© Denis Makarov

v. 1.0
Feb 2, 2025

DEX market research based on whales multibuys

Following report presents short Web3 DeFi tokens analysis results, having its base on information gained from so-called 'whales' and 'smarts' multibuy alerts. Primary data was exported from Telegram bot, which sends notifications about unusual token activity (multiple buy) by Web3 wallets, that are known as traders with big winrate, PnL and balance.

This paper includes short description of the code and following results, hypotheses and insights. Work still in progress.
Not financial advice.

Glossary

- **DEX** – decentralized exchange = ecosystem independent from cryptocurrency stock martkets
- **whale** – individuals or entities who hold large amounts of cryptocurrency and can influence markets with their trades
- **smart** – individuals whose moves are mostly smart: having extra high winrate on spot
- **token** – DEX coin which has its contract address (CA), its market capitalization (MC) and price per 1 token
- **multibuy** – unusual occasion of multiple buying of 1 token by several whales/smarts
- **insight** – valuable conclusion or idea, extracted from data analysis results
- **winrate** – % of successful (positive) entry-exit deals
- **MC** – contains the value of all existing tokens, considering prices there were bought at

Stack

Programming language: Python
Libraries: Pandas, Numpy, Matplotlib, Seaborn, JSON
Instruments: Telegram Desktop app, Jupyter Notebook

Critical path

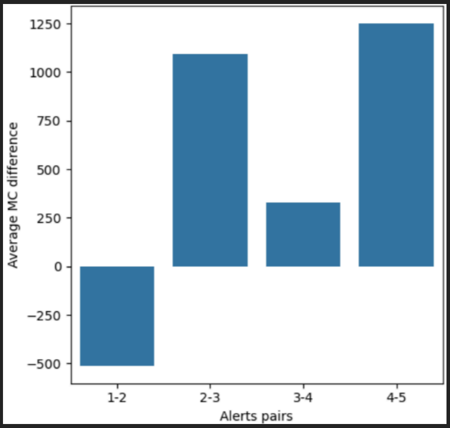
TO BE FINALIZED

1. Go to alert bot
2. Export chat history in Developer Mode of Telegram Desktop app as .json file
3. Parse it in json_to_csv.py → we got raw data, completely unreadable
4. That's why we clear it, deleting completely unnecessary columns and save it as unsolved.csv
5. Further, load it in normalize.py and get parsed_messages.csv as our messages context is presented as JSON objects → we actwally need to parse it properly.
6. We need to extract and create the needed columns from strings of messages, so we read 'parsed_messages.csv' in 'to_finish.py' and finally get 'wallet_data.csv'! We will work with this file in Jupyter.
7. Open Jupyter → crypto.ipynb → retire your bloodline (kidding)

Insights

Firstly, we may want to know when it is less risky to enter the token. After first, second or even further alert? Let's create a histogram, which illustrates different Market Cap diffrences in each pair of alerts.
Hint: token value in USDT ($) directly depends on its MC. When a person exits from token when at MC and its price higher than values that were when he entered, he gains extra %.

Average MC difference for each pair of alerts, thousands of $



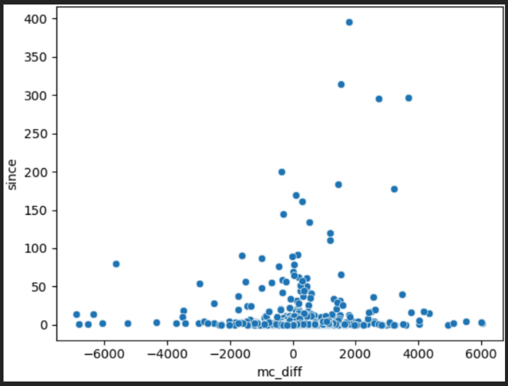
Average MC difference for each pair of alerts, %

Though, absoulte growth <> relative. That is why we want to see the difference in % and create the right plot.
To draw a conclusion, we need just one more thing to find out. What is the chance of particular alert to be the last one on each coin? Look at the histogram.

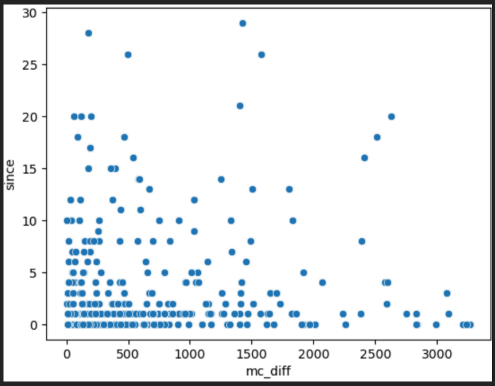
Second alert is the less risky and most profitable call to enter

Time is the most expensive resource we have. Therefore, we might want to know, where is the best duration of us staying in the token. Look at the plots above.

Scatterplot for detecting dependance between MC difference (K$) and Time (min) since last alert for pair 2-3

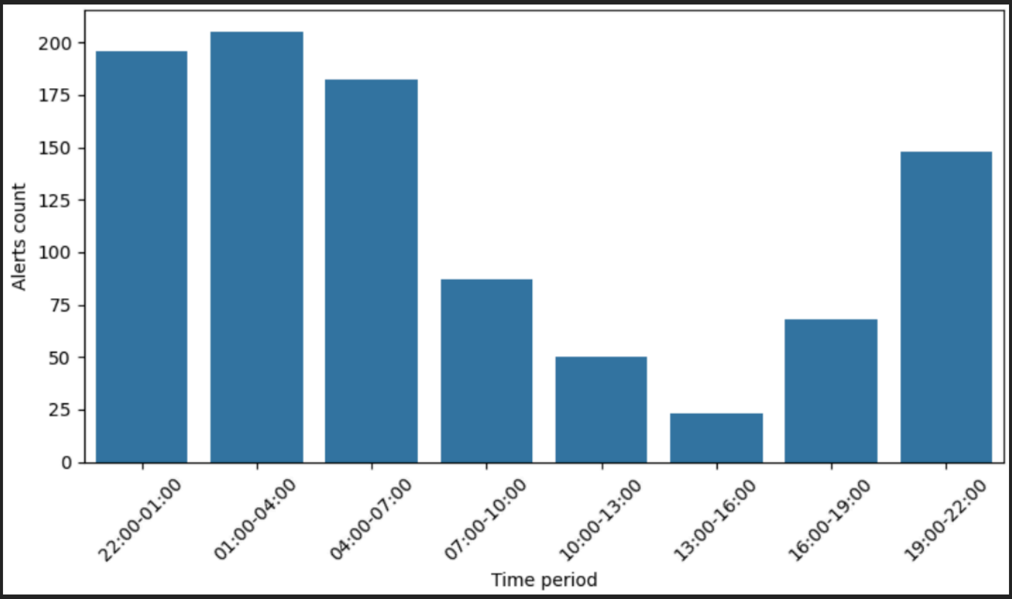


-> More detalised



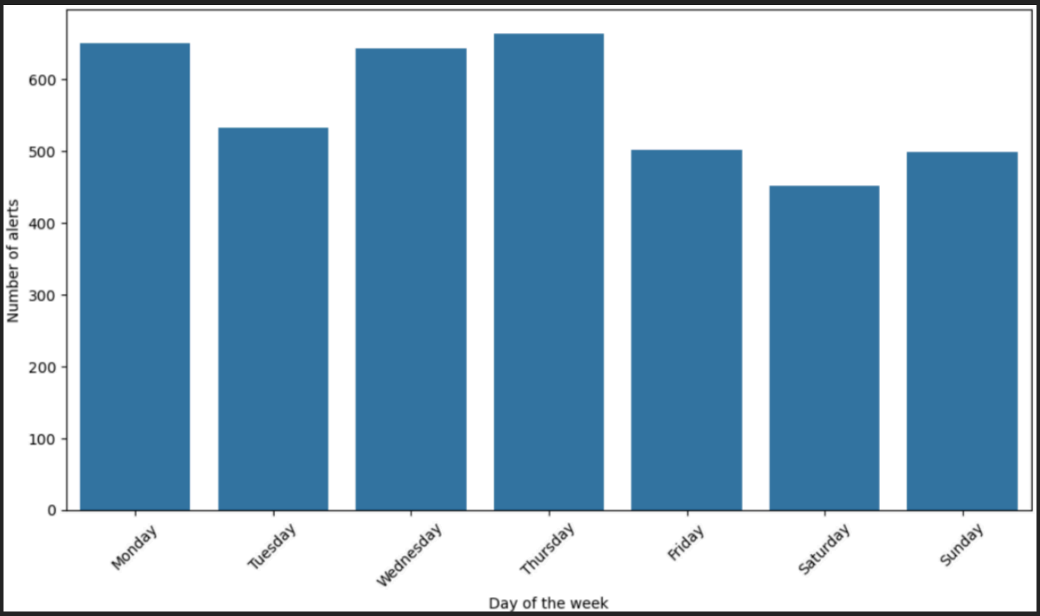
What is the most green time to hear the callouts?

Average number of 2nd alerts during the day



If you want the market not to be dead, you better live in EST timezone.

But which day?



Sunday-Monday night & Wednesday-Thursday night are the most profitable.

Not financial advice. To be continued...