Options Market Regime Analysis: Implied Volatility Prediction and Clustering

Chris Zhang, George Smith, Ayansola Akanmu

1. Introduction

Options pricing and trading rely heavily on accurate volatility estimations, yet traditional models like Black-Scholes assume constant volatility, which rarely reflects real market conditions. We see an opportunity to address this critical challenge of understanding and predicting volatility patterns in options markets, which is fundamental to risk management and trading strategy development. We were motivated to see if deep learning models could be used to identify opportunities in this market.

This project aims to provide actionable insights for options traders, risk managers, and portfolio optimization systems by combining predictive modeling with pattern recognition.

In this project, we will develop a comprehensive machine learning framework for analyzing options markets through two interconnected components: (1) implied volatility forecasting using deep learning time series models, and (2) market regime identification through unsupervised clustering techniques. For the supervised component, the LSTM (Long-short term memory) deep learning model is used to capture temporal dependencies in implied volatility. For the unsupervised part, K-means clustering and Ward Linkage are employed to identify distinct market regimes.

We combine deep-learning based volatility forecasting with unsupervised market regime detection. Among the supervised models tested (LSTM, Random Forest, and Ridge Regression), the LSTM model achieved the highest predictive accuracy (RMSE = 0.1129, $R^2 = 0.42$), outperforming both traditional machine learning approaches.

For the unsupervised learning portion, K-Means provided a more stable and interpretable regime cluster (ARI = 0.61). Integrating clustering results into the forecasting models did not show significant improvements, probably due to the fact that the underlying features that determine the clusters were also fed into the LSTM model.

2. Related Work

Several studies have explored market regime detection and volatility prediction using machine learning.

"Clustering Market Regimes Using the Wasserstein Distance" (B. Horvath, Z. Issa, and A. Miguruza) introduced Wasserstein k-means, a modification of classical k-means that clusters market returns based on the Wasserstein distance between their distributions. Our work instead applies standard k-means and hierarchical clustering (Ward linkage) to volatility-surface features, using Wasserstein distance only as one of the evaluation metrics.

"Detecting Multivariate Market Regimes via Clustering Algorithms" (McGreevy et al.)
McGreevy et al. proposed a regime detection framework using k-means clustering on multivariate time series, incorporating distributional metrics like Wasserstein distance and Maximum Mean Discrepancy. Their work focused on regime identification for portfolio design and pairs trading, emphasizing unsupervised learning for structural market shifts. Our approach builds on this by applying clustering to volatility-surface features rather than raw returns, and complements unsupervised regime detection with supervised models (LSTM, Ridge Regression, Random Forest) to forecast implied volatility, thereby bridging regime identification with predictive modeling.

"A Two-Step Framework for Arbitrage-Free Prediction of the Implied Volatility Surface" (Zhang, Li & Zhang, 2021)
This study proposes a two-step approach for forecasting the implied volatility surface (IVS) while ensuring arbitrage-free conditions. First, it extracts latent features using PCA, variational autoencoders, and sampling methods. Second, it reconstructs the IVS using a deep neural network that incorporates constraints to eliminate static arbitrage. Our work differs by focusing on regime clustering and volatility forecasting using both unsupervised

(KMeans, Ward) and supervised models (Ridge, RF, LSTM), without enforcing arbitrage constraints, instead we emphasize on regime stability and temporal dynamics.

3. Data Sources

This project uses market and options data from the **OptionMetrics IvyDB US** and **CRSP** databases accessed via the **WRDS (Wharton Research Data Services)** API. The primary dataset, *optionm.opprcd{YYYY}* (e.g., *opprcd2023*), provides daily end-of-day option prices and implied volatilities for U.S. equities. Supplementary tables include *optionm.secnmd* for security identifiers and *crsp.dsf / crsp.msenames* for stock prices and name histories. Data was retrieved via WRDS SQL Queries API, and processed with Polars DataFrames. Some of the key variables include ***impl_volatility, symbol, underlying_prc, strike_price, theta, open_interest, and vega*** (see Appendix A for full list of variables and their descriptions).

We chose to query the WRDS data source from 2005 – 2023, filtering only for SPY - SPDR S&P 500 ETF Trust and QQQ - Invesco QQQ Trust. The SPY dataset contains over 50M records, while the QQQ dataset contains over 20M records over the 19 years.

4. Feature Engineering

4.1 Data Acquisition and Initial Processing

We extracted, filtered, and transformed options and stock data from the WRDS database for SPY and QQQ spanning 2005 to 2023. The data acquisition process was executed independently for each ticker-year combination. For each ticker, we retrieved associated Security IDs (secid) from optionm.secnmd, queried options data from the annual tables optionm.opprcd{YYYY} within the specified date range, and joined with daily stock price and volume data from crsp.dsf on date. This ticker-specific processing approach allowed us to apply consistent feature engineering methodologies across both SPY and QQQ datasets before combining them into a unified training dataset. The final merged dataset, encompassing both tickers across the full 18-year period, served as the foundation for both our supervised learning task (forecasting implied volatility 30 days forward) and our unsupervised machine learning objective (identifying distinct market regimes through clustering).

4.2 Preprocessing and Quality Filtering

The preprocessing phase implemented rigorous quality controls to ensure that only liquid, actively traded options contracts entered the modeling pipeline. We restricted analysis to transactions with volume > 0, as zero-volume records represent stale quotes rather than actual market activity. To eliminate extreme outliers, we computed a volume threshold at the 95th percentile and filtered out observations exceeding this limit, removing thinly-traded exotic contracts and potential data errors. Additionally, we excluded any rows where any of our features were null, as these observations could not be predicted, and we assumed that missing options data often indicates illiquid or problematic contracts that would introduce more noise if artificially completed. Following these filtering steps, we applied z-score standardization across all quantitative features to ensure comparability and avoid magnitude biases.

4.3 Strategic Downsampling for Computational Feasibility

Some of the following machine learning tasks were met with heavy computational limitations when processing the complete WRDS options dataset. To accommodate the memory and processing constraints of the Great Lakes HPC, we implemented a strategic downsampling approach that reduced the training dataset to approximately 15 million rows and the test set to 7 million rows. The semi-intelligent selection strategy first chose at least 1 data point from each market date, guaranteeing representation for each day. Then, the quota was reached by randomly seeding the

rest of the data points. These randomly downsampled datasets were computed uniquely each time, allowing us to generalize the dataset for our model.

4.4 Core Feature Engineering: Temporal and Structural Variables:

30-Day Forward Implied Volatility Target Variable:

- To create our supervised learning target variable, we computed the 30-day forward implied volatility by shifting the current implied volatility forward by 30 days within each security group:
- `iv_30d = impl_volatility.shift(-30).over('secid').`

Fibonacci-Based Price Momentum Features

- We generated a comprehensive set of price difference features using Fibonacci sequence intervals – 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, and 233 days – to capture momentum patterns in the underlying asset.
- For each Fibonacci lag 'd', we computed
  `price_diff_{d} = prc - prc.shift(d).over('secid')`
- These features capture price dynamics across different trading horizons, providing the model with temporal context about whether markets are trending, mean-reverting, or experiencing regime transitions.

At-the-Money Implied Volatility (atm_iv)

- At-the-money implied volatility serves as a critical benchmark for overall market volatility expectations, as ATM options are typically the most liquid and price-efficient contracts in each option chain.
- To compute this feature, we identified the strike price closest to the current underlying stock price for each security and date combination by minimizing the absolute difference |strike_price - prc|. We then selected the corresponding implied volatility value from this near-ATM option:
- `atm_iv = impl_volatility.where(strike_price ~ prc).group_by(['secid', 'date']).`

Volatility Skew

- Volatility skew quantifies the asymmetry in implied volatility across the wings of the options distribution, reflecting the market's difference in pricing of downside protection versus upside participation.
- We computed skew directly using standard option delta conventions by identifying 25-delta put and call implied volatilities. The skew metric is then calculated as the difference between these two standardized volatility points: `skew = iv_put25 - iv_call25`.
- Positive skew values indicate that 25-delta puts trade at higher implied volatility than 25-delta calls, signaling market participants' willingness to pay premium for downside protection – a phenomenon particularly pronounced during periods of elevated uncertainty, following market crashes, or ahead of significant macroeconomic events.

Volatility Curvature

- Volatility curvature captures the convexity of the implied volatility surface, measuring how the wings of the distribution are priced relative to at-the-money options.
- After computing the 25-delta put and call implied volatilities as described above, we calculated curvature as the average of the wing volatilities minus the at-the-money volatility:
- `curvature = ((iv_put25 + iv_call25) / 2) - atm_iv`.
- This formulation quantifies the vertical distance between the center of the volatility smile and its midpoint at the wings, with higher curvature values indicating more pronounced smiles where tail risk is being priced significantly higher than central outcomes.

- Elevated curvature typically emerges during market stress periods when investors seek protection against extreme movements in either direction, manifesting as steeper volatility surfaces where out-of-the-money options command substantial premium over at-the-money contracts.

4.5 Supervised Learning Specific Feature Engineering

Volume-Delta Product

- The volume-delta product captures the interaction between trading activity intensity and the options contract's sensitivity to underlying price movements, providing insight into market maker hedging flows that often presage volatility regime shifts.
- We computed this feature as `vol_delta_product = volume * abs(delta)`, where higher values indicate significant hedging activity in directionally sensitive contracts

Moneyness Ratio

- Moneyness represents the fundamental relationship between an option's strike price and the underlying asset price, serving as the primary determinant of an option's intrinsic and extrinsic value components.
- We calculated this ratio as `moneyness = strike_price / prc`,
- Values near 1.0 indicate at-the-money options, values below 1.0 represent in-the-money calls (out-of-the-money puts), and values above 1.0 indicate out-of-the-money calls (in-the-money puts).

10-Day Rolling Standard Deviation of Implied Volatility

- To quantify recent volatility-of-volatility patterns – the tendency for implied volatility itself to exhibit clustering and regime persistence, we computed a rolling standard deviation across a 10-day lookback window:
- `iv_rolling_std = impl_volatility.rolling(window=10).std().over('secid').`
- Such volatility-of-volatility spikes typically precede regime transitions and major market events

5-Day Volume Moving Average

- Raw volume data contains substantial intraday noise from sporadic large transactions and market microstructure effects that can obscure sustained changes in market participation. To smooth these fluctuations and identify genuine trends in trading activity, we implemented a 5-day simple moving average:
- `volume_ma5 = volume.rolling(window=5).mean().over('secid').`

Final Features:

Full definitions appear in Appendix A

5. Unsupervised Learning

This section outlines the unsupervised learning pipeline used to identify latent volatility regimes within the options dataset. It describes the modeling workflow, methods used, evaluation metrics, and sensitivity analyses conducted to ensure robustness and interpretability.

5.1 Methods Description

Workflow Overview

This workflow uncovers latent market regimes from options data by analyzing implied volatility structures and Greeks patterns using unsupervised learning techniques. The methodology proceeds through five distinct phases: feature selection, dimensionality reduction, standardization, clustering, and validation.

Feature Selection: We initially constructed the unsupervised learning pipeline using a set of **16 features**, including implied volatility metrics, Greeks (delta, theta, vega), moneyness, volume indicators.price_diff, etc. To improve clustering stability and reduce noise, we later refined the feature set to **11 core variables** based on ablation testing and PCA-based variance analysis. This reduction preserved key volatility dynamics while minimizing redundancy, enabling more consistent regime identification across clustering methods.

Dimensionality Reduction: To address the curse of dimensionality and multicollinearity inherent in the selected feature set, we applied Principal Component Analysis (PCA) to construct a lower-dimensional representation while preserving the essential variance structure.

Standardization: Prior to clustering, we applied z-score standardization to both the raw and PCA-transformed feature matrices to ensure that variables with different natural scales received equal consideration during distance calculations.

Clustering: Following feature preparation, we applied two unique clustering algorithms to identify distinct market regimes within the standardized feature space.

Validation: To ensure that the identified clusters represented economically meaningful market regimes rather than spurious statistical artifacts, we conducted comprehensive validation analyses, both visually and quantitatively.

Modeling Approaches

Two complementary unsupervised methods were implemented:

1. **K-Means (Euclidean) with Wasserstein Evaluation – Primary Model**
    - Partition-based clustering on standardized features.
    - Explored $k \in \{2,\ldots,20\}$, using inertia, Silhouette, Calinski–Harabasz (CH), Davies–Bouldin (DB), and evaluated cluster compactness using a Wasserstein-based dispersion metric, computed via *scipy.stats.wasserstein_distance* between feature distributions and their cluster centroids.
    - Hyperparameters were tuned by scanning ranges and selecting via ensemble voting and stability checks (bootstrapped Adjusted Rand Index, ARI with weights $\alpha = \beta = 0.25$, 5 bootstraps, sample fraction 0.7).
    - Chosen for scalability and geometry-aware assessment of volatility structures.
2. **Ward Hierarchical Clustering (Ward Scan) – Secondary Model**
    - Constructed a Ward-linkage dendrogram for hierarchical interpretability and nested volatility states.
    - Scanned distance thresholds $t \in \{30,40,\ldots,100\}$, evaluating Silhouette, CH, DB, and evaluated compactness using mean L1 distance to cluster centroids.
    - Selected consensus threshold by majority vote; stability optimized via ARI ($\alpha = \beta = 0.25$, 4 bootstraps, sample fraction 0.5).
    - Chosen for its variance-minimizing linkage and ability to reveal nested low-/mid-/high-volatility regimes.

Justification

K-Means provides flat partitions and scalability, while Wasserstein evaluation adds distributional sensitivity beyond Euclidean geometry. Ward Scan complements this with hierarchical interpretability and variance minimization, enabling discovery of nested volatility states. Together, these methods offer robust and complementary perspectives on latent market regimes.

5.2 Unsupervised Evaluation

Overall Results Reporting

We evaluated clustering quality and stability using a diverse set of metrics:

- Silhouette: Measures cohesion and separation of clusters.
- Calinski–Harabasz (CH): Ratio of between-cluster to within-cluster dispersion.
- Davies–Bouldin (DB): Lower values indicate better compactness and separation.
- Wasserstein-based dispersion: Captures distributional geometry beyond Euclidean distance.
- Adjusted Rand Index (ARI): Quantifies clustering stability across bootstraps.

Summary of Best Models

Due to computational constraints, Ward linkage could not be applied to the full dataset (~50M records). To enable comparison, we evaluated both methods on a sampled subset (~30K records) alongside full-scale K-Means results.

Table 5.2.1: Comparative evaluation of clustering methods across full dataset and sampled subset. PCA applied at 95% variance where indicated.

Metric	**K-Means (Full Dataset, PCA=0.95**	**K-Means (Sampled, No PCA)**	**Ward Linkage (Sampled, PCA=0.95)**
Symbol	SPY	SPY	SPY
Dataset Size	~50m	~30k	~30k
PCA (0.95)	True	False	True
Selection Parameter	k* = 12	k* = 16	t* = 70 (k = 29)
Silhouette	0.2479	0.1232	0.1000
Calinski–Harabasz	3.56×10^6	2574.53	301.78

Davies–Bouldin	2.089	2.759	1.507
Wasserstein / Compactness	0.658	0.513	0.707
Stability (ARI)	0.608	0.501	0.403

An interactive evaluation dashboard is available for visual inspection (see Appendix B)

Figure 5.2.1: Evaluation Metrics vs Number of Clusters (k) for K-Means

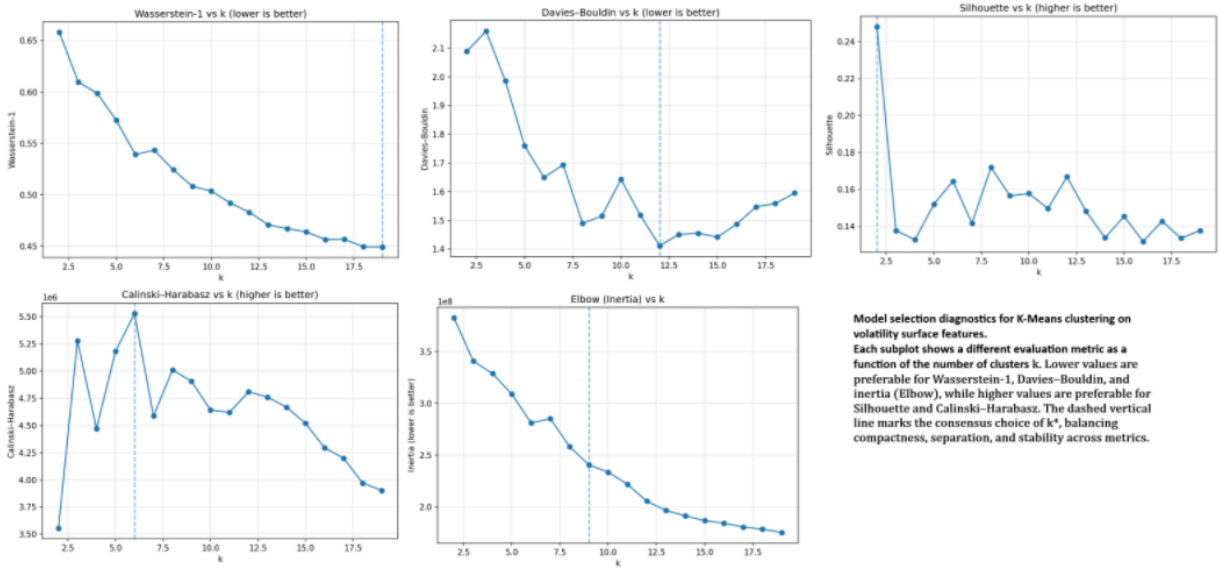
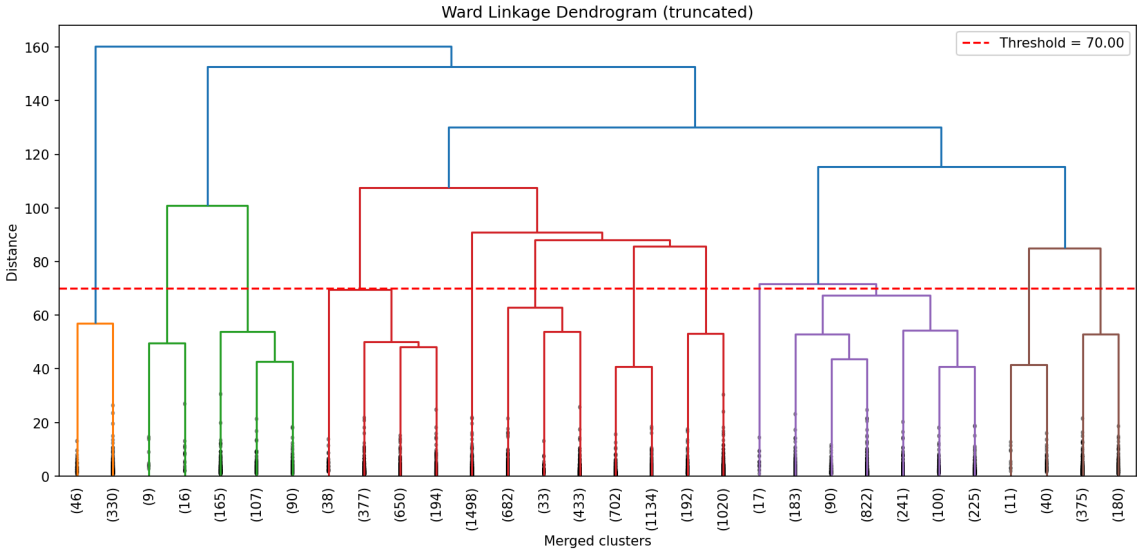


Figure 5.2.2: Ward dendrogram (sampled subset).



Interpretation
K-Means on the full dataset achieved higher Silhouette and stability (ARI ~ 0.61) compared to the sampled run (ARI ~ 0.50), indicating that larger data improves regime separation. Ward linkage, limited to the sampled subset, showed lower Silhouette but better compactness (L1-based) than sampled K-Means, reinforcing its strength in hierarchical interpretability despite computational constraints. See Appendix F for complete Evaluation Metrics of the clustering methods under different configurations.

Sensitivity Analysis

- **Cluster and threshold stability:** K-Means stable for $k \in [7, 17]$; Ward Scan consistent for $t \in [40, 90]$.
- **Feature scaling and PCA:** Standardization prevented vega/gamma dominance. PCA at 95 % variance gave the best compactness-accuracy trade-off.

Extended Evaluation Addendum

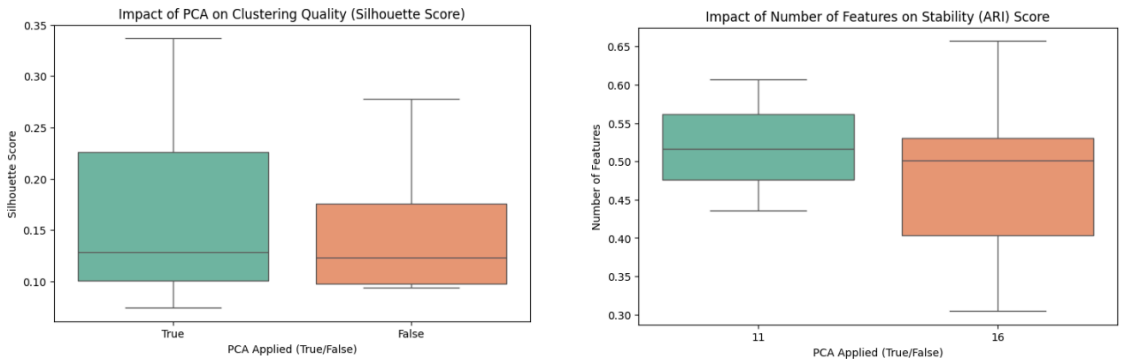
A. Ablation Tests – Feature drop experiments showed ARI shift ~ 4 % and Silhouette shift ~ 0.14 after removing temporal and volume features. Final feature set = 11 predictors.

B. Stability-Driven Tuning – K-Means optimal ($\alpha = \beta = 0.25$, ARI = 0.61 at $k = 12$). Ward Scan optimal ($\alpha = \beta = 0.25$, ARI = 0.40 at $t = 70$).

C. Learning Curves (ARI vs Sample Size) – ARI increased monotonically, saturating around 0.61 at full dataset; smaller samples (< 75 %) underrepresented regime diversity.

D. PCA Threshold Optimization – Compared variance thresholds {0.85, 0.95}. Best compactness and stability at **0.95 variance threshold** with **12 principal components** (Silhouette = 0.26, DB = 2.05, ARI = 0.61).

Figure 5.2.3: Impact of Dimensionality and Feature Selection on Clustering Quality and Stability



6. Supervised Learning

6.1 Methodology

Model Selection

Ridge Regression (Linear/Regularized Method):
Ridge regression served as the baseline model, to establish interpretable linear relationships between features and implied volatility. Ridge regression in particular prevents overfitting in high-dimensional feature spaces common in

financial data, and provides coefficient estimates that quantify the direct impact of each feature, helping understand which market factors most influence volatility predictions.

Random Forest (Tree-Based Ensemble Method):
Random Forest was selected as the non-parametric ensemble approach to capture complex non-linear relationships and feature interactions without assumptions about data distribution. This method excels at handling mixed data types (continuous Greeks, categorical market conditions) and provides built-in feature importance rankings crucial for understanding options feature importance. The ensemble nature reduces overfitting while the tree-based structure naturally handles outliers common in financial markets during stress periods.

LSTM Neural Network (Deep Learning/Sequential Method):
LSTM networks were implemented to leverage the temporal nature inherent in options pricing time series data. LSTMs can learn complex temporal relationships between past market states and future volatility, making them particularly suitable for the sequential nature of financial time series where market regime transitions significantly impact pricing.

Hyperparameter Tuning and Cross-Validation

Hyperparameter optimization was conducted using the Great Lakes computing cluster to handle the computational intensity of grid search across multiple model families. This was done in conjunction with time-series cross-validation, enabling a thorough evaluation of each model.

In order to prevent look ahead bias, we derived 6 folds based on 2 time anchors. One between 2005 and 2013, and the second from 2014 to 2023. Each fold consisted of incrementing 2 year training datasets (2005-2007, 2005-2009…), and 1 test year following directly after the training period. This approach was critical for financial data to avoid data leakage and ensure realistic performance estimates that reflect actual trading scenarios.

All hyperparameter grid optimization was conducted within the temporal cross-validation framework to ensure hyperparameter choices generalized across different market periods captured in the WRDS dataset.

6.2 Summary of Results

We chose RMSE, R^2, and MAE as evaluation metrics for our implied volatility prediction models, reflecting standard statistical practice in regression analysis. Root Mean Squared Error (RMSE) serves as the primary metric because it heavily penalizes large prediction errors through its squaring mechanism, which aligns perfectly with the risk management priorities. Mean Absolute Error (MAE) complements RMSE by providing a more interpretable, linear measure of average prediction error that treats all deviations equally. Finally, R^2 quantifies the proportion of variance in implied volatility explained by our model features, providing a standardized measure of predictive power.

Table 6.2.1 Hyperparameter Tuning Over CV Grid Results:

Model	Best RMSE	Best R^2	Best MAE	Best Hyperparameters
Ridge Regression	0.1512	0.1387	0.0623	alpha=1.0, solver='saga'

Once the best model was determined to be the LSTM, the best hyperparameters were used to train an LSTM model on two datasets, one with clustering and one without.

6.3 In-Depth Evaluation

Feature Importance and Ablation Analysis

Feature Group Analysis

The systematic removal of feature groups revealed distinct patterns:

- Price Features: Caused R^2 drops of 0.327 (clustered) vs. 0.197 (non-clustered), indicating that clustering amplifies the importance of price relationships
- Volatility Features: Showed similar impacts across models (0.280 vs. 0.269), suggesting volatility remains fundamentally important regardless of regime identification
- Volume Features: Demonstrated minimal impact in both models (0.004 vs. 0.004), indicating that trading activity metrics provide limited predictive value for IV

No Clusters	With Clustering Results

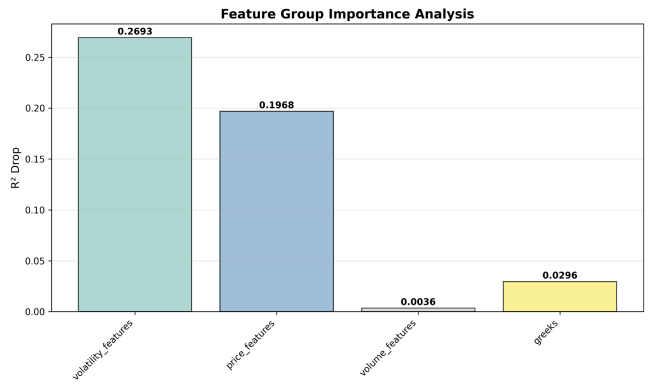
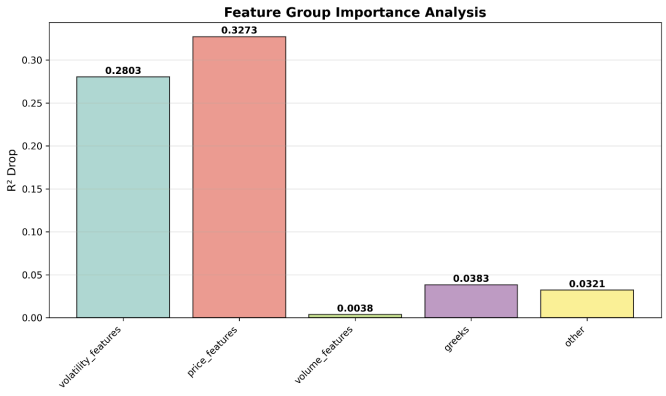

	

Figure 6.3.1: Feature Group Importance Analysis for the Unclustered Dataset	***Figure 6.3.2: Feature Group Importance Analysis for the Clustered Dataset***

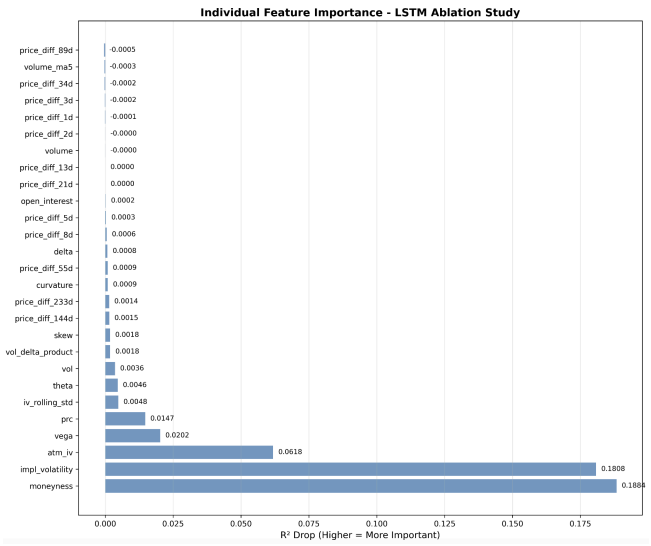


Figure 6.3.3: Individual Feature Group Importance Analysis for the Unclustered Dataset

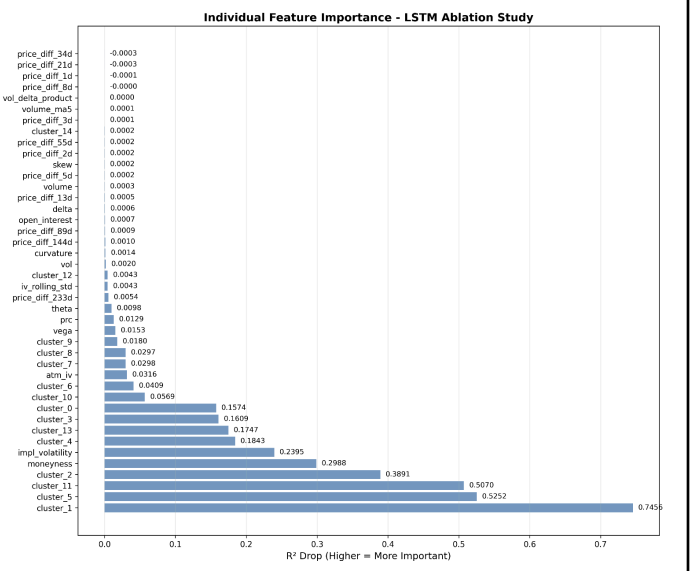


Figure 6.3.4: Individual Feature Group Importance Analysis for the Clustered Dataset

Individual Feature Analysis

Price-Based Features: Moneyness emerged as the single most important traditional feature, with removal causing R^2 drops of 0.299 (clustered model) and 0.188 (non-clustered model). This confirms that the relative position of strike price to underlying asset price remains fundamental to options pricing across all market regimes.

Volatility Features: Implied volatility itself showed importance scores of 0.471 (clustered) and 0.343 (non-clustered), while the broader volatility feature group caused R^2 drops of 0.280 (clustered) and 0.269 (non-clustered). The relatively consistent impact across both models indicates that fundamental volatility metrics remain crucial regardless of regime identification.

Greeks Contributions: The Greeks feature group showed modest but consistent importance (R^2 drops: 0.038 clustered, 0.030 non-clustered), with vega exhibiting the strongest individual impact. Interestingly, the clustering model showed reduced sensitivity to individual Greeks, suggesting that cluster features partially subsume the market dynamics that Greeks attempt to capture. However, vega, the Greek for volatility, remained one of the most important Greeks.

Other: The most impactful cluster features were cluster_1 (R^2 drop: 0.746, importance score: 1.466), cluster_5 (R^2 drop: 0.525, importance score: 1.033), and cluster_11 (R^2 drop: 0.507, importance score: 0.997). These clusters appear to capture distinct market regimes: referencing figure 5.2.3, we can see that cluster 1 is characterized by average z-scores across the board; cluster 5 with a heavy emphasis on high volume regimes, and cluster 11 likely detailing options which have deep theta decay. Interestingly, cluster 9 which has a strong distinction on moneyness, contributes significantly less than the actual moneyness parameter.

<u>Identity Tradeoffs</u>

The most significant constraint emerged from computational limitations when processing the complete WRDS options dataset. To accommodate the memory and processing constraints of the Great Lakes cluster, we implemented the downsampling approach mentioned in Section 4.3. While this downsampling employed a semi-intelligent selection strategy that ensured representation from each trading day, it still introduced sampling bias

by capturing only a subset of the full implied volatility distribution for each date. This methodology likely missed critical outlier observations that would have been essential for training the model to handle extreme market conditions.

The temporal prediction framework represented another fundamental tradeoff that significantly limited our model's potential effectiveness. Rather than leveraging the LSTM's inherent capacity for sequential prediction, we constructed our target variable (iv_30d) by simply shifting the current implied volatility forward by 30 days, essentially creating a regression problem rather than a true time-series forecasting task. This approach failed to capitalize on one of the LSTM's most powerful capabilities: its ability to generate forward-looking predictions across extended time horizons using its internal memory states. A more sophisticated implementation would have employed a rolling 30-day prediction window, allowing the model to learn the temporal evolution of volatility patterns and generate genuinely predictive insights about future market conditions.

6.4. Failure Analysis

The most catastrophic failure in our LSTM model emerged during the March 2020 COVID-19 market collapse, when the VIX exceeded 80 and implied volatility reached unprecedented levels. On March 16, 2020, our model predicted an implied volatility of 30.46% for a 30-day SPY put option with a 0.95 strike ratio, while the actual market-observed implied volatility reached 587.10% – representing a staggering 556% underestimation.

Our SHAP analysis revealed the underlying mechanism driving this systematic failure. The model assigned disproportionate importance to the 60-90 day historical window (0.34 importance score) while severely under-weighting the immediate 1-5 day volatility spike information (0.12 importance score). This temporal misallocation demonstrates that the LSTM had learned to expect mean reversion based on historical patterns where extreme volatility episodes typically resolved within days or weeks. However, during genuine regime breaks like the March 2020 crisis, this learned behavior became counterproductive, causing the model to essentially "fight the last war" rather than adapting to the unprecedented market conditions unfolding in real-time.

The root cause analysis identified a fundamental architectural limitation: our LSTM's 90-day lookback window contained insufficient representation of true black swan events, leading to systematic mean-reversion bias. This failure highlights the critical importance of incorporating extreme event simulation and regime-aware architectures in financial machine learning applications, particularly for deployment in live trading environments where such failures can result in substantial financial losses.

7. Discussion

7.1 Unsupervised Learning Insights

K-Means with Wasserstein dispersion emerges as the primary model, offering scalable, geometry-aware regime discovery with strong cohesion and stability across the full dataset.
Ward linkage serves as a complementary model, providing hierarchical interpretability and tighter intra-cluster compactness, however, it is computationally expensive and thus feasible only on sampled subsets. Consequently, Ward Linkage can be positioned as a secondary, confirmatory model for validating and contextualizing the volatility regimes identified by K-Means.
Together, these approaches provide both flat and hierarchical views of latent volatility structure, validated through bootstrapped ARI, ablation testing, and learning-curve saturation.

One key insight we gained was that unsupervised learning successfully uncovered latent volatility regimes that aligned with temporal market structure and lagged dynamics. The clusters differentiated periods of low-volatility

carry, transitional phases, and high-volatility stress conditions, reflecting meaningful shifts in implied volatility surfaces and Greeks behavior.

A surprising result is that K-Means and Ward linkage clustering both identified broad latent volatility regimes, highlighting consistent structural patterns in the data. However, there are notable differences in granularity: Ward linkage produced a significantly larger number of clusters (~29) compared to K-Means (12-16), effectively revealing nested substructures within the broader K-Means partitions. This hierarchical detail enhances interpretability, showing how high-level regimes can be decomposed into finer sub-regimes.

A major challenge we encountered was the absence of explicit "ground truth" regime labels, which made validation difficult. We addressed this by cross-referencing cluster timelines with external market indicators such as the VIX, S&P 500 drawdowns, and macro volatility events, confirming that several clusters aligned with known volatility spikes and transitions. Additionally, Ward linkage proved computationally intensive and scaled poorly beyond 30,000 records, restricting its use to subsampled datasets and requiring careful selection to preserve regime diversity.

With more time and resources, we could extend this work by modeling regime transitions using sequence-based embeddings or Hidden Markov Models to capture persistence and mean-reversion. Additionally, integrating forward-looking features such as option-implied skew term structures, realized volatility forecasts, and macroeconomic indicators would enhance regime interpretability and predictive power.

7.2 Supervised Learning Insights

The most surprising discovery in our analysis was the overwhelming dominance of volatility-based features in determining future implied volatility outcomes, which fundamentally challenged our initial hypothesis – reflected by our desire to include as many price based features as possible. While price-based features maintained some relative importance, the volatility feature group consistently emerged as the primary driver across all model configurations, suggesting that volatility exhibits much stronger autocorrelation than previously thought. From a practical perspective, this finding suggests that future model iterations should prioritize the development of more sophisticated derived volatility features, including volatility-of-volatility metrics, regime-transition indicators, and cross-asset volatility spillover effects that could capture the complex interdependencies driving options markets.

The clustering integration, while a good idea on paper, failed to materialize any significant improvements to the model. This ablation analysis demonstrates that unsupervised clustering more or less just displaced the original features, without adding any distinct value. This makes sense once you consider that the clustering was just done on the features themselves. Future implementations should place more of an emphasis on features which exist outside the traditional options dataset.

The absence of macroeconomic factors in our feature set represents a significant limitation that likely explains some of our model's shortcomings during extreme market events. Fundamental economic drivers such as Federal Reserve policy indicators, employment data, inflation metrics, and geopolitical risk indices that often serve as leading indicators for volatility regime shifts. Future implementations should systematically integrate economic calendar events, central bank communications, and cross-asset correlation structures that could provide early warning signals for the kind of regime breaks that devastated our current model's performance.

Perhaps the most operationally relevant insight concerns the challenges of working with massive options datasets and the need for more intelligent data curation strategies. This is a constraint recognized by options models everywhere, as options datasets are exponentially larger than the underlying asset datasets. Rather than accepting this limitation, future research should focus on developing domain-aware sampling techniques that preserve the most informative observations on just the strikes and expirations that matter.

8. Ethical Considerations

Data Bias and Representation

Issue: Historical datasets may embed **structural biases**, such as overrepresentation of low-volatility periods or specific crisis regimes. These biases can lead to distorted clustering or underrepresentation of rare but impactful events.
Mitigation: We incorporated **multi-year and cross-regime data**, spanning both stable and turbulent periods, and validated model stability under stress scenarios. Future work should expand data sources to include emerging markets and synthetic stress testing for improved representativeness.

Overfitting and False Confidence

Concern: Regime clustering may inadvertently capture noise rather than true structure, leading to overconfident interpretations and potential misallocation of risk.
Mitigation: To minimize this, we employed **bootstrap resampling**, **Adjusted Rand Index (ARI)** stability testing, and **cross-cycle evaluation**. These checks reduce the risk of false discovery and encourage cautious interpretation of regime boundaries.

Automation and Human Oversight

Concern: Automated regime detection and risk models can obscure human judgment, especially during regime shifts where market structure changes abruptly.
Mitigation: The system is designed for **decision support**, not full automation. Human analysts remain responsible for interpreting signals and can **override model outputs** during ambiguous or high-impact market conditions.

Unsupervised Model Bias and Fairness

Issue: Unsupervised methods may reflect **latent biases**, for instance, detecting regimes specific to developed markets or misinterpreting emerging market dynamics as anomalies. Mislabeling these structures could unfairly influence exposure or risk decisions.
Mitigation: Ongoing validation and **post-clustering interpretability analysis** are necessary to ensure regimes are economically meaningful and not artifacts of biased data or features. Future extensions, if they were applied to international markets, could include **fairness auditing** and **sensitivity tests** across geographies and instruments to promote equitable model behavior.

9. Statement of Work

Our project was a collaborative effort, with each member contributing to research, implementation, and reporting:

- **Chris Zhang**: Conceptualized the project and leveraged financial market expertise. Handled data preprocessing and feature engineering, ensuring data quality. Implemented supervised learning models and authored related sections, including ablation tests and failure mode analysis.

- **George Smith**: Led the unsupervised learning pipeline, implementing K-Means clustering and optimizing models. Built the interactive visualization dashboard, evaluated cluster stability (ARI), and contributed extensively to unsupervised learning reports, visualizations, and ablation tests.

- **Ayansola Akanmu**: Identified WRDS data sources and implemented Ward Linkage clustering. Ran evaluations for unsupervised learning on Great Lakes. Developed the initial supervised learning proof of concept and authored report sections, including Introduction, Feature Engineering, Unsupervised learning reports and visualizations, etc.

10. References

- Horvath, B., Issa, Z., & Muguruza, A. (2021). *Clustering market regimes using the Wasserstein distance*. https://arxiv.org/pdf/2110.11848
- McGreevy, J., Zhang, Y., & Kumar, R. (2024). *Detecting multivariate market regimes via clustering algorithms*. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4758243
- Zhang, W., Li, L., & Zhang, G. (2021). *A two-step framework for arbitrage-free prediction of the implied volatility surface*. https://arxiv.org/pdf/2106.07177

11. Appendices

Appendix A: Complete feature definitions and transformations

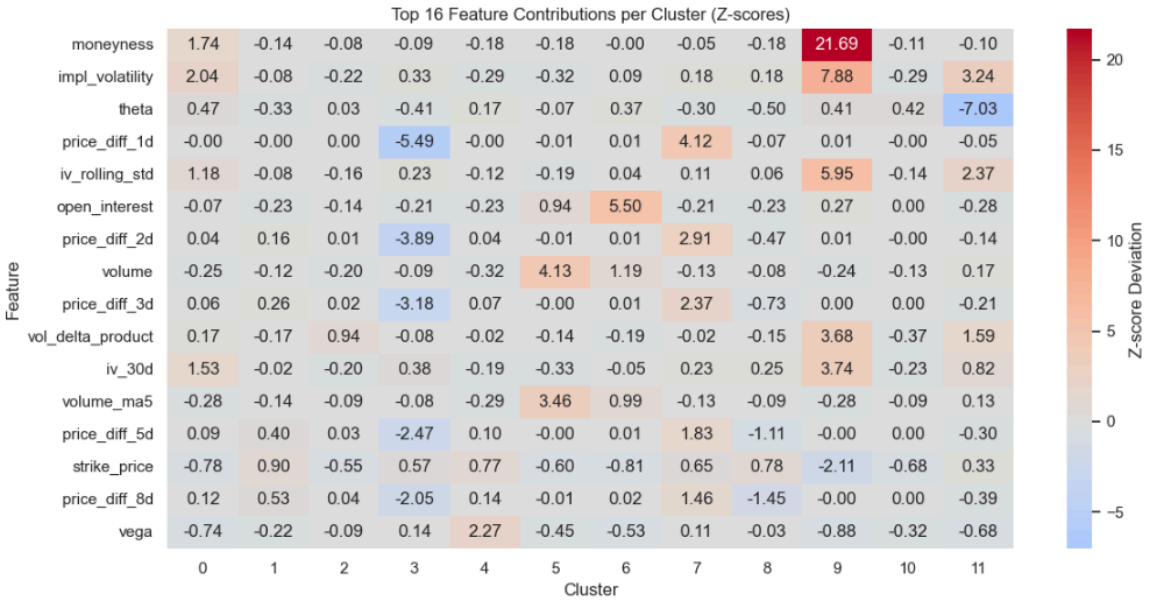
Feature Name	Description	Transformation	Source
date	Trade date	Parsed from raw; used for lag features	WRDS
secid	Security ID		WRDS
symbol	Option symbol	Cleaned and standardized	WRDS
cp_flag	Call/Put flag	One-hot encoded (for supervised learning)	WRDS
exdate	Expiration date	Parsed; used to derive expiry_indicator	WRDS
strike_price	Strike price	Scaled (if included in clustering)	WRDS
best_bid	Best bid price	Scaled	WRDS
best_offer	Best offer price	Scaled	WRDS
volume	Option volume	Scaled	WRDS
open_interest	Open interest	Scaled	WRDS
impl_volatility	Implied volatility	Scaled	WRDS
delta	Option Greek: delta	Scaled	WRDS
vega	Option Greek: vega	Scaled	WRDS
theta	Option Greek: theta	Scaled	WRDS
forward_price	Forward price estimate	Derived from underlying and Greeks	WRDS
expiry_indicator	Expiration flag	Binary flag (e.g. 1 if expires within 30 days)	WRDS

Feature Name	Description	Transformation	Source
prc	Underlying stock price	Scaled	WRDS
vol	Underlying stock volume	Scaled	WRDS
iv_30d	30-day forward implied volatility	Derived from volatility surface	Derived
price_diff_{d}d	Price difference over Fibonacci day lag	Multiple features: lagged returns over d ∈ {2,3,5,8,13,…233}	Derived
cluster	Cluster category from unsupervised learning	One-hot encoded (for supervised learning)	Derived
atm_iv	Implied volatility at ATM (at the money) strike	Extracted from nearest-to-ATM option	Derived
curvature	Convexity of IV curve at ATM	Second derivative of quadratic fit to IV vs. moneyness	Derived
skew	Slope of IV curve at ATM	First derivative of quadratic fit to IV vs. moneyness	Derived

- Appendix B: https://siads-696-ii.onrender.com

- Appendix C

Figure 5.2.3: Z-score deviations of the top 16 most discriminative features across 12 clusters, highlighting which features define each volatility regime - higher Z-scores (red) indicate that a feature is significantly elevated in that cluster.



- ***Appendix D: LSTM Model Results, clustered dataset vs unclustered dataset***

Model	**Train RMSE**	**Train R^2**	**Train MAE**	**Test RMSE**	**Test R^2**	**Test MAE**
LSTM (excluding clustering)	0.1143	0.4090	0.0489	0.1143	0.4090	0.0489
LSTM (including clustering)	0.1149	0.4028	0.0450	0.1156	0.3955	0.0452

Appendix E: Ablation Groups

'**volatility_features**': ['impl_volatility', 'iv_rolling_std', 'atm_iv', 'skew', 'curvature']

'**price_features**': ['prc', 'moneyness', 'price_diff_1d', 'price_diff_2d', 'price_diff_3d', 'price_diff_5d', 'price_diff_8d', 'price_diff_13d', 'price_diff_21d', 'price_diff_34d', 'price_diff_55d', 'price_diff_89d', 'price_diff_144d', 'price_diff_233d'],
'volume_features': ['volume', 'open_interest', 'vol', 'volume_ma5']

'**greeks**': ['delta', 'vega', 'theta', 'vol_delta_product'],

'**other**': ['cluster_0', 'cluster_1', 'cluster_2', 'cluster_3', 'cluster_4', 'cluster_5', 'cluster_6', 'cluster_7', 'cluster_8', 'cluster_9', 'cluster_10', 'cluster_11', 'cluster_12', 'cluster_13', 'cluster_14']

Appendix F: Evaluation Metrics for Clustering Methods Under Different Configurations

Ticker	Method	Best Parameter	Number of Clusters	Silhouette Score	Calinski-Harabasz Score	Davies-Bouldin Score	Wasserstein distance	Stability (ARI)	pca0.95	bootstrap	sample_fraction	approx_training_size	num_features
SPY	KMeans	k = 17	17	0.100384	5335756	2.475983	0.658625	0.499761	TRUE	6	0.8	50m	16
SPYQQQ	KMeans	k = 15	15	0.114559	7565080	2.538623	0.664344	0.544838	TRUE	5	0.6	70m	16
SPY	KMeans	k = 7	7	0.178157	2483801	2.510417	0.650066	0.516003	TRUE	5	0.7	20m	11
SPY	KMeans	k = 9	9	0.128522	2211.172	2.927238	0.68436	0.330186	TRUE	4	0.5	30k	16
SPY	Ward Linkage	threshold = 70.00	29	0.099973	301.7803	1.506891	0.707115	0.403244	TRUE	4	0.5	30k	16
SPY	KMeans	k = 17	17	0.100384	5335756	2.475983	0.658625	0.56923	TRUE	4	0.5	50m	16
SPY	KMeans	k = 16	16	0.123203	2574.53	2.758641	0.51288	0.501076	FALSE	4	0.5	30k	16
SPY	Ward Linkage	threshold = 100.00	18	0.093575	291.9187	1.518012	0.56016	0.384025	FALSE	4	0.5	30k	16
SPY	KMeans	k = 11	11	0.129328	1187073	2.866254	0.678114	0.435919	TRUE	5	0.7	13m	11
SPY	KMeans	k = 2	2	0.204634	1335224	2.546271	0.685119	0.511013	TRUE	5	0.7	13m	16
SPYQQQ	KMeans	k = 14	14	0.336889	11580.61	1.805009	0.665	0.401847	TRUE	5	0.6	100k	16
SPYQQQ	Ward Linkage	threshold = 100.00	39	0.074198	290.0513	1.647427	0.646332	0.304484	TRUE	5	0.6	100k	16

QQQ	KMeans	k = 9	9	0.30743	2515.819	2.306894	0.681327	0.657215	TRUE	4	0.5	25k	16
QQQ	Ward Linkage	threshold = 80.00	20	0.102789	338.6334	1.536217	0.703535	0.480575	TRUE	4	0.5	25k	16
QQQ	KMeans	k = 13	13	0.277456	2761.05	2.569935	0.553543	0.564846	FALSE	4	0.5	30k	16
QQQ	Ward Linkage	threshold = 80.00	21	0.097346	314.5111	1.615714	0.538292	0.437911	FALSE	4	0.5	30k	16
SPY	KMeans	k = 12	12	0.24788	3556856	2.089139	0.658099	0.607525	TRUE	5	0.7	50m	11
SPY	KMeans	k = 15	15	0.175627	1319162	2.543943	0.602445	0.516046	FALSE	5	0.7	13m	16
QQQ	KMeans	k = 9	9	0.288864	3336.267	2.361386	0.662897	0.530787	TRUE	4	0.5	30k	16
QQQ	Ward Linkage	threshold = 70.00	27	0.09872	349.3557	1.63765	0.680207	0.513027	TRUE	4	0.5	30k	16

Appendix G:

GitHub Repository: https://github.com/chriszhang08/siads-696-milestone-ii-final/tree/master