

Sabot

SABot is a machine learning powered automated trading bot that capitalizes on stablecoin instability.

Team Members

Name	email
Quintin Bland	quintinbland2@gmail.com
Kevin Corstorphine	kevincorstorphine@gmail.com
John Gruenewald	john.h.gruenewald@gmail.com
Martin Smith	msmith92663@gmail.com
Yanick Wilisky	yanickw@gmail.com

Project Description / Outline

Based on earlier analysis (see <https://github.com/CAMPSMITH/StableOps>), there may be lucrative arbitrage opportunities with relatively unstable stable coins, e.g. sUSD. This project will focus on arbitrage trading between a very stable coin, like **USDC** or **USDT**, and a more volatile stablecoin, **sUSD**. In order to minimize cost and maximize trading opportunities, a low cost platform and chain with high liquidity of the coins to trade will be used, candidates being explored are Uniswap V2 on Ethereum chain, Uniswap Optimish, or KuCoin.

Questions to Answer

- Can machine learning models be used to implement a profitable automated arbitrage trading bot?
- Which model is more effective?

Datasets to be used

Dataset	Description	Size	Records
CryptoCompare API	1 year of historical hourly data, BTC, Eth, USDC, sUSD, close price and volume		
Uniswap subgraph API	1 year of historical hourly data, sUSD, close price and volume		

Rough breakdown of tasks

- (T2) Uniswap API - python program to prep swap USDC <-> sUSD (K.) or KuCoin / USDT
 - define a trading library with
 - buy function: `buy_susd(<amount of usdc>)` return tuple (`<amount of sUSD bought, gas fee, other fee>`)
 - sell function: `sell_susd(<amount of susd>)` return tuple (`<amount of sUSD received, gas fee, other fee>`)

- file append each transaction to a CSV
 - prefer to get data from uniswap optimism, may need to settle on using uniswap v2
- (T1.5) **get historical hourly sUSD data from uniswap (Y.)**
 - the graph API - python function to get hourly sUSD Uniswap historical data, pair is sUSD and USDC or KuCoin
 - `get_susd_price_data(unixstarthour)` return dataset.
 - <https://thegraph.com/hosted-service/subgraph/uniswap/uniswap-v2?query=Example%20query>
 - unix time online tool: <https://www.epochconverter.com/>
 - Create a scheduler to schedule each hour data (M.)
 - objective is about a year of historical hourly data
 - file append to a CSV
- (T1) skeleton framework for training and testing classifier models (M.)
 - training dataset size
 - fast and slow SMA window sizes
 - Models
- (T1) **Identify historical hourly data for BTC, sUSD, USDC and download data (J.)**
 - CryptoCompare API - hourly BTC, Tehter, USDC, sUSD, USDT, OHLCV in that hour
 - file append to a CSV
- (T1.5) Aggregate data to form coin dataset
 - concat data from historical hourly CryptoCompare API and hourly graph Uniswap API
- (T1) Research finta package to explore more than bollinger that might be good options.
 - pick indicators
 - prototype in a python program (Q.)
- (T1) Notebook to generate trading signals (M.)
 - sweep future period
 - sweep the offset factor
 - Trading signal
 - 0 - no trade
 - 1 - long buy
 - -1 - short buy
- (T1) Augment dataset with analytics
 - Fast SMA for each coin price and volume
 - Slow SMA for each coin price and volume
 - add bollinger curves for each coin
 - add additional Finta indicators
- ----- Training Milestone 7/7 -----
- (T1) Train - classifier (several classifier) slow, fast, training dataset size, mode
- (T1) Backtesting
 - \$\$\$\$ - was it successful
 - classification - accuracy, recall and precision
- (T2) deploy into inference (M.)
 - SageMaker Endpoint
- ----- Trading Milestone 7/11 -----
- (T2) Paper trading (M.) [stretch]
- ----- Presentation 7/14 -----
- **Create Github (M.)**
- **Revise proposal (M.)**