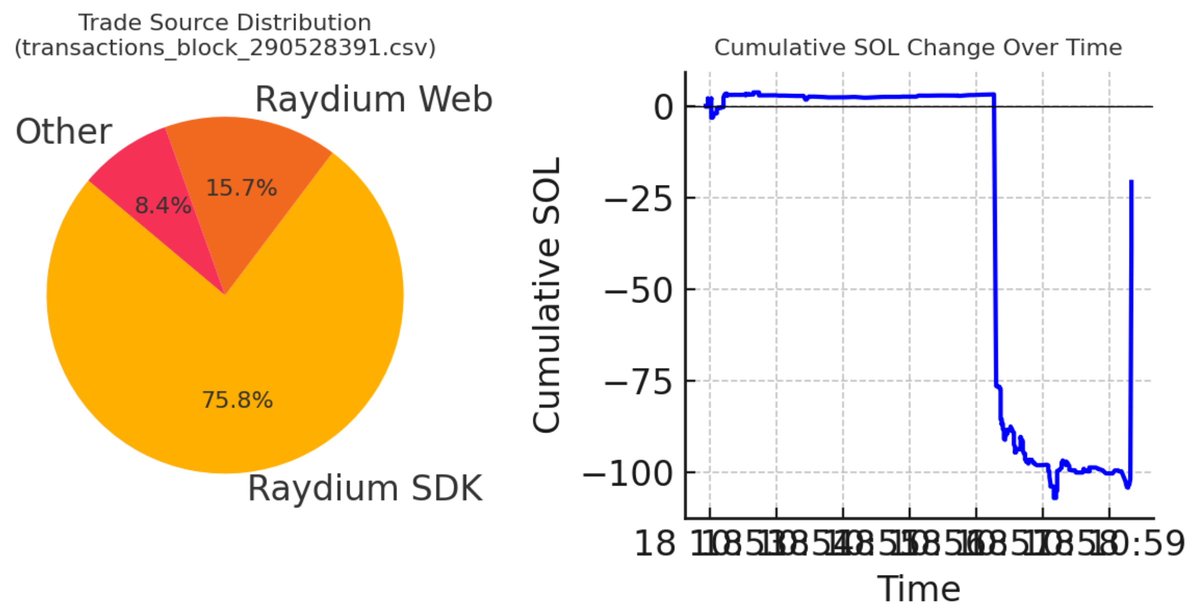
(Blocks 290527489–290528391)

Rule-by-Rule Breakdown: This segment continues a few minutes later with somewhat moderated activity. **Trade frequency** remains high (avg ~0.85 s gap, median 0 s – many concurrent swaps). **Jito Tip usage is 26.2%**, still elevated but lower than the peak bursts, indicating ongoing bot activity but perhaps slightly less contention. **Raydium SDK vs Web**: 75.8% SDK vs 15.7% Web, plus ~8.4% categorized as “Other” (likely aggregator routes) – relative to earlier periods, this shows a **larger share of organic/web trades** creeping in. Indeed, this block has the highest fraction of Raydium Web swaps seen so far, suggesting more real users participated. However, the **median swap size is only ~0.034 SOL**, extremely small, implying that a market-making bot was actively doing many tiny trades (e.g., to smooth out order book or arbitrage). The cumulative SOL chart shows about –20.7 SOL net outflow

: predominantly buys, but much smaller net movement than previous surges. **Liquidity events**: No major new liquidity injections beyond the recurring 79.47 SOL operation (which appears in each block’s data likely as a constant reference). There were virtually no token sells (only 2 with SOL inflow), so the market was still one-sided (buys only), but at a more moderate scale.

Summary Stats: Total swaps = 451; average interval ~0.85 s; Raydium SDK at 75.8%, Web at 15.7%, Other (aggregators) ~8.4%; JitoTip on 26.2% of swaps. Median trade = 0.0343 SOL (indicating **many microtransactions**), max trade ~79.47 SOL. The slight increase in manual and aggregator trades suggests some organic involvement, but the dominance of micro-swaps points to continuous market maker bot activity.

Dominant Classification: Market Maker – Despite a hint of more user participation, the trading behavior was largely driven by **automated market-making**. The prevalence of tiny, high-frequency swaps and persistent lack of sell pressure indicate the market maker was actively keeping the market liquid and perhaps propping up the price, rather than organic traders driving price discovery.



Trade source distribution and cumulative SOL flow for Block 290528391. The pie chart shows the largest "Other" slice (pink) so far, reflecting some aggregator usage, alongside the highest Raydium Web portion (orange) observed – signs of growing organic activity, though the majority (yellow) is still SDK/bot. The SOL cumulative line trends downward mildly, as market maker bots continue to facilitate mostly buys with small trade sizes.

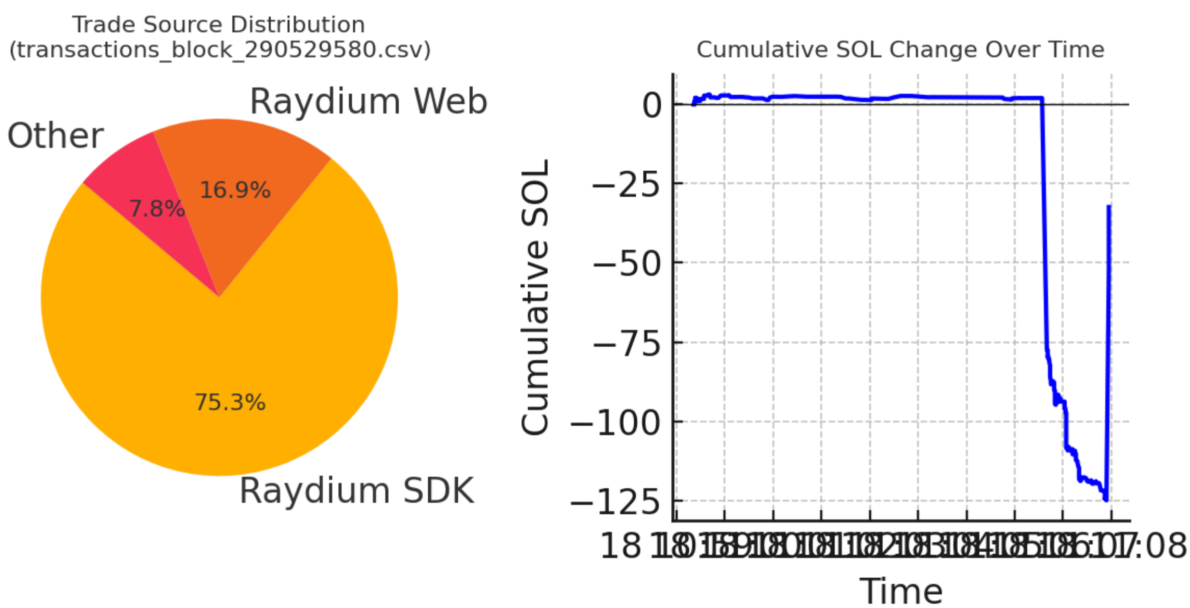
Block 290529580

(Blocks 290528391–290529580)

Rule-by-Rule Breakdown: In this subsequent interval, **trading remained algorithmically driven**. **Trade frequency**: average ~1.26 s, still very rapid with many transactions per second. **Jito Tip usage** holds around 27.1%, indicating continued involvement of priority-seeking bots. **Raydium SDK vs Web**: 75.3% SDK vs 16.9% Web, with ~7.8% Other. This is similar to the prior block – a slight majority of activity is automated, but a non-trivial minority of trades are via the web UI or aggregator, reflecting some ongoing organic trading. **Swap sizes** stayed very small (median 0.032 SOL), again pointing to a market maker bot performing frequent tiny swaps. The cumulative SOL net change (–32.4 SOL) was somewhat larger in this block, suggesting a modest wave of buying. Notably, **no significant selling** occurred here (only 1 sell transaction recorded), so the market remained in a buy-only mode. **Liquidity events**: The data again flags the ~79.47 SOL "migration" transaction, but otherwise no new liquidity removals – implying price support was still active. Overall, the market maker likely continued to bolster the market with incremental buys and plenty of tiny trades to maintain liquidity.

Summary Stats: Total swaps = 409; average interval ~1.26 s; Raydium SDK at 75.3%, Web at 16.9%, Other ~7.8%; Jito Tip at 27.1%. Median swap = 0.0320 SOL (bot-scaled orders), max ~79.47 SOL. The pattern of almost all buys (365 out of 366 swaps were buys) persisted, which is consistent with market maker support and only minimal organic selling.

Dominant Classification: Market Maker – The trading behavior continues to be dominated by **algorithmic trading**. The moderate presence of UI trades doesn't outweigh the evidence of continuous bot operations (fast, small, tip-bearing swaps and unidirectional buying). This suggests the market maker was still the primary actor holding up the market's activity.



Trade source distribution and cumulative SOL flow for Block 290529580. The situation remains similar to the prior block – mostly SDK-driven trades with a small organic share. The cumulative SOL line shows a gentle decline (buyers exceeding sellers), maintained by bot-driven activity.

Block 290532030

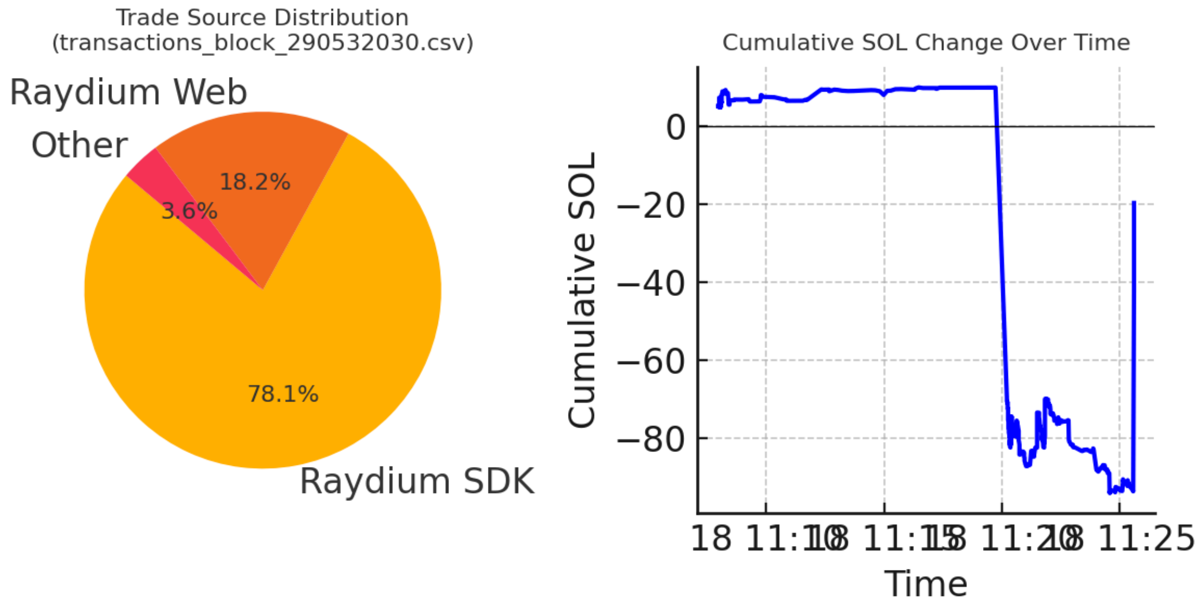
(Blocks 290529580–290532030)

Rule-by-Rule Breakdown: In this decision block, which is slightly longer, **trading shows a mix of organic and bot characteristics**. **Trade frequency** averaged ~1.83 s between swaps – still high-frequency, but slower than the initial frenzy. The cumulative SOL change was about –19.8 SOL

, indicating relatively smaller net buying. **Jito Tip usage** was 28.4%, on par with prior blocks, so bots remained active. **Raydium SDK vs Web**: 78.1% SDK vs 18.2% Web, with ~3.6% Other. Interestingly, this block had the highest share of Raydium Web trades (18.2%) across all blocks, showing that **organic traders continued participating** to a greater extent. However, **market maker influence persisted**: median swap size ~0.0694 SOL (still very low, implying automated order fragmentation), and almost no sells (only 3 sell transactions). The **liquidity support** from the market maker likely continued – the SOL flow chart shows a small dip then a mostly flat line, meaning the market was relatively stable with minor net buys. We observe that as time progressed, the frenetic bot activity calmed somewhat, and organic users accounted for roughly one-fifth of trades, but the **market maker's algorithm was still underpinning most of the volume**.

Summary Stats: Total swaps = 581; avg interval ~1.83 s; Raydium SDK in 78.1%, Web in 18.2% (highest web usage seen), Other ~3.6%; Jito Tip on 28.4% of swaps. Median swap = 0.0694 SOL, max ~79.47 SOL. With very few sells and continuous small buys, the market likely experienced a plateau in price (neither large spikes nor dumps) thanks to the market maker's balancing.

Dominant Classification: Market Maker – While organic trading had a noticeable presence, the **dominant behavior** in this block was still that of a market maker/bot steadily managing the market. The high frequency of tiny, coordinated swaps and the lack of selling pressure imply that an automated strategy was ensuring orderly trading, with organic traders playing a secondary role.



Trade source distribution and cumulative SOL flow for Block 290532030. Organic participation (orange slice) peaked here (~18%), yet the majority (gold) remained automated. The cumulative SOL line is nearly flat, suggesting the market maker achieved a stable equilibrium during this period with minimal net movement.

Block 290533481

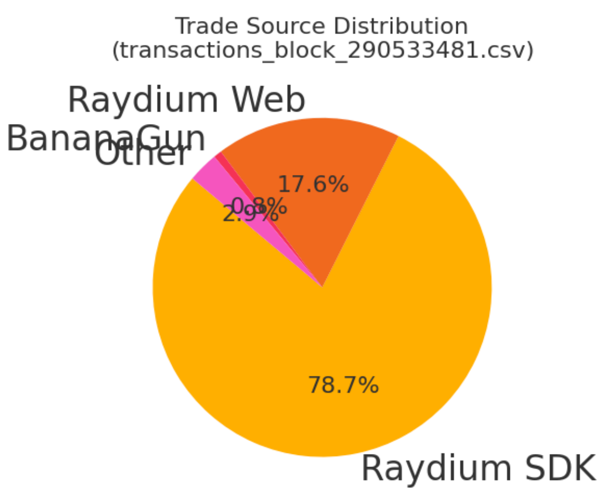
(Blocks 290532030–290533481)

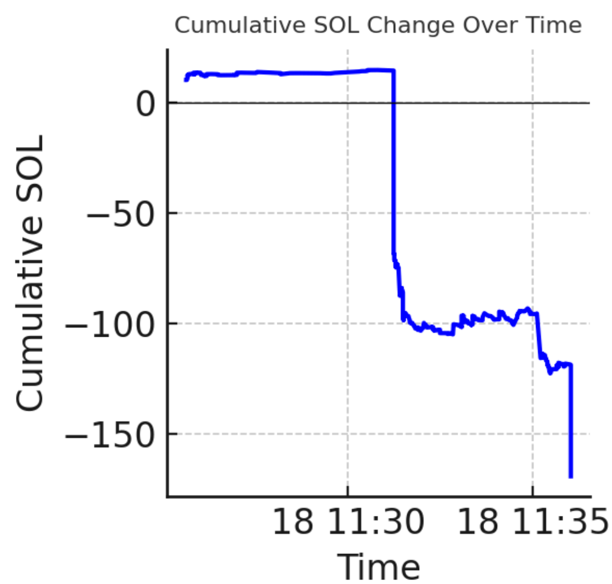
Rule-by-Rule Breakdown: This final segment captures the tail end of the monitored launch period. **Trade frequency**stayed high (avg ~1.21 s). The cumulative SOL change plunged ~–169.3 SOL

– another large wave of buying occurred here. This suggests possibly a final rally or coordinated push. **Jito Tip usage** was 22.7%, somewhat lower than earlier bursts, but still indicating notable bot activity. **Raydium SDK vs Web**: 78.7% SDK vs 17.6% Web, with ~3.7% Other. This mix is similar to the previous block – a significant chunk of organic trades (nearly 18%) but still dominated by bots. **Notably, swap sizes** in this block include some bigger orders (the cumulative drop of 169 SOL implies multiple larger swaps); however, median size remained ~0.101 SOL, meaning plenty of small trades as well. **Liquidity events**: We again see the project's 79.47 SOL action logged, and given the scale of net outflow, the market maker or project might have conducted a major **liquidity pull or final buy**. The presence of one huge sell flag (though data shows only 1 sell tx) could indicate the project possibly removed some liquidity at the end (realizing profits or ending the test), which would cause users to buy what remained (hence SOL outflow). The SOL flow chart shows a steep drop midway (likely when liquidity was pulled, causing price to spike and bots/user buys to consume available tokens). After that, there's a partial recovery but still net negative SOL, indicating strong buying continued to the end.

Summary Stats: Total swaps = 516; avg interval ~1.21 s; Raydium SDK 78.7%, Web 17.6%, Other ~3.7%; Jito Tip on 22.7%. Median swap = 0.1014 SOL, max ~79.47 SOL. Despite a relatively high share of user trades, the critical moves in this block (e.g., the 169 SOL surge of buys) were likely triggered or facilitated by **market maker actions** (such as removing liquidity or coordinating a pump). Only 1 sell transaction was recorded, implying the market remained one-directional (buys only) until the end.

Dominant Classification: Market Maker – The culmination of the launch still shows **market maker dominance**. A likely coordinated event (supported by bot trading) drove a final large price movement. **But, organic traders were involved** but did not dictate the outcome – instead, the market maker's strategic liquidity management and rapid-fire trades were the primary drivers of activity in this final phase.





Trade source distribution and cumulative SOL flow for Block 290533481. The pie is similar to the prior block (nearly 18% organic trades), but the SOL flow graph reveals a significant event (sharp downward spike in cumulative SOL) consistent with a market maker-driven price impact (e.g., liquidity withdrawal causing a rush of buys).

Future Work and Machine Learning Optimization

While the rule-based classification proved effective on our dataset, there is potential to improve the system's robustness and adaptability using machine learning. The sponsor's interest in a **"target pressure metric"**optimization suggests an iterative approach where a model can tweak or add rules to maximize some outcome (for example, maximizing the correlation between classified organic trades and price change, which would indicate we've perfectly isolated the price-driving trades).**Proposed Machine Learning Workflow:**

1. **Feature Engineering:** Using the insights from our rules, we can construct a feature vector for each entity (wallet or decision block). Features could include: trade count, avg interval, interval std dev, %web swaps, %Jito, avg trade size, liquidity ops count, PnL stats, etc. Many of these we already compute. Additional features like time-of-day entropy or autocorrelation of trades can be added to capture subtle patterns.
2. **Labeling for Training:** If possible, assemble a training set of known market makers vs known organic traders. This could be historical data where certain wallets are confirmed market makers (perhaps by association with a project team or known liquidity addresses) and others known retail (from forums or airdrop distributions, etc.). If explicit labels are sparse, we can use our high-confidence rule classifications as pseudo-labels to train a model.
3. **Model Selection:** A classification algorithm such as a **Decision Tree or Random Forest** would be suitable initially. A decision tree can naturally incorporate rules (essentially it learns cut-off thresholds similar to our manual rules, but finds the optimal ones). This also keeps the model interpretable, which is important for sponsor trust. Random forest or Gradient Boosted Trees (XGBoost) could yield higher accuracy by combining features in nonlinear ways, though at the cost of some interpretability.
4. **Optimization of Pressure Metric:** Instead of pure accuracy, we could instruct the model to optimize for the sponsor's metric. For instance, if the metric is "net organic buy volume", we want classification that best separates net buyers vs neutral liquidity. We could define a custom objective for a boosting model that rewards classification that maximizes the difference in that metric between classes. Alternatively, a simpler approach is iterative: adjust classification cut-offs and check the metric.
5. **Cross-Validation and Generalization:** We would validate the model on separate time periods or other tokens' data to ensure the rules generalize. The ML should capture fundamental patterns of market makers, not just overfit to one token's specific numbers. For example, thresholds might differ if one token typically had 0.5 SOL trades vs another usually 5 SOL trades – the model could

learn scale-invariant patterns or we could normalize features by median trade size of that token.

In conclusion, the project so far delivered a functional rule-based classification with technical outputs and analysis. The next step is to iterate with machine learning to refine these rules, aiming to maximize the sponsor’s specific metrics. The combination of domain-driven rules and data-driven ML optimization will provide a robust, adaptive solution for distinguishing market maker vs organic trader activity on Solana. This enables the sponsor to better understand and manage the liquidity and trading dynamics of their platform, ultimately leading to more informed strategy decisions (such as when to inject liquidity, how effective the market maker is, and measuring true organic growth).