Stock Return Volatility Forecasting with Machine Learning Models

Qunkun Ma, Xin Pan, Rohit Baddam

Introduction

The financial market is a highly volatile and complex environment. Stock return volatility is a fundamental measure of market risk and a critical component in option pricing. Investors are interested in the factors that would affect volatility in the drastic market. Forecasting stock volatility is essential for risk management and portfolio construction for investors, portfolio managers, and options traders. Previous studies have found that machine learning models can better investigate the complex relationship of the stock market through multi-dimensional factors. Our project intends to forecast the 5-day rolling volatility of the SPY with a series of macroeconomic variables, including interest rates, consumer price index (CPI), producer price variables (PPI), unemployment rate, and gross domestic product (GDP).

We have applied both supervised and unsupervised methods for the project. For the supervised learning approaches, we used GARCH/GARCHX, ARIMA/ARIMAX, and XGBoost. As for the unsupervised learning, we have firstly applied PCA as a dimension reduction method and then we have clustered the groups with k-mean clustering and DBSCAN. The results showed that XGBoost has outperformed both GARCHX and ARIMAX in terms of forecasting volatility. Through unsupervised learning, we reduced the data dimension and figured out the features of different market regimes after the K-mean and DBSCAN.

Related Work

One previous study was done by Song et al. (2022) and they investigated stock market volatility with low-frequency economic variables based on a model that integrates deep learning methods with generalized autoregressive conditional heteroskedasticity (GARCH). They found that adding macroeconomic variables can significantly improve the accuracy of prediction after comparing different models. Similarly, Christensen et al. (2023) have compared several ML algorithms for volatility forecasting and found ML models are better at handling complex structure of the financial markets and extracting important information from additional variables. Another interesting study by Kotsompolis et al. (2025) has applied the ARIMA model as the baseline with the XGBoost model to forecast the carbon price movement. They found that XGBoost outperforms the baseline model and suggested that ML-based approaches can be an efficient alternative to traditional statistical and econometric models.

However, there are some differences from our project when compared with the related studies. Our dataset is focused on the SPY index with various indicators such as the interest rate, CPI, PPI, and unemployment rate, which enable us to capture market expectations about inflation and economic growth. We have employed models with different mechanisms, such as linear time series, decision trees, etc. Moreover, we have added unsupervised techniques, including k-mean clustering and DBSCAN, to explore any patterns like market regimes in the financial market.

Data Sources

The project has retrieved data primarily from two resources, i.e. Yahoo Finance and FRED, to construct the time-series dataset for modeling, which covers a period from January 2021 to September 2025. In total, this amounted to approximately 1,200 daily records.

First, daily SPY and VIX data was accessed via the yfinance Python library, which acts as an API to the Yahoo Finance service. Second, macroeconomic indicators were sourced from the FRED database. Using the pandas-datareader library, seven key time-series variables were retrieved, including various interest rates, inflation indicators, i.e. CPI, PPI, the unemployment rate, and GDP. These series were also returned as pandas DataFrames with their reporting frequencies of daily, monthly, or quarterly.

Feature Engineering

We have performed several steps to transform the raw data into our final dataset. First we merged daily SPY data from Yahoo Finance, daily VIX data from CBOE, and low-frequency macroeconomic data from FRED into a dataframe indexed by data. For our most important target variable, we calculated the log returns for the SPY index and the rolling realized volatility (RV) within a 5-day rolling window. In this step, we also calculated an annualized RV which is more comparable for the unsupervised learning section. Instead of using the absolute value of macroeconomic variables such as CPI, PPI, and GDP, we calculated the percentage change (%) of each variable. Since those data are monthly or quarterly-based, then we performed a forward-fill (*ffill*) to populate these variables with daily information. We also calculate the log return of VIX to capture its daily fluctuation in the market. As there are some missing values after the percentage change or rolling RV calculation, we dropped those rows containing null values and used the final dataframe as the version for machine learning models.

Feature name	Description	Type	Source
SPY_log_return	Daily log return of the SPY index	Numerical	Engineered
SPY_vol_5d	5-day Rolling Realized Volatility	Numerical	Engineered
Overnight Rate	Borrowing rate between financial institutions	Numerical	FRED
1-Month Rate	Interest rate for short term loans and securities with one-month maturity	Numerical	FRED
3-Month Rate	Interest rate for short term loans and securities with three-month maturity	Numerical	FRED
CPI_pct_change	Percentage change in CPI, the rate of inflation by consumers for a basket of goods and services	Numerical	Engineered

PPI_pct_change	Percentage change in PPI, the rate of price change by domestic producers	Numerical	Engineered
Unemployment Rate	Percentage of labor force that is jobless and actively seeking employment	Numerical	FRED
GDP_pct_change	Percentage change in GDP, measuring the economic growth rate	Numerical	Engineered
VIX	A real-time market index representing the expectations for volatility	Numerical	CBOE
VIX_log	The logarithm of the vix value to make the data normalized	Numerical	Engineered

PART A Supervised learning

Our supervised learning workflow has included the linear regression as a base line model, GARCH/GARCHX as the key econometric model, ARIMA/ARIMAX as a time series model, and XGBoost as the regression model for volatility forecasting. For all models, the data was split chronologically into an 80% training set and a 20% testing set. We have also conducted cross-validation on each model to calculate the mean of performance metrics.

Linear Regression

A linear regression model was implemented to serve as a baseline to contrast performance with other models. The linear regression model takes external factors in the data frame as feature variables and *SPY_vol_5d* as the target variable.

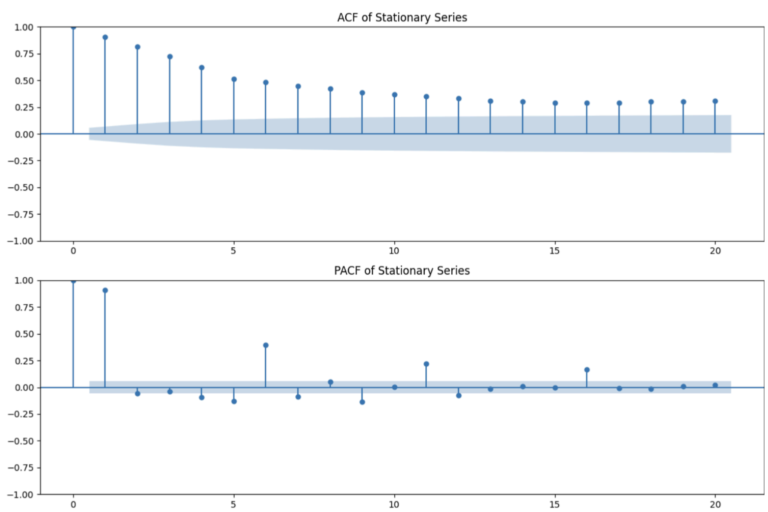
GARCH and GARCHX

The GARCH (1,1) model is chosen as it is designed to model the dynamics of market volatility. Unlike complex models which require massive computations, it is able to capture the key market behaviors with three parameters: volatility clustering, mean reversion, and realistic shock persistence (NYU Stern School of Business., n.d.). It provides an effective framework for volatility forecasting for us to interpret. In addition to the GARCH(1,1) model, we experimented with a GARCHX(1,1) model, with the X referring to the inclusion of external factors in the dataset. This model was selected as it fits in well with the objective of our project to measure effects of external variables on stock volatility.

As the GARCHX model requires exogenous variables to be forecasted, we selected the top three correlated variables (*VIX, CPI_pct_change,* and *GDP_pct_change*) within the dataset. Additionally, we plan to perform PCA for dimensionality reduction and see whether it would improve model performance. When forecasting results, we selected a horizon of 1 day using yesterday's data with the exogenous variables to predict SPY 5 day volatility as the target variable.

ARIMA and ARIMAX

Both the ARIMA and ARIMAX work under the assumption that the future values of a series are primarily dictated by its past values. ARIMA is used for modeling the level of the time series itself and measures the relationship between a time series and a lagged version of itself over the time interval. This is fundamentally utilizing historical data to forecast future values with models such as AutoRegressive (AR), Moving Average (MA), and their combination (ARMA) (Majka, 2024). While ARIMA looks at the univariate component of the past values of the time series, ARIMAX makes it possible to track shocks with independent predictors by introducing exogenous variables. The ARIMAX model is considered to be a form of multivariate regression model which extends the advantage of autocorrelation that may be present in residuals of the regression (Tamuke et al, 2018). In our case, we will use the macroeconomic variables as exogenous predictors with the ARIMAX mode to see if it can improve the accuracy of a forecast.



The major parameters that are tuned are the order (p, d, q). We used the Augmented Dickey-Fuller (ADF) test and got an ADF p-value of 0.0048 which rejects the null hypothesis and indicates that the series is non-stationary; therefore we used a differencing of 0 (d=0). Once we have a stationary series, we plotted the Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) to capture the significant autocorrelation at the lag. The gradual decline in the ACF shows a long-term dependency with slow decay in the data, but it does not show periodic peaks at regular intervals. From the PACF plot, we found significant spikes at lag 1, lag 6, and lag 11, which occurs as the time series, i.e. rolling volatility, is using a moving average of 5-day.

XGBoost

XGBoost is a tree based ensemble machine learning algorithm that is widely applied in complex forecasting schemes (Kotsompolis et al., 2025). Since the financial dataset can contain a complex structure, XGBoost would be a top-tier competitor in volatility forecasting with high accuracy and robustness for our model comparison.

For parameter selection for the XGBoost model, we utilized GridSearchCV from the scikit-learn library. After specifying the range of *n_estimators* of [100, 200, 300], *max_depth* of [3, 4, 5], *learning_rate* of [0.01, 0.05, 0.1], *subsample* of [0.8, 1], *colsample_bytree* of [0.8, 1], GridSearchCV assisted testing out the model parameters and outputted the best parameters. It turns out the best parameters are *n_estimators*: 100, *max_depth*: 3, *learning_rate*: 0.05, *subsample*: 0.8, *colsample_bytree*: 0.8. The estimated RMSE is around 0.01068. Hence, we used the parameters for our XGBoost Model.

These parameters performed well in later testing. The fewer number of trees with *max_depth* of 3 could avoid the model being overly complex. A slower learning rate and a subsample set to 0.8 could help prevent the model from overfitting. Thus, as the result indicated, the XGBoost model resulted in a lower R-Squared score but is more robust and can make better generalized predictions in comparison to other models.

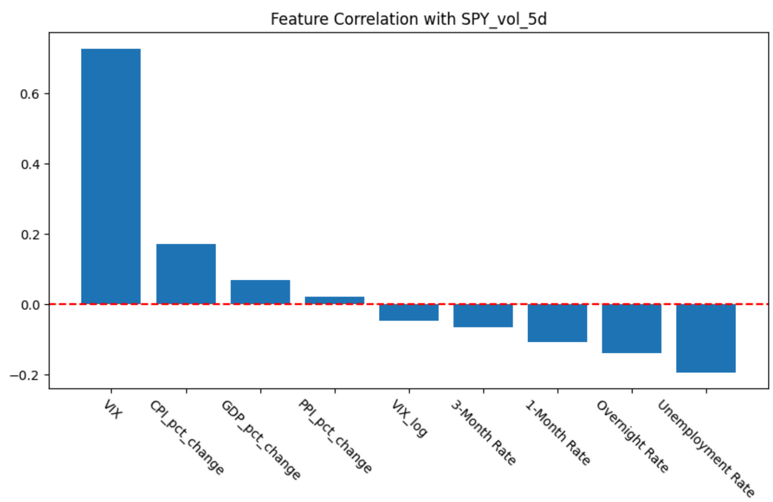
With all the models we applied above, we have evaluated them with commonly used loss functions including the Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) (Vee et al. 2011). The mean metrics are calculated across 5-fold cross validation, and below is a summary of the metrics.

Model	RMSE	MAE	R-square	Key Findings
Linear Regression	0.015	0.010	0.623	Baseline model
GARCH	0.020	0.0166	0.732	The model performed better than a simple linear regression model, using such model as baseline in comparison to the GARCHX model
GARCHX	0.019	0.0161	0.839	Improved performance when including exogenous variables; a higher R-Squared value covering a wider range of variances in the target variable
ARIMA	0.012	0.0084	0.816	No significant autocorrelation in residuals (Ljung-Box Test p>0.05); however, residuals are NOT normally distributed (Jarque-bera 3199, Heteroskedasticity p <0.05); outliers exist in the dataset
ARIMAX	0.013	0.0087	0.816	No significant autocorrelation in residuals (Ljung-Box Test p>0.05); however, residuals are NOT normally distributed (Jarque-bera 2150, Heteroskedasticity p <0.05); outliers exist in the dataset
XGBoost	0.0101	0.0071	0.811	Lower R-Squared value than the GARCHX model, but lower score on MAE and RMSE on its volatility predictions, outperforming all supervised learning methods attempted

Feature Importance Analysis

To measure feature importance, the correlation between the external variables and the target variable *SPY_vol_5d* has been calculated. As the result of the table below, VIX has the highest correlation with the target variable, followed by *CPI_pct_change, GDP_pct_change,* and *PPI_pct_change*. Then there are other variables that have negative correlations, but their values are not as significant. For our supervised learning models, we will try to include all exogenous variables as they have underlying relations with each other. However, for GARCHX, which requires forecasting exogenous variables, only the top 3 most important ones were included.

A GARCHX Model using the top 3 absolute values of correlation with the target variable has also been experimented with. However, no significant improvement was observed; the mean R-Squared value of the model decreased, while RMSE is roughly the same as the GARCHX Model using the top 3 most correlated features, ranked from high to low (*VIX, CPI_pct_change, GDP_pct_change*)



Feature	Correlation
VIX	0.728244
CPI_pct_change	0.173100
GDP_pct_change	0.069981
PPI_pct_change	0.022299
VIX_log	-0.046392
3-Month Rate	-0.065291
1-Month Rate	-0.105819
Overnight Rate	-0.139574
Unemployment Rate	-0.193069

Sensitivity Analysis

While optimizing model performance with GARCH and GARCH-X, we experimented with different tuning of different parameters. First, we attempted to increase p and q to 5, but this did not result in significant improvements to model performance. However, adding a specified parameter "lags" seems to improve model performance significantly. Using our GARCH model as an example, with the same model parameters of p=1 and q=1, with "lags=0" as default, the model produced an R-squared value of 0.0, with AIC and BIC being 2822.15 and 2842.59, respectively.

Setting the lags of 1 while keeping all other parameters constant helped improve R-Squared to 0.823. This shows a large improvement in the coverage of variances on the target variable, as longer lags could benefit financial data modeling with delayed effects, The decreased AIC and BIC to 1573.29 and 1598 indicates a better model fit. This also seems to corroborate our findings from the ACF graph. Additionally, we also tried different values for lag, but the effect isn't as drastic as switching from 0 to 1.

Identify Tradeoffs

We have observed a tradeoff between model accuracy with training data size when we were conducting the Time Series 5-fold CV. With this method, the training set would be much smaller for earlier folds and larger for the most recent folds. Though training the model on the larger dataset would improve training performance for complex patterns, it comes with the cost of overfitting the model and getting less accurate forecasts. In our case, the model trained on the CV5 split is based on the largest and most recent data, therefore it gets a higher R-squared on its training periods, indicating that it explains most of the variance in the data. However its performance on unseen test data was worse, with higher RMSE (0.0222) and MAE(0.0149). The model trained on the CV4, on the other hand, obtains lower prediction error with RMSE (0.013) and MAE (0.0121) on the test set.

Failure analysis

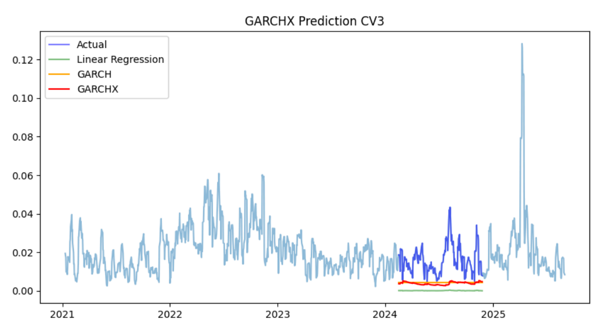
During our supervised learning modeling, we have identified a few issues with regards to feature relevance, data quality, and model limitations. Some potential methods for improvements may include building a more dynamic framework, i.e. an ensemble as a meta-model with multiple algorithms which is more robust in feature selections.

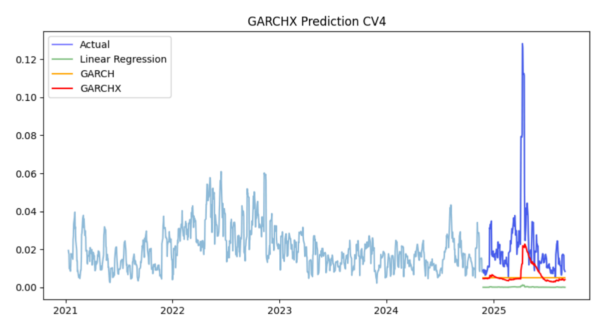
1) Selection of external variables (Feature Relevance)

As mentioned, the external economic factors we selected in the dataset should be correlated with stock data (i.e. CPI and GDP). However, some features may have stronger correlation with the stock price rather than volatility. This resulted in external variables in the dataset not having as big an impact as initially expected on predicting stock volatility.

2) Market Uncertainties (Data Quality/External Factors)

Despite our best efforts to limit the scope of the project to relevant data, the stock market is not easily predictable. As shown below, in the selected timeframe, the stock market experienced unusual volatility around April 2025 due to the announced government strategies on import tariffs. As a result, the market experienced unexpected volatility in a short period of time. Since the model is trained with previous data, it fails to capture such sudden events well. As demonstrated in the graph below, the GARCHX performs better with CV3 training data in comparison to CV4, resulting in a similar R-Squared to the model trained with CV4 data but lower MAE and RMSE, as the volatility is relatively normal compared with data after April, 2025. The second graph shows that GARCHX reacted better with the sudden increase of stock volatility compared to other models; however, as the previous training data did not contain such extreme values, the model's prediction suffers as a result when compared to its predictions trained with CV3 data.





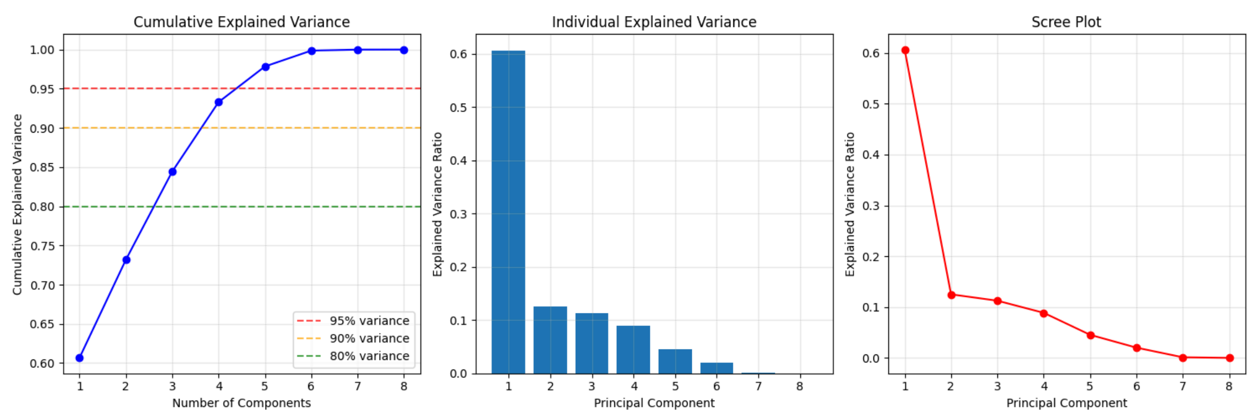
3) ARIMA Model (Model Limitation)

One critical assumption of the ARIMA and ARIMAX model is the assumption of constant variance. Though we have narrowed down the timeframe to less outliers in history, market volatility may have a sudden outbreak at irregular time intervals with extreme high and low values. We observed heterogeneity in our model and the normal distribution assumption of residuals is not met. Therefore, there exist limitations of the model itself in the financial market for the analyzed time series (Petrica & Tindeche, 2016). That's one of the factors that both of the models are not performing well on volatility forecasting for our dataset.

PART B Unsupervised Learning

The primary goal of the unsupervised learning techniques is to uncover hidden structure that underlies the volatility and macro-economic data, such as market regimes over time. We used Principal Component Analysis (PCA) for dimensionality reduction, and then applied the k-means algorithm to segment data points into groups based on similarity. PCA was chosen since it can enhance the efficiency and interpretability of k-mean clustering by reducing the impact of irrelevant variables and navigating to the most dominant data patterns. K-means was applied for its capability to provide centroids which enable us to interpret the market regimes. On the other hand, considering there are some potential outliers over time, i.e. extreme volatility, within our dataset, which may greatly change the position of centroids of each group, we also applied the DBSCAN as a way to identify noise and analyze the potential groups.

Principle Component Analysis
Considering the amount of variables we have and potential multicollinearity among some features, we started our exploration with PCA to reduce the dimension of our dataset while aiming to capture the most informative variance. In this step, the main hyperparameter we tuned is the number of components (*n_components*). Based on the plot for cumulative explained variance and the scree plot, we can see that 5 components can preserve more than 95% of the total variance, while the "elbow" shows the plot turn and begins to flatten out at 5. With both plots considered, we use a *n_components* of five in our dataset, which preserves around 95% variance of the original dataset and allows for visualization of the processed dataset.

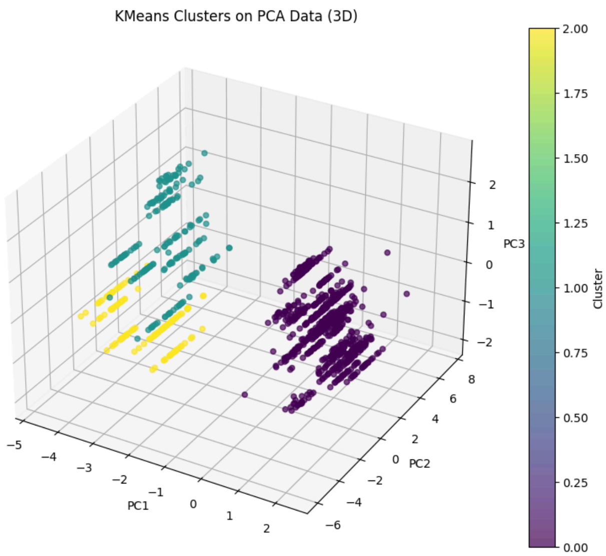


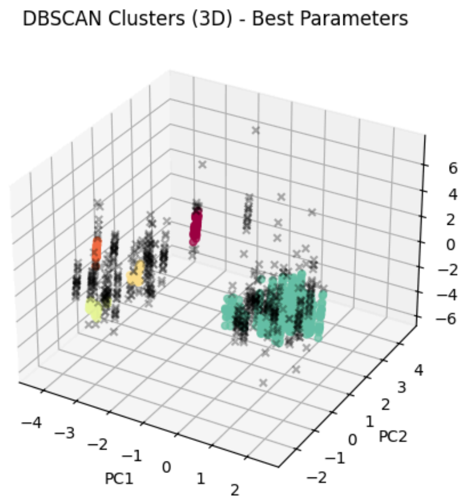
K-mean Clustering

After PCA reduces the number of dimensions while preserving the most important variance, we were able to project our data into a lower dimensional space. In order to decide on the number of clusters (*n_clusters*), we used the elbow method and Silhouette Score to measure how similar the observation is to its own cluster compared to other clusters. We found a cluster of 3 getting the best silhouette score of over 0.5. In order to get a more detailed and meaningful perspective, we chose to conduct a k-mean clustering with 3 clusters and its Silhouette score reaches 0.5, which shows a relatively strong structure. Based on this clustering output, we can interpret the three distinct market regimes and economic environments.

DBSCAN

To compare the result of k-means clustering, we also used DBSCAN to explore groups based on data density. For DBSCAN, we don’t need to specify the number forehead and it can help us get rid of outliers, which k-means cannot perform. In this step, we tuned the epsilon (*eps*) to control the radius of points to be grouped as neighbors of each other and *min_samples* parameters to define the minimum number of points to form a dense region. The result finds the best model being also a cluster of 5 with an *eps* of 0.7 and *min_samples* of 20, with a silhouette score of 0.671.





Evaluation

To evaluate the PCA reduced dataset, we used the Random Forest to calculate the R-squared of its prediction on our target variable, rolling return volatility. While PCA is able to preserve around 80% of the overall variance, it has dropped some information for predicting our rolling volatility. The underlying relationship between our macroeconomic variables and volatility is quite complex, and the features that explain the most variance in feature variables may not necessarily correspond with the features with the most predictive power. Therefore, in our supervised learning approaches, we chose not to include the dataset with PCA.

In order to evaluate our clustering results, we used a series of metrics, including the silhouette score, Calinski-Harabasz Index, and Davies-Bouldin Index to measure the clustering performance. The result of the two best performing models are presented in the table below:

Model	Number of k	Silhouette	Calinski-Harabasz Index	Davies-Bouldin Index	Key Findings
K-means	3	0.552	1433	0.812	Some overlapping between cluster 1 and cluster 2 observed; but better on the Calinski-Harabasz Index

DBSCAN	5	0.598	687.5	0.675	Higher Silhouette Score and lower Davies-Bouldin index shows better performance, as 40% of the data treated as the noise, therefore contains less data records

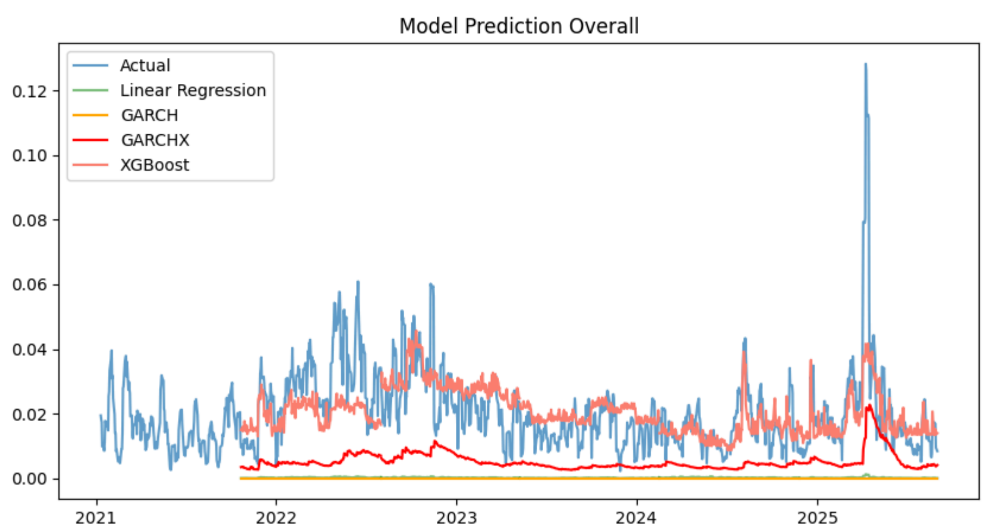
Sensitivity Analysis

In our DBSCAN model, we found the model clustering results depend on both the eps and min_samples. As we increase the size of epsilon, from 0.7 to 0.8, with the same number of min_sample, the model finds more clusters, but the silhouette score dropped dramatically to 0.27; when we decrease the min_samples from 20 to 15, its silhouette score dropped to 0.15. As the approach that DBSCAN defines clusters is based on absolute density and the change will be dramatic when it deals with noises, the clustering result is not quite stable. In the k-mean clustering, both a cluster of 2 and 3 get a high silhouette score of more than 0.5 but the score also decreases sharply if we define 4 clusters.

Discussion

Part A

In the Supervised Learning section, we applied a Linear Regression Model as a baseline, then a GARCH Model for volatility prediction, a GARCHX model with external variables, and similarly ARIMA and ARIMAX models, and lastly an XGBoost Model. Out of the four models, we found that GARCHX performed better than the GARCH model in both R-Squared, as well as MAE and RMSE metrics, while ARIMA performed slightly better than ARIMAX with both evaluation metrics. However, we are surprised to find that the XGBoost model was able to outperform both GARCHX and ARIMAX. This could be due to there being underlying, non-linear relations between our feature variables. As an ensemble model, XGBoost could better handle complex predictions in comparison to a GARCH and ARIMA model. Below is a visualization of model prediction, XGBoost managed to outperform GARCHX as it responds to fluctuations of volatility better than other models.

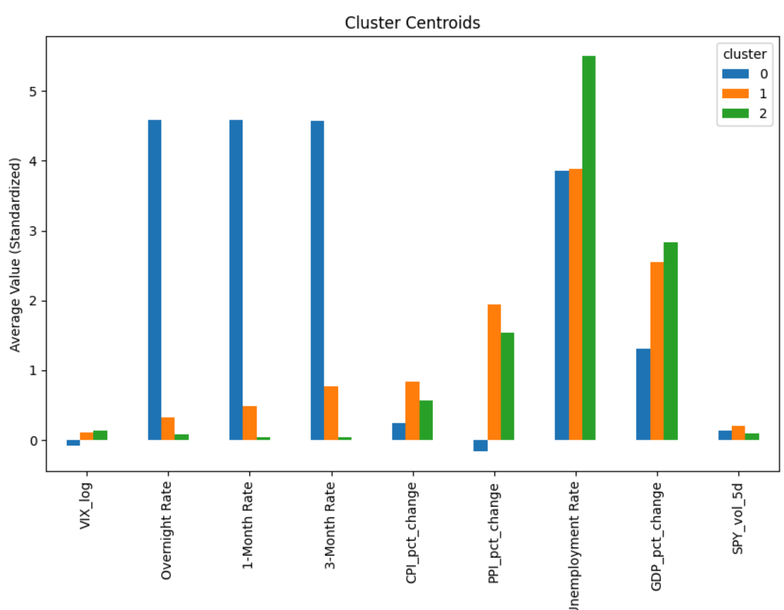


Some of the challenges we encountered were parameter tuning, as mentioned in the previous section; setting “lags=1” significantly improved model performance for GARCH and GARCHX. There were also difficulties in getting the GARCHX model running, as the forecasted exogenous variables need to be in a specific shape. We tackled this problem by breaking down the process into smaller steps and printing out the shape of the forecasted exogenous variables, and eventually got the model working. Having to forecast exogenous variables also limited the number of exogenous variables we could include in the model. Eventually, we decided on the top three, as correlations between the target and feature variables seem to drop off significantly after 3. This could, in turn, help GARCHX perform better than GARCH; however, it may also explain why XGBoost performed better due to having more feature variables and being able to discover relations among them in correlation to the target variable.

If we get more time and resources, we will conduct a hybrid model such as the ARMAX-GARCHX, ARIMA-GARCH model, as some researchers have found these hybrid models effective when forecasting exchange rate volatility and gold prices (Dinga, 2025). Another good approach would be building a more dynamic framework, i.e. an ensemble as a meta-model of multiple algorithms which is more robust in deciding on the methods and features.

Part B

With both k-mean clustering and DBSCAN, we found it interesting to capture the clusters that can be interpreted through the market regimes from 2021 to this year. The largest cluster of both k-mean and DBSCAN manifests periods with low volatility and high interest rate, and it was considered an environment where there was balanced growth and low inflation without being too hot or too cold. The second cluster is a classic post-crisis market environment where the central bank has pinned interest rates near zero and the economy is rebounding; the market is stable and confident because of the Fed’s policy. The third cluster features high inflation for both producers and consumers and the market is volatile as investors are concerned with the risk associated with economic slowdown.



Our initial exploration on unsupervised learning models contains a broad timeframe from 2000 to the present, which contains several significant market disruptions, such as the Dot-com Bubble, the Financial Crisis, and the COVID-19. We face challenges due to the large number of outliers and the clustering results were disrupted by these abnormal data points, which is reasonable but does not help much in identifying different market environments with more insightful interpretations. Therefore, we decide to narrow down the analysis timeframe with more consistent market behaviors so that we can distinguish between different market regimes. If we get more time and resources, we would like to investigate broader timeframes with a more dynamic framework like multi-stage clustering which enables us to capture both local and global relationships and gain a more solid understanding of different layers.

Ethical considerations

The adoption of ML models in finance raises ethical and legal concerns regarding transparency, fairness, and accountability (Gupta et al., 2025). Most of the ethical concerns relate deeply to the supervised learning model, while both could cause some issues with regard to market risk and market inequality.

For supervised learning models, as the model uses historical data to make forecasts, the lack of transparency of the machine learning model could give rise to excessive risk-taking behaviors in the market. Consequently, if different parties of interest apply a similar model, they might react simultaneously to the market with same trading behaviors, such as short selling, and this herding behavior can in turn cause extreme volatility and systemic risk. To deal with some of the concerns, we would advocate for a more Human-in-the-Loop (HITL) approach by incorporating human insights at critical points, especially when making high-stakes decisions (Joshi, 2025).

As for the unsupervised learning model, much as its risks are not directly concerned with forecasting, the model can also erode market trust through its depiction of the current market regime. When investors consider its results as a truth, they could blindly follow the model without making their own judgement. Moreover, as resources of advanced modeling are more accessible to institutional investors, this could create information asymmetry and widen the gap between institutional players and individual investors.

Statement of Work

Qunkun Ma	Xin Pan	Rohit Baddam
Data preprocessing, PCA/ICA, linear regression, GARCH/GARCHX model, XGBoost model, report writing for Part A and discussion section	Data preprocessing, K-mean clustering, DBSCAN, ARIMA/ARIMAX, report writing for lit review, feature engineering, Part A/B, discussion, and ethics	Exploratory analysis, XGBoost model, GARCH model, report writing data sources / feature engineering

References

Christensen, K., Siggaard, M., & Veliyev, B. (2023). A machine learning approach to volatility forecasting. *Journal of Financial Econometrics*, *21*(5), 1680-1727.

Dinga, B., Claver, J. H., Kum, C. K., & Che, S. F. (2025). Predicting Exchange Rates Volatility Using Hybrid ARIMA-GARCH Model: A Comparative Analysis. In M. Galety, J. Claver, A. Sriharsha, N. Vajjhala, & A. Natarajan (Eds.), *Data Analytics and AI for Quantitative Risk Assessment and Financial Computation* (pp. 107-130). IGI Global Scientific Publishing. https://doi-org.proxy.lib.umich.edu/10.4018/979-8-3693-6215-0.ch005

Gupta, S. K., Rosak-Szyrocka, J., Rena, R., Shieh, C. S., & Bayram, G. E. (2025). *The Impact of Artificial Intelligence on Finance: Transforming Financial Technologies.*

Joshi, R. (2025). Human-in-the-Loop AI in Financial Services: Data Engineering That Enables Judgment at Scale. *Journal of Computer Science and Technology Studies*, *7*(7), 228-236.

Kotsompolis, G., P. Cheilas, K. Konstantakis, E. Sfakianakis, S. Goutte, and P. Michaelides. 2025. " Smart Forecasting of Carbon Prices Using Machine Learning and Neural Networks: When ARIMA Meets XGBoost and LSTM." *Journal of Forecasting* 1–14. https://doi-org.proxy.lib.umich.edu/10.1002/for.70025.

Majka, M. ARIMAX Time Series Forecasting with External Variables. *ResearchGate (Sept. 2024).* Retrieved October 21, 2025 from: https://www. researchgate. net/publication/384196976_ARIMAX_Time_Series_Forecasting_with_External _ Variables.

New York University Stern School of Business. (n.d.). GARCH. NYU Stern Volatility Lab. Retrieved October 18, 2025, from https://vlab.stern.nyu.edu/docs/volatility/GARCH

Petrică, A. C., Stancu, S., & Tindeche, A. (2016). Limitation of ARIMA models in financial and monetary economics. *Theoretical & Applied Economics*, *23*(4).

Song, Y., Tang, X., Wang, H., & Ma, Z. (2023). Volatility forecasting for stock market incorporating macroeconomic variables based on GARCH‐MIDAS and deep learning models. *Journal of Forecasting*, 42(1), 51-59.ik

Tamuke, E., Jackson, E. A., & Sillah, A. (2018). FORECASTING INFLATION IN SIERRA LEONE USING ARIMA AND ARIMAX: A COMPARATIVE EVALUATION. MODEL BUILDING AND ANALYSIS TEAM 4. *Theoretical and Practical Research in Economic Fields*, *9*(1 (17)), 63-74.

Vee D. N. C. Gonpot P. N. Sookia N. (2011). Forecasting Volatility of USD/MUR Exchange Rate using a GARCH(1,1) model with GED and student's-t errors. *University of Mauritius Research Journal,* 17, 1–14.