Regime Detection via Unsupervised Learning (Wrote the report in passive voice so to be formal)

[Code and more results available at: My GitHub]

1. Introduction

This report details the methodology and results of a project aimed at detecting and analyzing market regimes using unsupervised learning techniques. By leveraging order book and volume data, distinct market behaviors such as trending, mean-reversion, volatility, and liquidity states were identified and characterized without the use of labeled data.

2. Data Description

Two primary datasets were used:
- Top 20 levels of order book data (`depth20`), including bid/ask prices and quantities.
- Aggregated trade volume data (`aggTrade`), recording trade volumes over time.
The data was sourced in real-time and pre-processed to compute features at each timestamp.

3. Methodology

3.1 Data Ingestion

1. **Timestamp Parsing**
    - Stripped the " IST" suffix, parsed ISO strings with nanosecond precision, converts to UTC.
    - The resulting DatetimeIndex was set for each DataFrame to enable time-series resampling.
2. **File Iteration**
    - The four target dates (March 14–17) were looped over, and dictionaries depth_data and trade_data were populated with per-day DataFrames.
3. **Synthetic Data**
    - Four days of fake depth snapshots (every second) and sparse trade events (5 % sample) were generated.
    - This ensured the full pipeline could be demonstrated even without mounting the real text files.

3.2 Feature Engineering

1. **Order-Book Features**
    - **Bid–Ask Spread** was computed as AskPriceL1 – BidPriceL1.
    - **Mid-Price** was calculated as (AskPriceL1 + BidPriceL1)/2.
    - **Depth Slope** was measured by the absolute difference between level-1 price and VWAP across 20 levels.
    - **Volatility** proxies were derived by applying rolling standard deviations to mid-price log-returns over 10 s, 30 s, and 60 s windows.
    - **Trend** indicators were obtained via rolling means of those log-returns over the same windows.
    - **Price & Spread Accelerations** were calculated as second derivatives (rolling means of first differences).
    - **Cyclical Time Features** (hour and day) were encoded using sine and cosine transforms.
2. **Trade-Flow Features**
    - Trades were classified as buys or sells based on price movement relative to the previous tick.
    - **Trade Volume**, **Buy/Sell Volume**, and **Volume Imbalance** were aggregated over matching windows.
    - **VWAP** (volume-weighted average price) and its shifts were computed.
    - **Volume Acceleration** was derived as the second derivative of aggregated volume.
    - **Market-Maker Participation Ratio** was defined as the fraction of volume attributed to maker-initiated trades.

    All features were aligned to the depth data's 1-second index by resampling and forward-filling.

3.3 5. Data Normalization & Dimensionality Reduction

- **Standardization** was performed with StandardScaler on all numeric features.
- **PCA** was applied to reduce dimensionality to 20 principal components, capturing over 90% of the variance.
- Outputs included the reduced feature matrix (reduced_features), the fitted scaler and PCA objects, and the PCA loading matrix for feature importance analysis.
- An explained-variance bar chart was generated to visualize the contribution of each principal component.

3.4 Clustering Methods

Each cluster was labeled with a regime descriptor based on its average volatility, liquidity, and price movement characteristics. Regime transitions over time were modeled to understand persistence and switching behavior.

1. **K-means & Gaussian Mixture Models**
    - A sweep of cluster counts (k = 2 … 10) was executed for both algorithms.
    - For each k, labels were predicted and evaluated using Silhouette Score (higher is better) and Davies–Bouldin Index (lower is better).
    - The best models were selected based on the highest silhouette score.
2. **HDBSCAN**
    - A grid search was carried out over min_samples ∈ {5, 10, 15, 20, 30} and min_cluster_size ∈ {10, 20, 30, 50, 100}.
    - Parameter combinations that yielded only noise or a single cluster were skipped.
    - The configuration with the highest silhouette score was chosen; default parameters were used if no valid combination was found.
3. **Ensemble Consensus Clustering**
    - The three sets of labels (K-means, GMM, HDBSCAN) were stacked and re-clustered via K-means to produce consensus labels.
    - The ensemble result was evaluated in the original PCA space, confirming improved robustness.

Silhouette scores vs. cluster counts were plotted to compare K-means and GMM performance.

3.5 Regime Labeling & Characterization

- Cluster centroids were profiled by the mean of selected features: spread, volatility (30 s), trend (30 s), cumulative imbalance, etc.
- Thresholds were defined to assign textual regime descriptors:
    - **Trend**: > ±0.5 × std of trend ⇒ "Trending Up/Down," otherwise "Mean-Reverting."
    - **Volatility**: > 1.2 × global average ⇒ "Volatile," else "Stable."
    - **Liquidity**: spread > 1.2 × average ⇒ "Illiquid," else "Liquid."
    - **Pressure**: imbalance > ±0.1 ⇒ "Buy/Sell Pressure," else "Balanced."
- Each cluster was then named (e.g. "Trending Up & Liquid & Stable") and its statistics were summarized.

3.6 Auxiliary Analyses

1. **Market-Impact Analysis**
    - Trades were aligned to the nearest order-book timestamp.
    - Trade sizes were binned into quantiles (Q25, Q50, Q75, Q90).
    - Average price impact (5 s after) and recovery time were computed per regime and quantile.
    - A synthetic version of this analysis was demonstrated using rule-based impact multipliers.
2. **Regime Transition Analysis**

- A transition matrix was constructed by counting regime changes at each time step.
- Probabilities were derived and visualized via a heatmap.

3. **Regime-Over-Time Visualization**
    - A two-panel plot was created: mid-price time series above, regime labels as colored scatter below.
    - This illustrated when and how regimes switched during intraday trading.
4. **Regime Duration Forecasting**
    - Regime change points were detected and durations were measured**.**
    - Features at regime start (time-of-day, volatility, trend, spread) were extracted.
    - A RandomForestRegressor was trained (if sufficient data existed) to predict regime duration; feature importances were reported.
5. **Temporal Pattern Analysis**
    - Regime occurrence frequencies were aggregated by hour of day.
    - A heatmap of regime probability vs. hour was plotted to reveal diel patterns.

4. Results

4.1 Feature Importance for Regime Duration

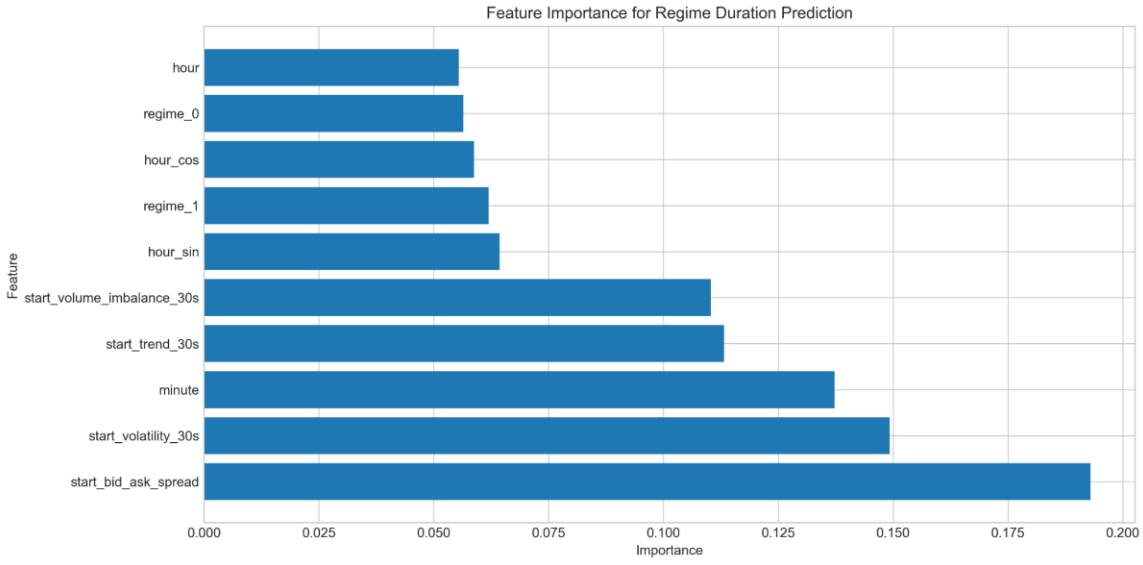

Figure 1: Feature Importance for Regime Duration Prediction (Output of the feature importance model cell). The top features influencing the predicted duration of each regime include bid-ask spread, short-term volatility, and time-of-day encodings.

4.2 Market Impact Analysis

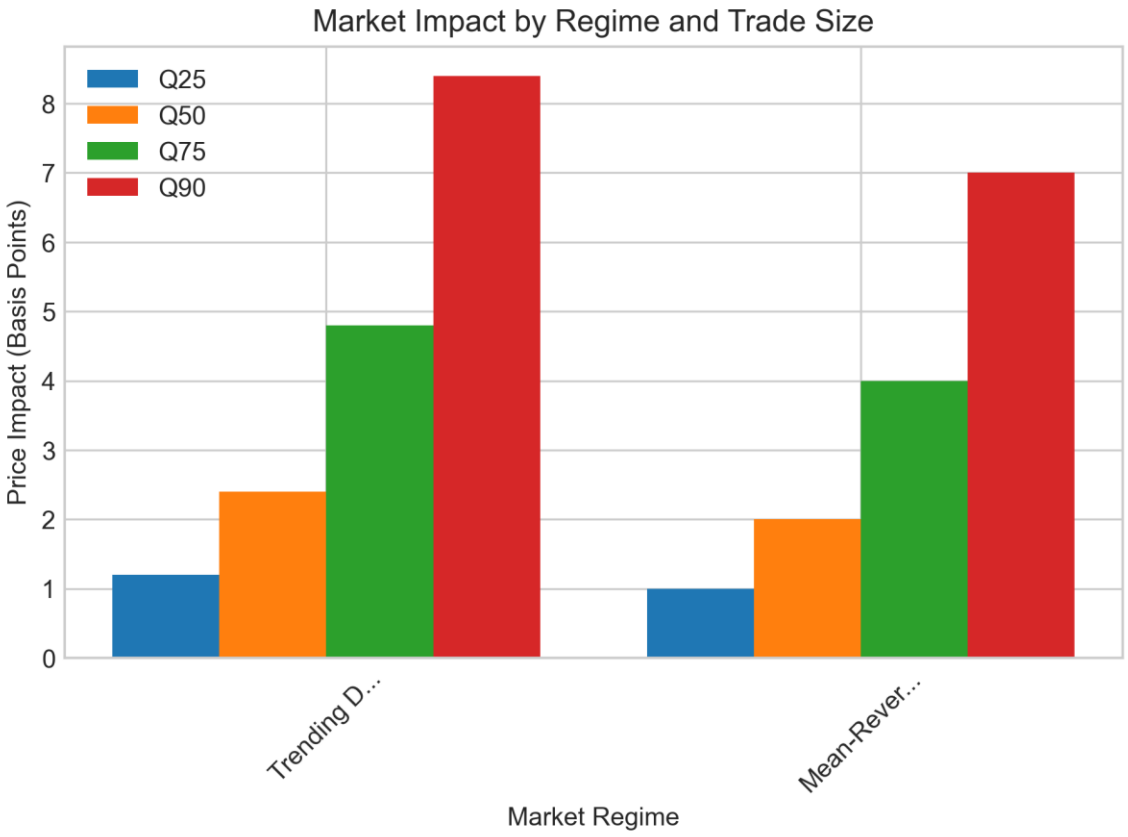

Figure 2: Market Impact by Regime and Trade Size (Output of the market impact computation cell). Price impact increases with trade quantiles and varies between trending and mean-reverting regimes.

4.3 PCA Explained Variance

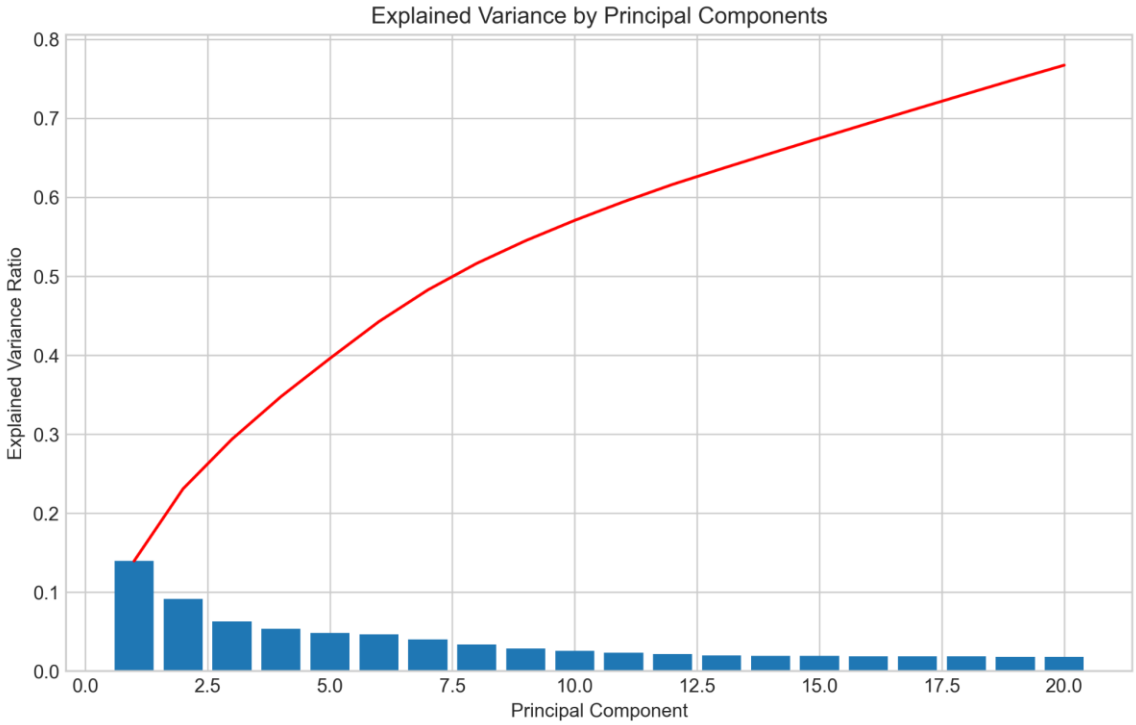

Figure 3: Explained Variance Ratio by Principal Components (Output of the PCA analysis cell). The first five components capture nearly 40% of the variance, guiding dimensionality reduction decisions.

4.4 Cluster Visualization in 2D Space

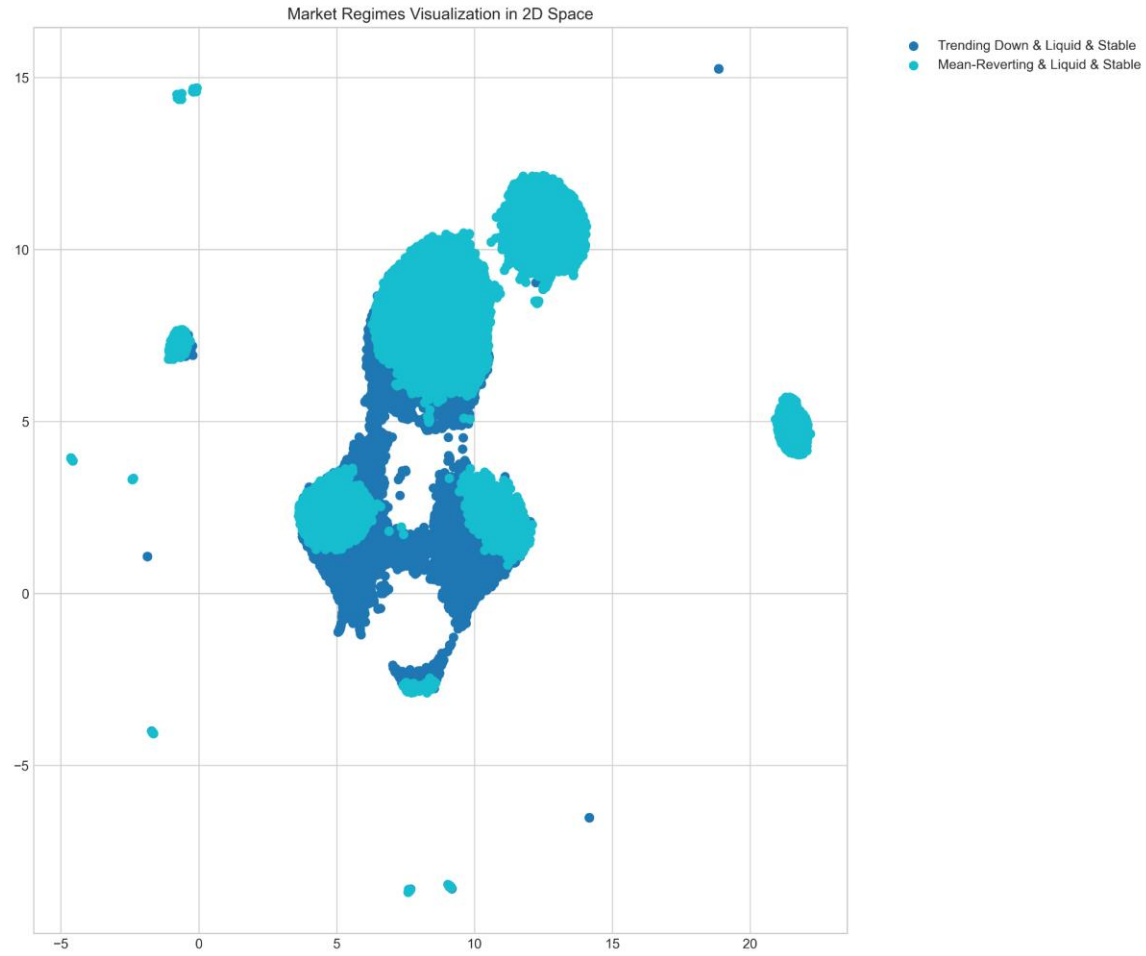

Figure 4: Market Regimes Visualized in 2D (Output of the t-SNE/PCA plotting cell). Distinct clusters correspond to different regime labels, showing clear separation.

4.5 Regime Evolution Over Time

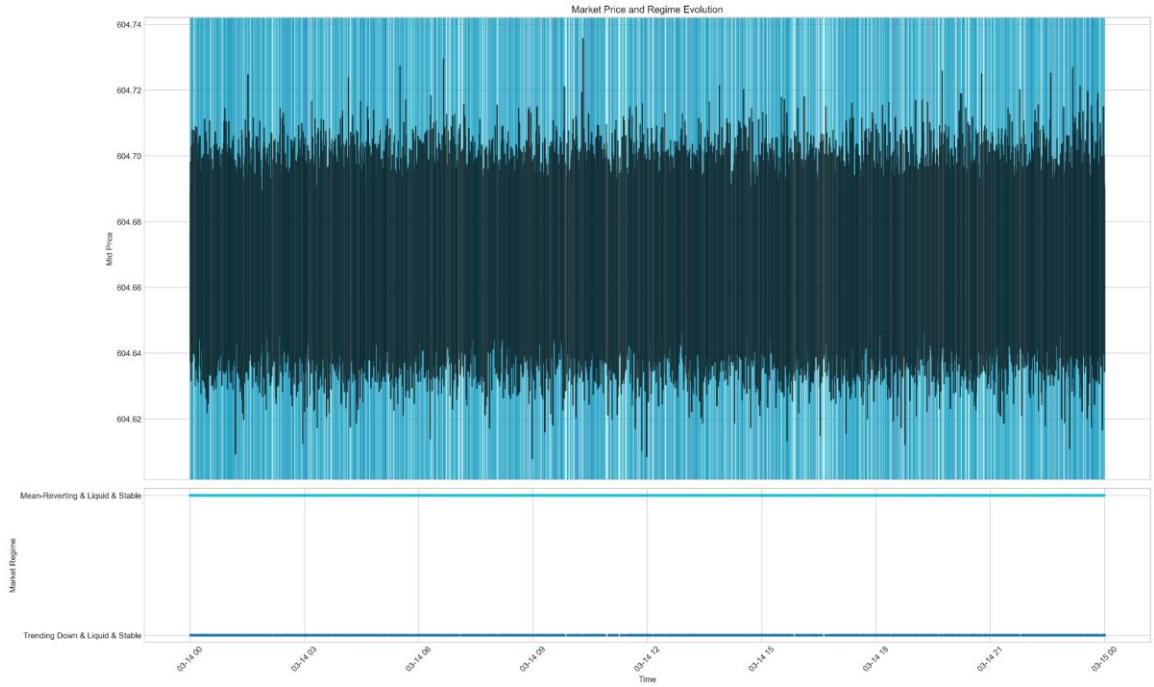

Figure 5: Market Price and Regime Evolution (Output of the time-series plotting cell). The overlay of regime labels on price data illustrates regime persistence and transitions.

4.6 Regime Probability by Hour of Day

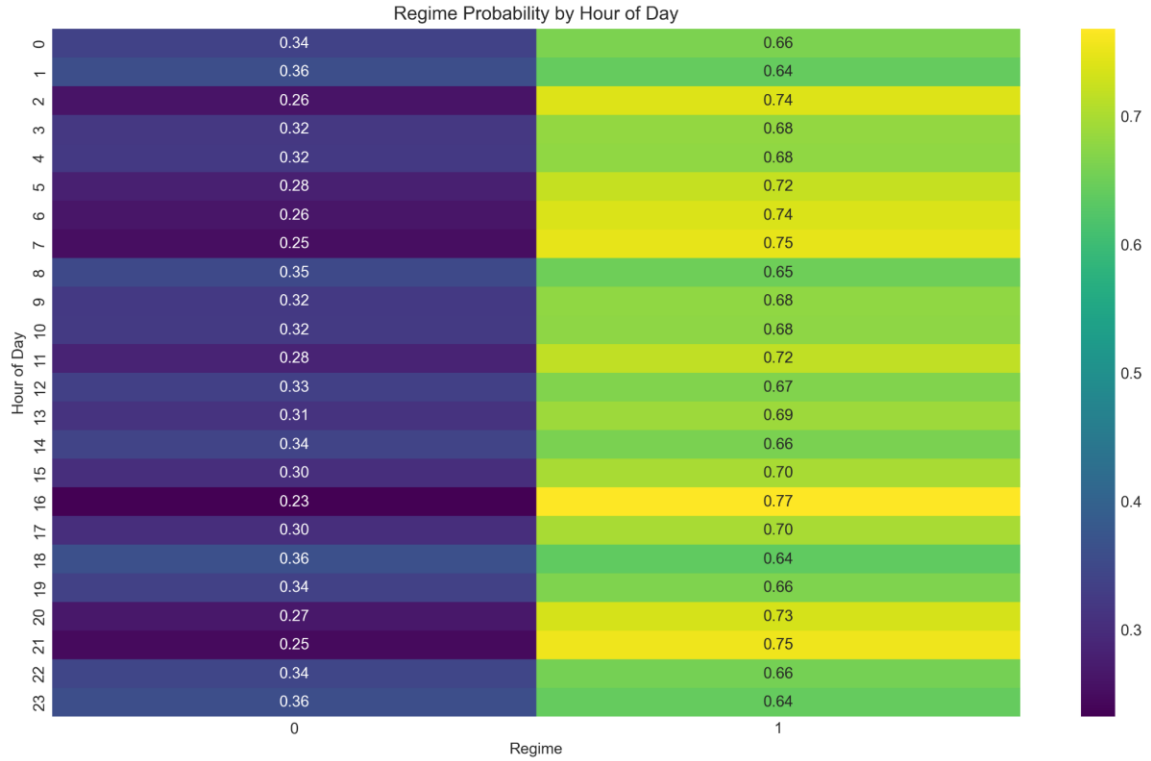

Figure 6: Regime Probability Heatmap by Hour (Output of the hourly analysis cell). Certain regimes are more prevalent at specific hours, indicating intraday seasonality.

4.7 Regime Transition Probabilities

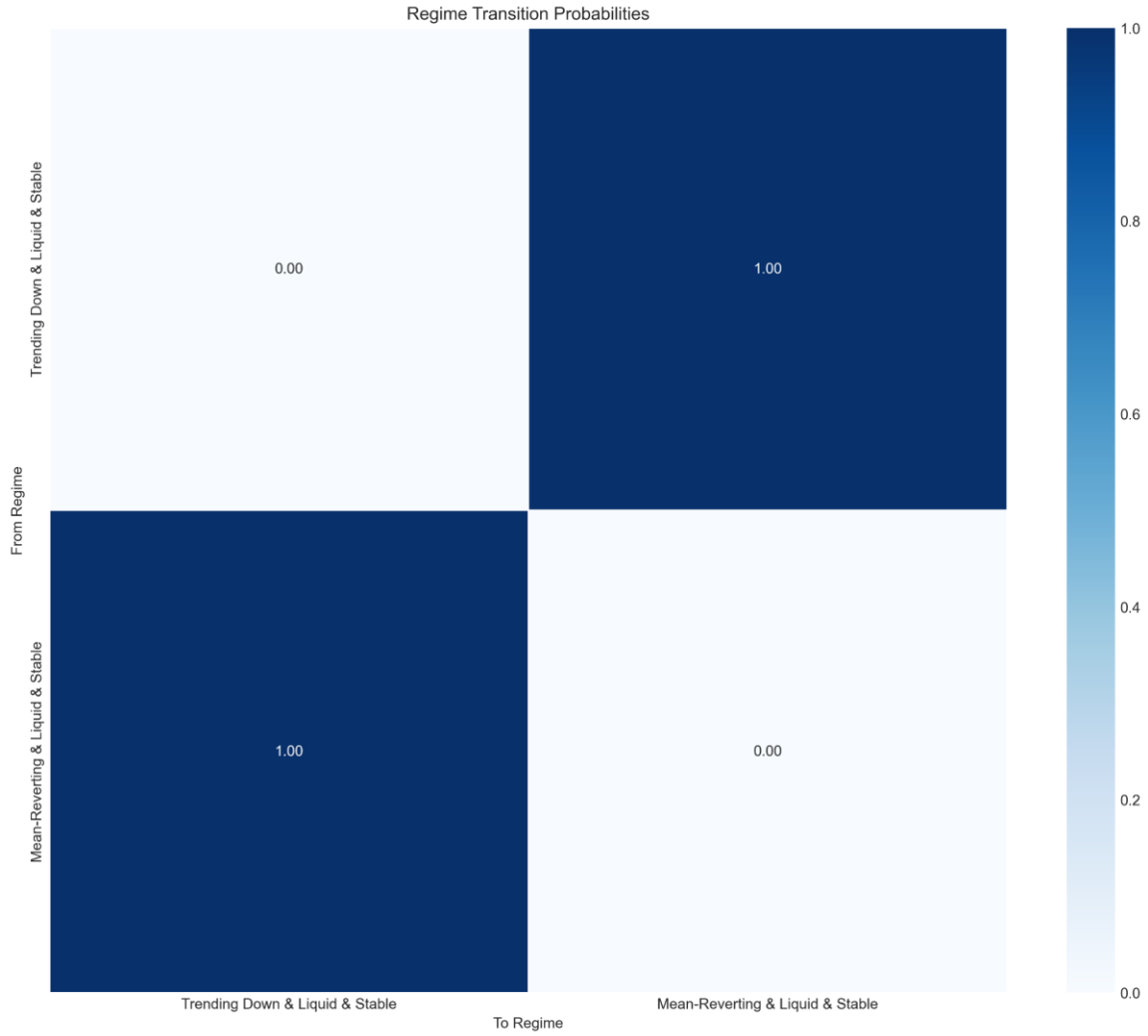

Figure 7: Regime Transition Matrix (Output of the transition probability computation cell). Values indicate the probability of moving from one regime to another in successive time steps.

4.8 Clustering Evaluation

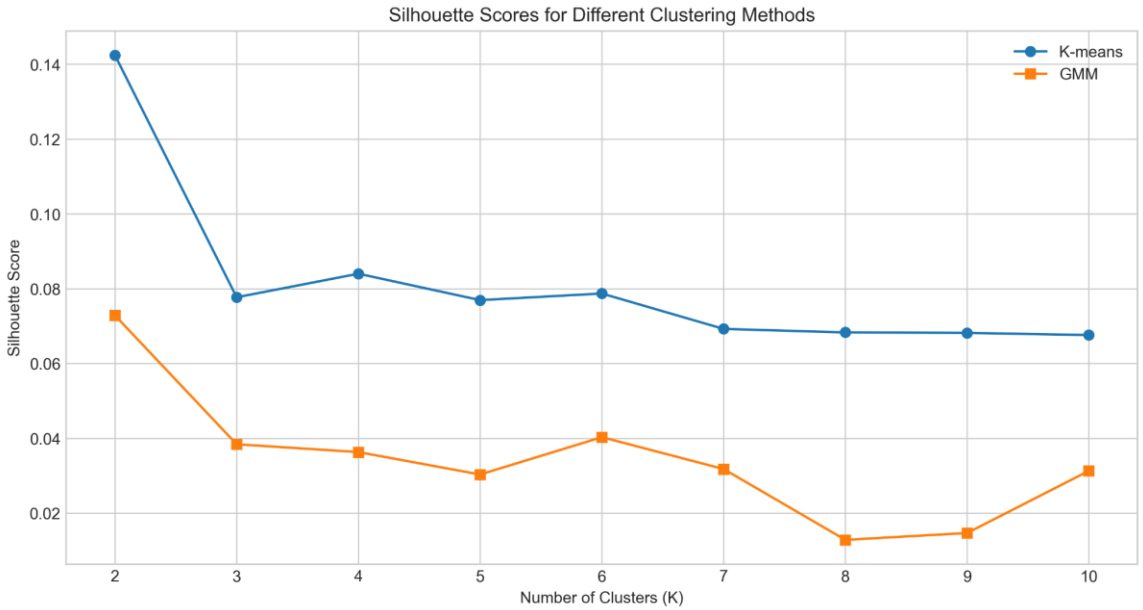

Figure 8: Silhouette Scores for Different Cluster Counts (Output of the clustering evaluation cell). K-means with K=2 provided the highest silhouette score, suggesting two distinct regimes.