

# Task@RozReturns

**Problem Statement ::** Regime Detection via Unsupervised Learning from Order Book and Volume Data

You might not know about many technical terms and jargon. Feel free to use chatgpt!

**Objective ::** Segment the market into distinct behavioral regimes depending on 3 factors:

1. Trending vs Mean-reverting
2. Volatile vs Stable
3. Liquid vs Illiquid

Using unsupervised learning, based on real-time order book and volume features.

**Data Link ::** [Data](#)

depth20 : Top 20 levels of order book data (price, quantity for bid/ask)

aggTrade : Trade volume data (e.g., per second or per event)

**Step-by-Step Task Breakdown ::**

**Note: This is an example breakdown, you can build differently from this or more on top of this. That would be counted as a plus.**

## 1. Feature Engineering

Extract meaningful, hand-crafted features from each timestamp's data. For example:

Liquidity & Depth Features: Bid/Ask spread =  $\text{ask\_1\_price} - \text{bid\_1\_price}$

Order book imbalance at each level:

$$\text{imbalance\_lvl1} = (\text{bid\_qty\_1} - \text{ask\_qty\_1}) / (\text{bid\_qty\_1} + \text{ask\_qty\_1})$$

$$\text{microprice} = (\text{bid\_1\_price} * \text{ask\_qty\_1} + \text{ask\_1\_price} * \text{bid\_qty\_1}) / (\text{bid\_qty\_1} + \text{ask\_qty\_1})$$

Cumulative depth:

$\text{cum\_bid\_qty} = \text{sum}(\text{bid\_qty\_1 to bid\_qty\_20})$

$\text{cum\_ask\_qty} = \text{sum}(\text{ask\_qty\_1 to ask\_qty\_20})$

Volatility & Price Action:

Rolling mid-price return:  $\log(\text{mid\_t} / \text{mid\_t-1})$

Price volatility: standard deviation of returns in last 10s, 30s

Volume Features:

Volume imbalance (buy volume vs sell volume)

Cumulative volume in last 10s, 30s

VWAP shift (change in VWAP over short windows)

Derived Features:

Sloped depth: quantify how quickly size decays away from top of book

Trade Wipe Level : Average levels wiped in a duration by trades (10 sec, 30sec, etc)

## **2. Data Normalization**

Normalize features (z-score or min-max or any other norm)

Optionally, reduce dimensionality using PCA or any other metric.

## **3. Clustering**

Apply one or more clustering algorithms:

K-means (with elbow plot to choose K)

HDBSCAN (handles noise and non-spherical clusters)

Gaussian Mixture Models (to get soft probabilities)

Compare clustering quality via silhouette score, Davies-Bouldin index or other metrics.

#### **4. Regime Labeling and Analysis**

Assign a “market regime” label to each timestamp.

Analyze characteristics of each regime:

Average volatility

Typical spread and liquidity

Price movement directionality

Name and describe each regime (e.g., "Trending & Liquidity & Stable", "Mean Reverting & Illiquid & Volatile")

#### **5: Visualization (Good Informatic plots)**

Plot regime evolution over time (regime label vs time).

Overlay with price charts or volatility to visually validate.

Use t-SNE or UMAP to visualize clusters in 2D space.

#### **6. Regime Change Insights**

See if there is any correlation when one state of regime changes to another i.e. what is the probability a “Trending & Liquidity & Stable” follows “Mean Reverting & Illiquid & Volatile” etc.

#### **Deliverables:**

1. Ipython notebook or script with code and results

2. Report : 1-2 pages

Explanation of any custom features, clustering metric, clustering functions apart from above.

Clustering results

Regime insights

Visualizations