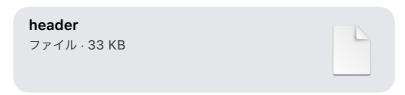
Solana Skill Sprint - Memcoin Graduation

Using transaction data from the first 100 blocks, predict whether the "Pump Fun" token price reached a specific level.



Overview
Data
Code
Models
Discussion
Leaderboard
Rules

Overview

Most memcoins created on the Pump Fun platform are rugged before they graduate to Raydium—and now to PumpSwap. But what if we could spot them early?

This competition challenges participants to build a machine learning model that predicts whether a newly launched Pump Fun token on Solana will **graduate**—using only data available in the first 100 blocks after the token is minted.

Your goal: Predict whether a token will reach at least 85 SOL in liquidity.

In practice, this means distinguishing tokens more likely to get rugged from those with a higher chance of success.

Along the way, you'll encounter the subtle patterns left by **serial deployers and snipers**—influential actors who shape a token's fate while trying to stay hidden.

Whether you're an ML enthusiast looking to apply your skills to crypto data or a blockchain analyst curious about predictive modeling, this competition is for you.

You can **ask your question** in "Discussion" section or at SSS community discord:

If you're new to ML, I highly recommend starting by reproducing the results from this baseline:

https://www.kaggle.com/code/dremovd/baseline-mem

Start

17 days ago

Close

12 days to go

Description

linkkeyboard_arrow_up

Most tokens on Pump Fun start strong but fade fast. Some graduate to larger DEXs. Many get rugged. And surprisingly, you can often spot the outcome using just the first transactions of on-chain activity. This track gives you access to curated datasets from the first 100 blocks post-mint:

- Initial swaps and liquidity events
- Token creation details
- Optional: any other historical Solana data available up to that block (external data use must follow the rules)

Your task: Predict whether the token will reach 85 SOL liquidity (graduate) based on early behavior.

This task is:

- Beginner-friendly, with a well-scoped prediction target and clean data
- **Rich for experts**, offering depth through behavioral signal detection, wallet profiling, and time-sensitive analysis
- **Relevant**, as skills here extend directly to tasks like real-time rug pull prediction and ML-powered trading tools

Prize eligibility requires public sharing of your code under a permissive license.

Evaluation

linkkeyboard_arrow_up

Submissions will be evaluated based on **log loss** between your predicted probabilities and the true target values on a private holdout set.

Lower log loss indicates better performance.

Submission File Format

The submission file must be a CSV with the following structure: mint, has graduated

9Wt3N7etKMX9cioTdEJ5S4b8A9nK3M66n9RFVgBGpump,0.734 9q5y2X2P8ZEKTjyXBVcS5q2EZM7HbNV8DURY2qnvqi2f,0.112 HL2di8dcQ7eYDmkcFoZ4zJyHX5SbRZXAJxTegL3JPfx2,0.905

• • •

Where:

- mint is the identifier of the token (from test_unlabeled.csv)
- has_graduated is the predicted probability that the token will graduate

Prizes

linkkeyboard_arrow_up

Total Prizes: 5,625 USDC

- 1st Place 3,000 USDC
- 2nd Place 1,500 USDC
- 3rd Place 750 USDC
- 4th Place 375 USDC

Citation

linkkeyboard_arrow_up

Dmitry Dryomov. Solana Skill Sprint - Memcoin Graduation. https://kaggle.com/competitions/solana-skill-sprint-memcoin-graduation, 2025. Kaggle.

Dataset Description

Big size data is located in a separate dataset:

https://www.kaggle.com/datasets/dremovd/pump-fun-graduation-february-2025

This section explains the structure and purpose of each file provided for the competition. After reading this, you should know:

- What each file contains
- Which ones are needed for training and prediction
- What you're predicting
- What each key column means

Special thanks to On-Chain Divers for providing database for swaps and token creation info.

https://t.me/onchaindiver/76

Files

train.csv

Labeled data with training examples.

Columns:

- mint Token mint address (unique ID)
- slot_min Slot when the token was created
- slot_graduated Slot when the token reached 85 SOL liquidity (if applicable)
- has_graduated Target label (True or False)

test_unlabeled.csv

Same format as train.csv, but without has_graduated or slot_graduated.

Your task is to predict the probability of graduation for each listed token.

You **don't** need to predict slot_graduated

chunk*.csv

Primary behavioral dataset. Each file contains parsed transaction data from the **first 100 blocks** after token mint.

These are the main features to analyze token dynamics. Columns:

- block_time UTC timestamp of the transaction
- slot Solana slot number
- tx_idx Transaction index within the block
- signing_wallet Wallet that signed the transaction
- o direction Transaction type: buy or sell
- base_coin Token involved in the swap
- base_coin_amount Amount of the token bought/sold
- o quote_coin_amount Corresponding amount in SOL
- virtual_token_balance_after Post-transaction virtual token balance (used for calculating virtual LP state)
- virtual_sol_balance_after Post-transaction virtual SOL balance

dune_token_info.csv

Token metadata from Dune Analytics. Columns:

o token_mint_address - Mint address

- decimals Number of token decimals
- name, symbol Token branding
- o token_uri IPFS metadata link
- created_at Timestamp of token mint
- o init_tx Initial transaction hash
- token_info_onchain_divers.csv

Token metadata + deployment stats.

Columns:

- block_time, slot, tx_idx When the token was minted
- o creator Wallet that deployed the token
- mint Token mint address
- bundle_size Number of tokens bundled in the mint tx (if applicable)
- gas_used Gas consumed during creation
- o name, symbol, url Token branding and metadata URI

Prediction Target

Your objective is to predict whether a token **graduates**, defined as reaching \geq 85 SOL liquidity.

This is a binary classification task.

Target label:

- has_graduated = True token graduated
- has_graduated = False token did not graduate

Submission Format

Your submission must be a CSV with the following structure: mint,has_graduated <MINT_ADDRESS>,<PROBABILITY>

Example:

mint,has_graduated 9Wt3N7etKMX9cioTdEJ5S4b8A9nK3M66n9RFVgBGpump,0.742 HL2di8dcQ7eYDmkcFoZ4zJyHX5SbRZXAJxTegL3JPfx2,0.102

Notes

- All modeling must use only data from the first 100 blocks after each token's mint.
- External data is allowed if:

- o Publicly available
- Restricted to pre-mint or ≤ 100-block post-mint data
- Shared in a "External Data" topic