# Equity Factor Rotation Strategy: A Two-Stage Machine Learning Approach


## Abstract

In this paper, unlike most studies that focus on the long-term returns of individual factors, we focus on the relative performance among factors. Specifically, we use a two-stage model to predict how factors perform relative to each other under different market conditions. In the first stage, we apply a clustering model to identify the number of market regimes, and then use a stacking model to predict the regime for the next period. In the second stage, we employ an ensemble model to predict the winning factor in each regime, and determine the portfolio weights based on the predicted probabilities. Through careful feature processing and label merging, our classification model achieves higher accuracy in predicting market regimes compared with the base paper. The backtesting results also show that our strategy performs significantly better than the benchmark.


## Background and Introduction

In recent decades, factor investing has received sustained and widespread attention in the field of quantitative investment. Building on Sharpe's Capital Asset Pricing Model (CAPM) (Sharpe, 1964), the concept of factor-based investing was first proposed by Fama and French (1993) and has since evolved in many directions. On one hand, there is a rich body of research focusing on the long-term return differences among factors, such as traditional multi-factor models. On the other hand, relatively few studies have examined the relative return differentials between factors themselves.

The paper we aim to replicate "Equity Factor Timing: A Two-Stage Machine Learning Approach" (DiCiurcio et al., 2024) proposes a strategy based on factor-style rotation across different market conditions.

In addition to studying relative returns among factors, another key contribution of this paper is its data-driven machine learning framework for identifying market regimes and examining factor performance within each regime. We believe that both the early detection of market states and the analysis of factor return differentials conditional on these regimes have significant implications for risk management.

For example, if we can identify in advance that the upcoming period is likely to be a bull market, we may increase leverage and allocate additional capital. Conversely, if the model predicts a bear market, we can prepare appropriate hedging strategies ahead of time. Moreover, identifying statistically significant winning factors under different regimes allows us to allocate assets from a factor-based perspective, thereby improving the portfolio's Sharpe ratio.

While these ideas are highly inspiring, this study primarily focuses on using machine learning to estimate the conditional probability of next-period factor outperformance under various market regimes.

Therefore, our working hypothesis is that using K-means to identify the number of existing market states, and then applying classification models to predict the current state of the market based on financial turbulence and macroeconomic indicators, provides a reliable foundation for machine-learning-based factor rotation investing. In other words, conditional on these predicted regimes, supervised ensemble learning models can better identify next-period factor outperformers than naive benchmarks, ultimately leading to higher risk-adjusted portfolio returns.

# Data & Feature Engineering

## Data Sources

### Macroeconomic & Financial indicators

Macroeconomic data have always served as an important reference for assessing financial market conditions. In the base paper, a total of five broad categories of macroeconomic factors are discussed comprising 25 macro drivers in total. However, only ten macro features are explicitly listed in the paper. After excluding the Conference Board Business Cycle Labels, which are proprietary and not freely available, we collected nine features in total, as shown in the table below.

| **Macro Categories** | **Variables Considered for Machine Learning Testing** |
|---|---|
| Inflation | Trailing ten-year annualized changes in core CPI<br>Trailing one-year annualized changes in core CPI<br><br>Sources: FRED, Federal Reserve Economic Data |
| GDP Growth / Business Cycle | US Real GDP Growth<br><br>Sources: FRED, Federal Reserve Economic Data |
| Financial Condition | Risk Financial Condition<br>Credit Financial Condition<br>Leverage<br><br>Sources: FRED, National Financial Conditions Index |
| Monetary Policy Expectations | Effective Federal Funds Rate<br>Monetary Policy Expectations |

| | |
|---|---|
| | Sources: FRED, Federal Reserve Economic Data |
| Equity Earning Yield | Cyclically Adjusted PE Ratio (CAPE Ratio)<br>Sources: Online Data Robert Shiller |

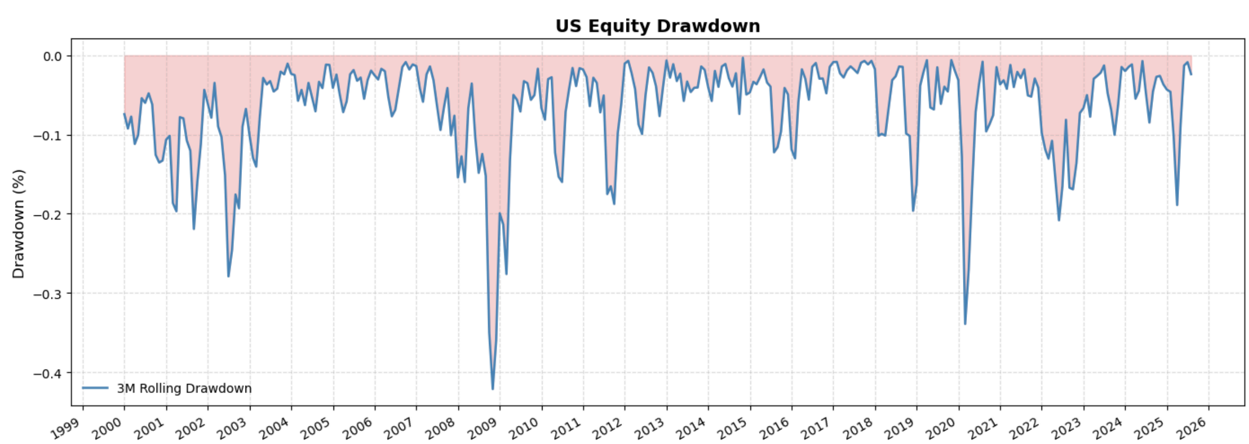
It is worth noting that the Cyclically Adjusted P/E Ratio (CAPE Ratio) is calculated based on S&P 500 constituents. In Stage Two, due to limited time and energy, we directly used factor data from the Kenneth R. French Data Library. While factors derived from the French library are based on a broader stock universe, which can enhance the diversity of our strategy, both datasets broadly represent the U.S. equity market and are highly correlated. Nonetheless, we acknowledge that the difference in underlying data sources may introduce some degree of discrepancy in our results.

In addition, although the original paper does not clearly specify this, based on several clues such as “Exhibit 9: Feature Importance of Selected Variables”, we infer that the macroeconomic inputs for the Stage One stacking model used to predict market regimes likely include three categories: Equity Valuation, Financial Conditions, and Monetary Policy Expectations. In contrast, for Stage Two, which predicts the next-period outperforming factors, all macroeconomic variables are utilized.

## US Equity Drawdown

The equity drawdown is calculated based on the S&P 500, in order to stay consistent with macroeconomic variables such as the Cyclically Adjusted P/E Ratio. Since these variables will serve as inputs for Stage One, which involves market regime clustering and regime prediction, it is important to ensure consistency among them. The specific calculation method is as follows:

$$RollingDrawdown(t) = \frac{P_t - Max_{t-63 \le s \le t} P_s}{Max_{t-63 \le s \le t} P_s}$$



## Financial Turbulence

Financial Turbulence is an indicator proposed by Kritzman and Li (2010), primarily designed to capture systemic risk in financial markets. Since this variable is used in Stage One to predict market risk regimes, we maintained consistency by computing it using data derived from the S&P 500. The calculation formula for Financial Turbulence is shown below.

$$d_t = (y_t - \mu)\Sigma^{-1}(y_t - \mu)^T$$

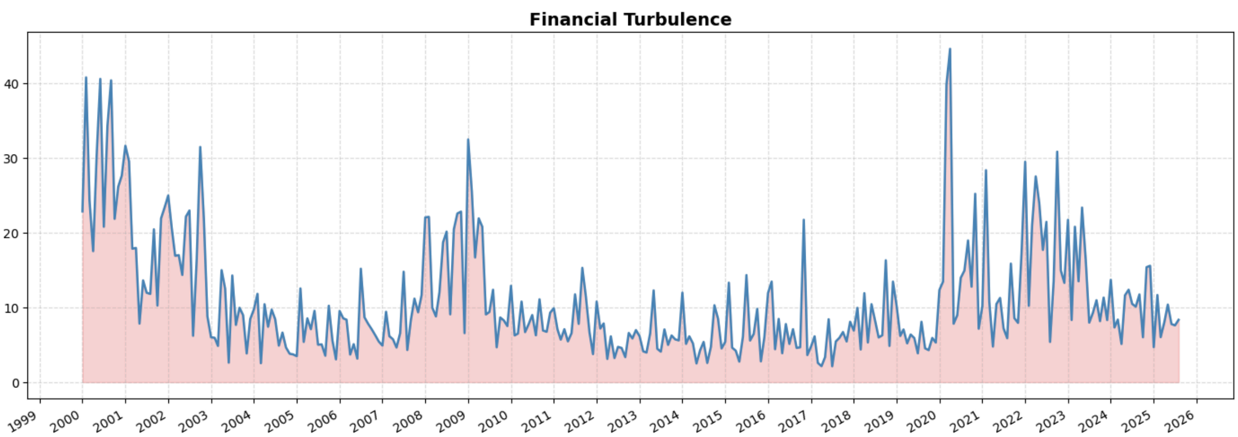
Where:

$d_t$ = financial turbulence during the time period t

$y_t$ = vector of asset returns for period t

$\mu$ = sample average vector for historical returns

$\Sigma$ = sample covariance matrix of historical returns




Based on the framework introduced by Kritzman and Li, we constructed the Financial Turbulence series using monthly returns from ten S&P 500 sector ETFs, Information Technology, Energy, Financials, Healthcare, Consumer Staples, Consumer Discretionary, Utilities, Industrials, Telecommunications, and Materials together with Treasury returns at three maturities (1-year, 2-year, and 10-year), as summarized in table above.

It is worth noting that the Communication Services Select Sector and the Real Estate Select Sector were introduced relatively late. We determined that interpolating their missing data would not be a robust solution. Therefore, in practice, we divided the Financial Turbulence series into two subperiods, calculated separately based on the availability of sector data.

Conceptually, the Financial Turbulence indicator is essentially a form of the Mahalanobis distance, which incorporates the covariance matrix into the distance calculation. It can be viewed as a modified version of the Euclidean distance. By introducing orthogonalization, it addresses the correlation among features, and by including standardization, it also resolves the issue of inconsistent scales and units across different features. In other words, it can be visualized as measuring the “distance” of a point from the center of an elliptical distribution, making it a theoretically sound measure of market systemic risk.

## Equity Style Factor Returns

In order to replicate the factor timing model in DiCiurcio et al. (2024), our study considers six equity style factor return series, which represent well-established compensated risk premia in equity markets. These factors come from the Kenneth R. French Data Library, one of the most widely referenced academic sources for factor investing research. The sample frequency is monthly, and the time horizon is January 2000 – Present.

The original paper constructs factor portfolios using Russell index constituent data, which requires extensive security-level filtering and portfolio reconstruction. Replicating this process would be computationally expensive and inefficient for our project. Instead, we use Fama-French factor returns from Kenneth R. French Data Library, which are widely accepted academic benchmarks (e.g., HML, SMB, Momentum, Quality). Although the data source differs, these factors capture the same style-based risk premia, allowing us to analyze regime-dependent factor performance in a transparent and consistent manner.

| **Factor Name** | **Data Source & Construction Summary** |
|---|---|
| Value (HML) | Return spread between high and low book-to-market firms (Fama–French model). Represents exposure to undervalued companies. |
| Growth (Small Low Book-to-Market) | Returns of small-cap firms with low book-to-market ratios. Captures “growth” style exposure. |
| Momtem (MOM) | Long high-momentum stocks (top 12-month performers), short low-momentum stocks. Matches the factor definition used in the paper. |
| Quality (RMW) | Return spread between firms with robust vs. weak operating profitability (proxy for quality factor). |
| Small-Cap (SMB) | Spread between returns of small-cap and large-cap firms (size effect). |
| Low Volatility (LoVol) | Portfolio of Stocks with lowest estimated daily return variance (constructed using 60-day trailing volatility). Represents defensive characteristics. |

# Methodology

## Stage 1: K-means Clustering and Market Regime Prediction by Stacking Model

### Overall structure

In Stage One, the modeling process can be summarized as follows: we first apply K-means clustering to the entire dataset to identify the distinct market regimes that exist over the full sample period, and then use various classification models to predict the market regime for the next period.

### Data preparation

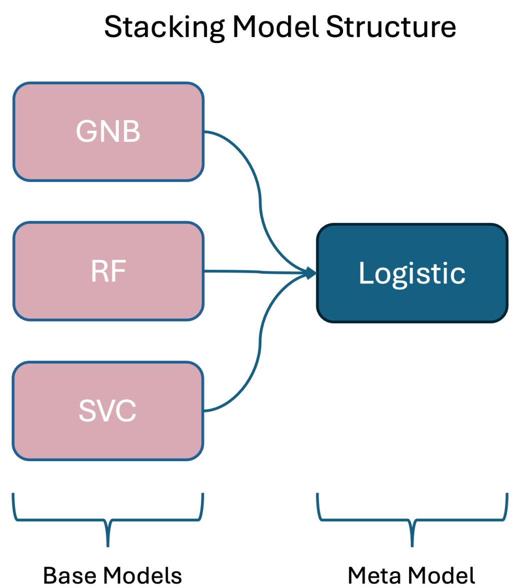
More specifically, in the K-means clustering stage, we use the three-month rolling maximum drawdown of the S&P 500 as the input variable. The Elbow Method is employed to determine the optimal number of clusters, which serves as the basis for our classification of market regimes.

In the subsequent classification models, we use macroeconomic variables, financial turbulence, and lagged maximum drawdown as input features. Unlike the original paper, we incorporate lagged terms of selected features in order to capture additional temporal information embedded in the data structure. Moreover, since the Gaussian Naive Bayes (GNB) model requires its input features to satisfy the assumption of conditional independence, a condition derived from Bayes' theorem, so it is important to address potential correlations among features. The original study does not explicitly explain how this issue is handled. Also, achieving true conditional independence in economic and financial data is extremely difficult in practice. To mitigate this problem, we employ methods like Principal Component Analysis to reduce correlations among features, thereby indirectly enhancing their independence.

In addition, during the actual prediction process, we merged the two market regime labels with relatively small proportions, which is quite different from the practice of the original paper. This decision was based on three considerations. First, in our experiments, we found that the model could identify the dominant regime quite well but often confused the two minority regimes. Second, in Stage Two of the original study, the authors also combined these two smaller regimes for convenience and to improve predictive accuracy. Third, we believe that our classification of market regimes is essentially based on a data-driven definition. Under different definitions, the two minority regimes may actually refer to the same type of market condition. Since this merging was already applied in the final usage stage, which is essentially the purpose of our Stage One setup, we believe it is reasonable to merge the labels earlier during prediction. Doing so helps improve both the accuracy and efficiency of our model.

## Comparison between different models

Another major difference between our approach and the original paper lies in the set of models used. The original study uses a stacking classification model composed of three base models (Random Forest, Gaussian Naive Bayes, and Support Vector Classification) and one meta-model, a logistic regression. In contrast, our replication experiment compares the predictive performance of the three base models individually with that of the stacking model. This comparison allows us to examine whether the stacking approach indeed achieves the theoretical advantage of combining the strengths of multiple models.




# Stage 2: Factor Performance Prediction using Random Forest

## Model chosen and data pre-process

In Stage 2, we aim to determine which equity style factor is most likely to outperform in the next month, conditional on the market regime predicted in Stage 1. Our final framework implements a single model — Random Forest for factor prediction,  consistent with the methodology presented in DiCiurcio et al. (2024). The model takes macroeconomic and financial variables, financial turbulence, equity drawdown, and the predicted market regime as features, and predicts the probability that a given factor will be the next-month winner within its regime.

The supervised learning target is a multi-label classification problem. For each month, we identify the winning factor as the one with the highest return among the six Fama–French style factors (Value, Growth, Momentum, Quality, Small-Cap, Low Volatility).  We label a factor as 1 if it is the best-performing factor in that month and label the remaining factors as 0.

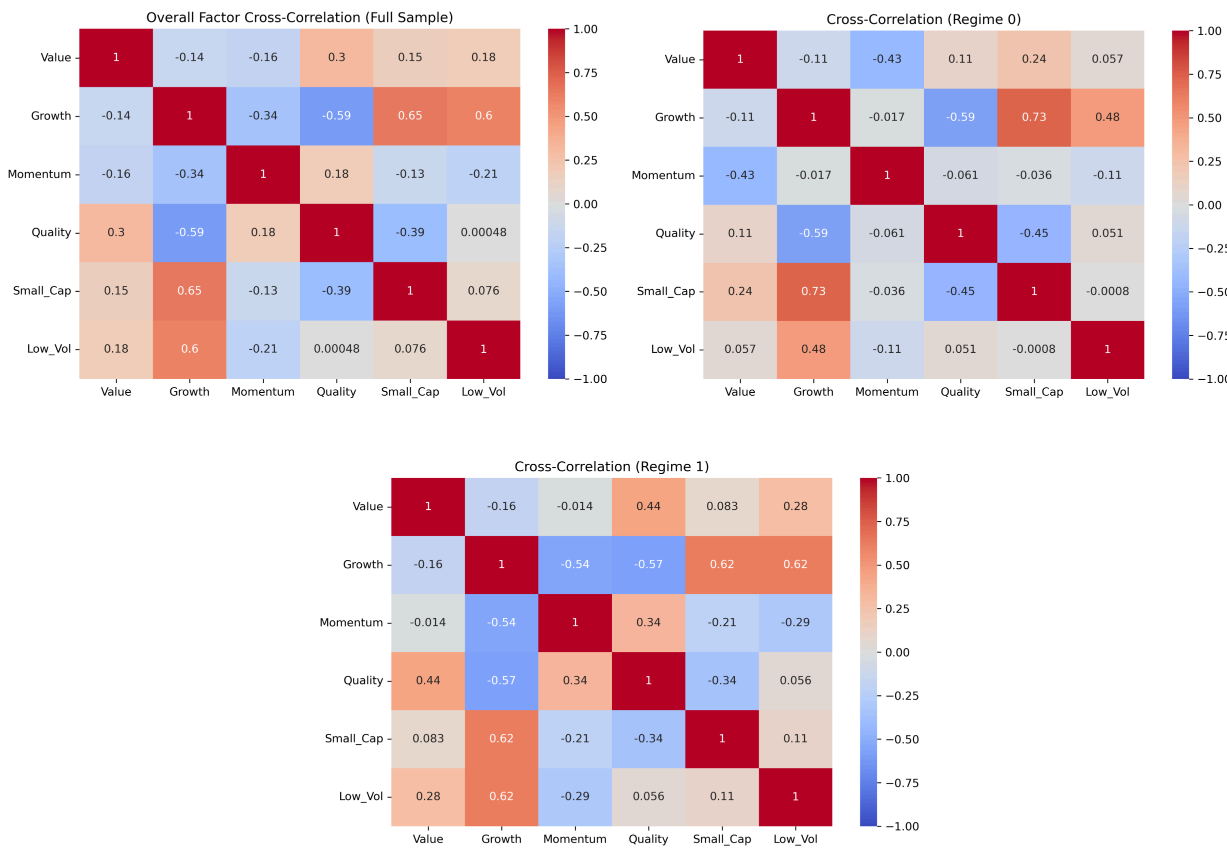
The Random Forest model learns to map macro-financial feature lagged by one period to next-month factor winners:

$$X_{t-1} \rightarrow y_t$$

We lag features intentionally to avoid look-ahead bias and leakage.

## Regime-specific model training

The original paper demonstrates that factor performance is regime dependent. To validate this effect, we first examined factor return correlation matrices across regimes (Figure below).The heatmaps show that factor co-variance changes materially between Regime 0 (normal periods) and Regime 1 (turbulent periods). For example, the correlation between Growth and Low_Vol is significantly higher in turbulent regimes, whereas Momentum becomes less correlated with the rest of the factors. These structural shifts indicate that the relative behavior of factors is not stable across regimes; therefore, using a single pooled model would mix different underlying data-generating processes.




To address this, we train a separate Random Forest model for each regime. When generating a prediction for month *t*, we use only data from the matching regime and only information available before month *t*. This ensures that the process mimics a real-time investment decision and aligns with the risk-management intuition of the original paper.

## Handling label imbalance

Consistent with the paper, our Stage-1 process generates three regimes (0, 1, 2). However, in our dataset, the number of observations in Regime 2 (high turbulence) is extremely small — making model training infeasible. To preserve model stability and still capture extreme market stress conditions, we follow a practical adjustment: Regime 2 events are merged into Regime 1, forming a single “turbulent regime” model, which preserves the regime-switching logic while ensuring sufficient training samples.

### Expanding (recursive) training window

Instead of using a fixed rolling window, we adopt an **expanding rolling training window**, where the model is retrained each month using *all historical data available up to that point*. In implementation, the training and prediction follow the structure below:

```
train = full_data.iloc[:t]      # expanding training set

test  = full_data.iloc[[t]]     # predict 1-step ahead
```

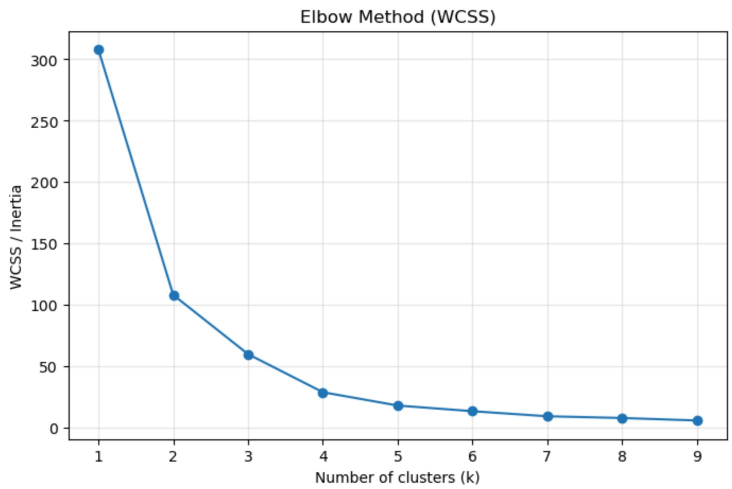
Using an expanding rolling window allows the model to learn from all available historical data up to the prediction month, which better reflects how investment research evolves over time. Unlike a fixed rolling window, the expanding window avoids unstable training caused by insufficient data—certain factors have very few “win” labels in short windows, which can lead to unreliable model estimates. Also,  the expanding setup strictly respects time directionality, ensuring that only past information is used to forecast future outcomes, effectively preventing data leakage. This enables true out-of-sample predictions and aligns with the “recursive real-time learning” approach used in the base paper.

# Results & Analysis

## Stage 1

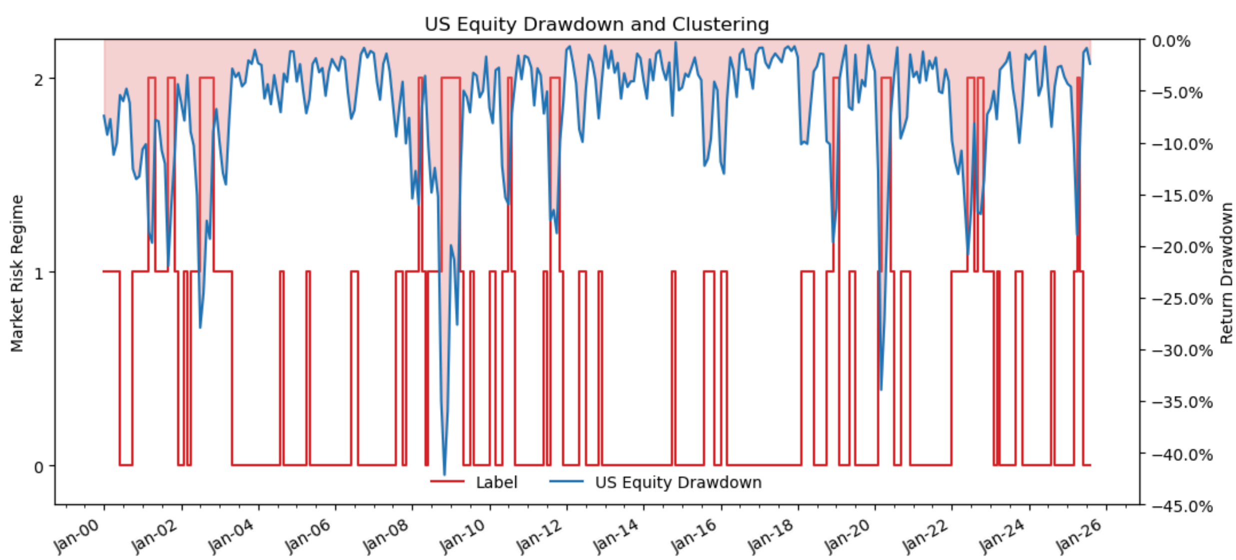
### K-means Clustering Result

As mentioned earlier in the Methodology section, the changes in distortion from our backtesting are shown below.



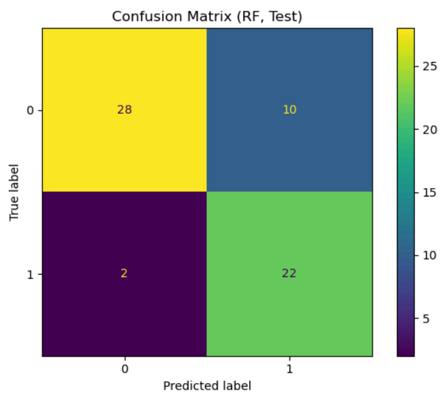

According to the Elbow Method, we selected K = 3. As a result, we obtained three clusters. The label Normal, which represents periods when the market experienced little overall drawdown, accounts for 0.626 of the observations. The label Correction, indicating moderate market drawdowns, accounts for 0.279, and the label Bear, representing strong market downturns, accounts for 0.094. Their transition matrix and the picture with the drawdown data are shown below.

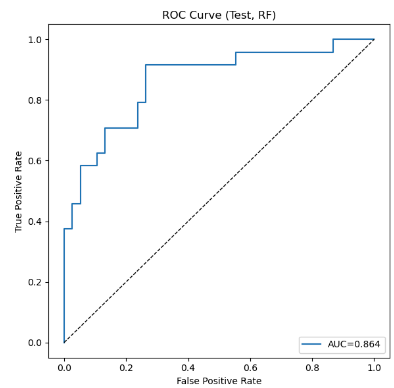
| | **Normal** | **Correction** | **Bear** |
|---|---|---|---|
| **Normal** | 0.848958 | 0.145833 | 0.005208 |
| **Correction** | 0.337209 | 0.534884 | 0.127907 |
| **Bear** | 0.034483 | 0.379310 | 0.586207 |

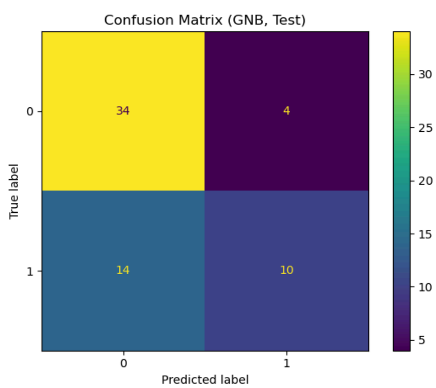


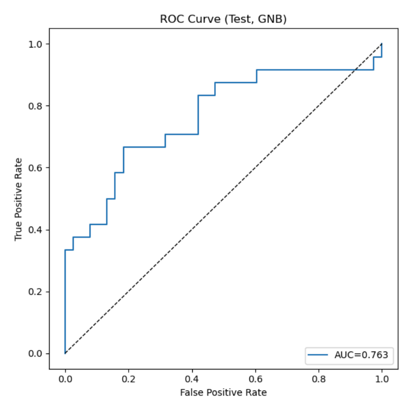
## Market Regime Prediction Result

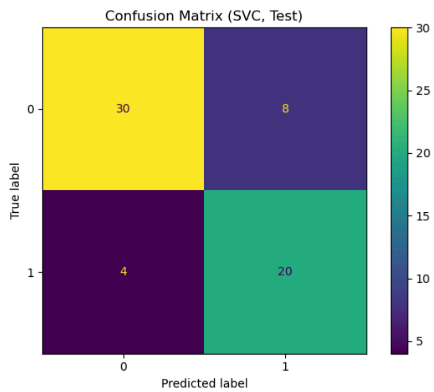
After merging the Correction and Bear labels, we applied SMOTE again, which essentially performs convex interpolation, to address the imbalance among the remaining classes. For data splitting, we used a time-series-safe method to ensure that no future information was used during model training. In terms of hyper parameter tuning, we explored the largest grid possible within our computational limits to help find the optimal set of parameters. The test set accounted for 20% of the total data. The prediction results of the four models are summarized below.

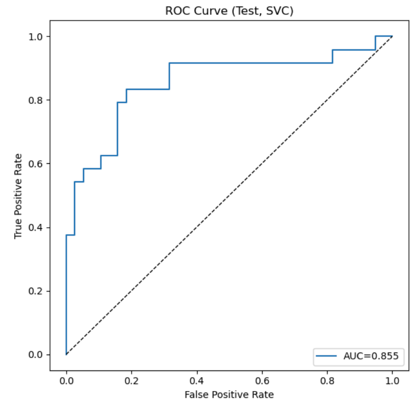


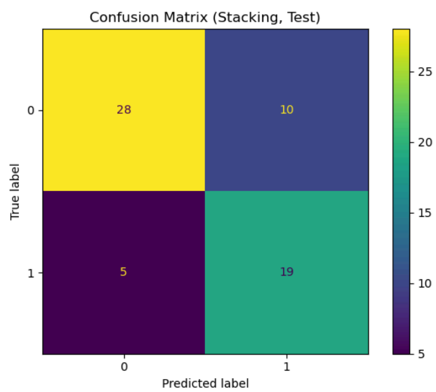



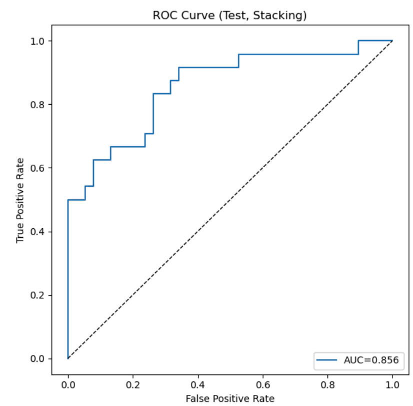






As we can see, all four models achieved strong predictive performance. Except for the GNB model, which clearly performed worse than the other three, the differences among the remaining models were quite small. With a larger sample size, these differences might become more noticeable, but since most macroeconomic data are collected monthly and often revised, using a much larger dataset is not feasible. Nevertheless, the prediction accuracy of our four models are acceptable, which provides a solid foundation for Stage Two.

# Stage 2

## Factor Prediction Accuracy

The table below summarize the prediction accuracy of the model for each factor, as well as how often each factor actually wins:

| Factor | Accuracy | Pred Cound | Actual Win % |
|---|---|---|---|
| Value | 0.556 | 9 | 0.118 |
| Growth | 0.429 | 28 | 0.363 |

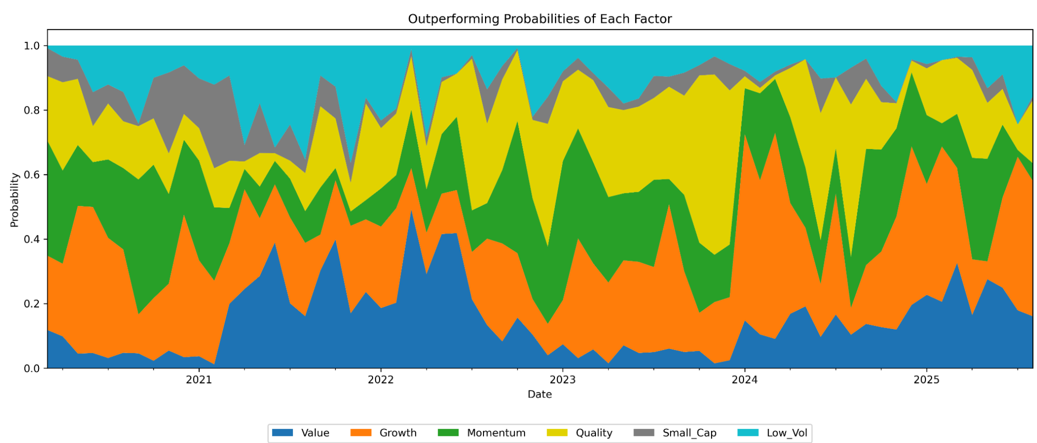
| | | | |
|---|---|---|---|
| Momentum | 0.267 | 15 | 0.206 |
| Quality | 0.273 | 11 | 0.127 |
| Small-Cap | 0.000 | 1 | 0.052 |
| Low Vol | 0.500 | 2 | 0.134 |

**Overall model accuracy: 0.379 (vs. random baseline: 0.167)**

From the results of the factor-level prediction, we can see that the model outperformed random selection, which confirms that the model possesses meaningful predictive power. Growth and Momentum are the most frequently predicted to outperform. On the other hand, Small-Cap is seldom predicted to be the winner, which is consistent with its generally weak performance over our sample period. Value and Low Volatility have moderate success rates despite being less frequently selected, indicating that the model is still able to identify their relative advantage when more defensive styles are favored by market uncertainty or valuation dispersion.

## Probability-Based Factor Allocation (Stacked Probability Chart)

The stacked plot visualizes how factor outperforming probabilities evolve through time:




Interpretation of the plot: Higher area → higher probability the factor will outperform next month.

From the stacked probability visualization, we can see that Growth and Momentum occupy larger shares for a long period, while Low Volatility and Quality show noticeable increases at other times. We also observe that sharp shifts in the probability distribution often align with changes in the regime labels produced by Stage 1, which suggests the model is capturing factor-rotation signals over time. Together, these patterns support the view that factor leadership is time-varying and that its short-term shifts can be at least partially anticipated by our probability-based predictor.

## Business Cycle Interpretation

Using predicted rolling probabilities and realized regime labels, we compute the average factor ranking within each regime.
The results match the patterns discussed in the original paper:

| True regime | Value | Growth | Momentum | Quality | Small_cap | Low_vol |
|---|---|---|---|---|---|---|
| 0 | - | + | + | - | - | + |
| 1 | + | - | - | + | + | - |

("+" indicates the factor tends to outperform and therefore receives higher model-assigned weights.)

To interpret factor rotations, we categorize the model's two regimes based on their risk characteristics. Regime 0 corresponds to a low-risk and stable environment, where equity drawdowns and turbulence are mild. Regime 1 corresponds to a high-risk and stressed environment, with elevated volatility and deeper drawdowns. While our model does not forecast macroeconomic cycles, these risk profiles loosely align with common business cycle patterns—Regime 0 resembles expansion/early slowdown, and Regime 1 resembles contraction/recovery.

Using the factors' average performance within each regime, we observe distinct rotation patterns. In Regime 0, our model predicts that Growth, Momentum, and Low Volatility tend to outperform, which is consistent with investor preference for risk-on factors during more stable environments where trends persist. In Regime 1, the model shifts toward Value, Quality, and Small-Cap. These patterns reflect how factor leadership changes depending on the risk environment captured by our regime classifier. Importantly, this regime-specific factor rotation is consistent with the findings of the base paper, which also reports that risk-on factors dominate in stable periods while defensive factors lead during stressed conditions. Overall, the results support the idea that factor returns are regime-dependent, and that incorporating regime information helps the model dynamically adjust factor allocations.

## Conclusion for Stage 2 results and analysis

Stage 2 demonstrates that factor returns exhibit predictive structure, particularly when conditioning on market regimes. By assigning probability-based weights rather than selecting a single factor, the model is able to adjust allocations over time and reflect factor rotation. The results are consistent with both academic literature and the findings of the base paper, indicating that market environments influence which factors tend to outperform. Overall, the evidence supports our hypothesis that incorporating regime awareness and machine learning improves factor selection compared with a simple equal-weight benchmark.

# Portfolio Simulation / Backtesting

## Equal-Weight Benchmark (Baseline Strategy)

As a benchmark, we construct a simple equal-weight portfolio (1/N), assigning the same weight to all factors every month. This benchmark represents a simple diversification approach and does not incorporate any factor prediction, timing, or regime awareness.

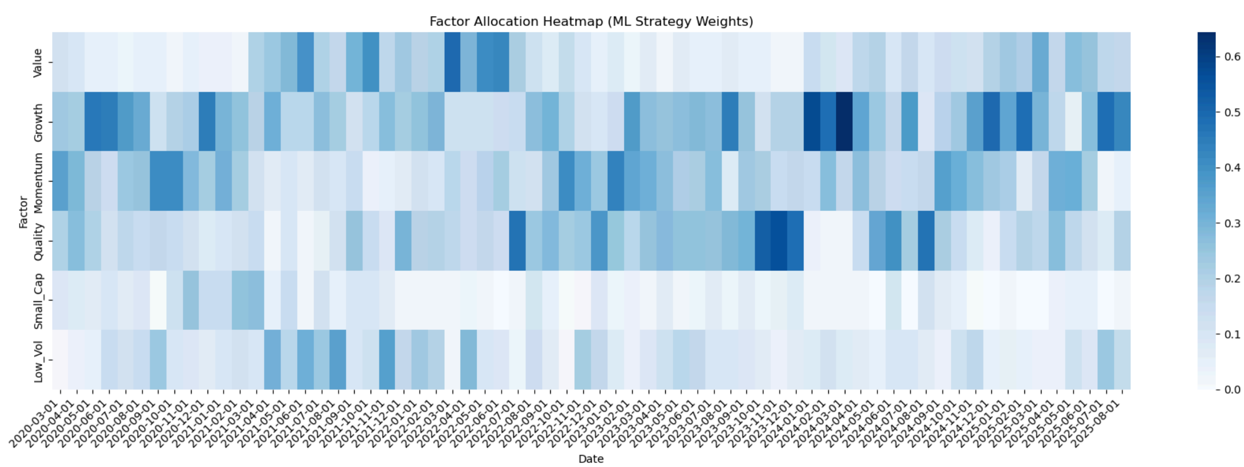
## ML Strategy (Probability-Weighted Factor Portfolio)

To evaluate whether machine learning–based factor selection can improve portfolio performance, we convert the model's predicted outperforming probabilities into portfolio weights:

$$\omega_{f,t} = P(f,t) / \sum_{j} P(j,t)$$

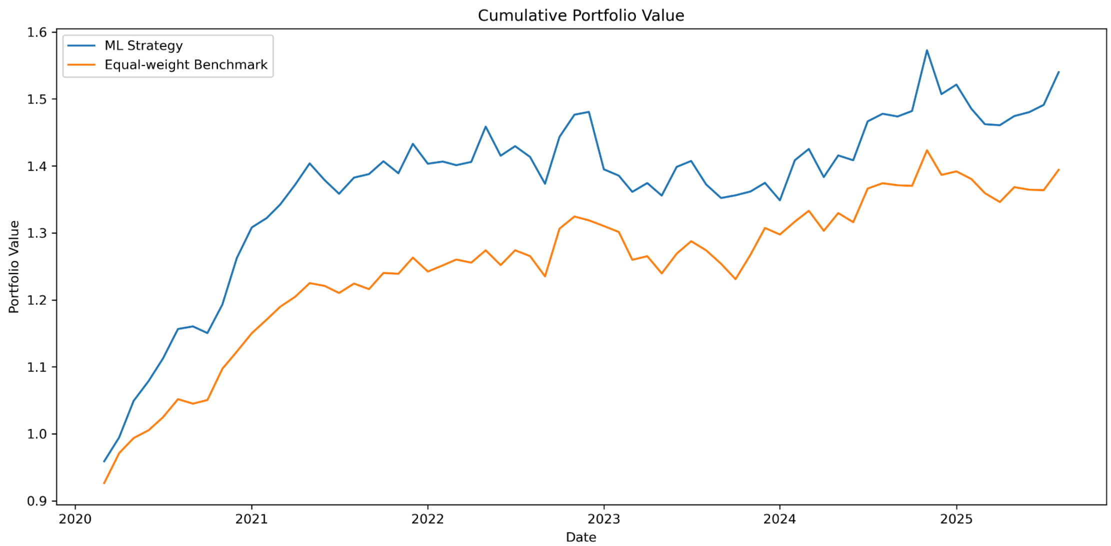
(P(f,t) is the predicted probability that factor f will outperform, w is the resulting portfolio weight)

This ensures higher-conviction factors receive larger allocations in the target month. The result is a dynamic probability-weighted factor portfolio that adjusts based on new predictions each month, instead of holding static exposures.




## Performance Metrics (Return, Volatility, Sharpe, Drawdown)

We run the backtest using monthly factor returns from March 2020 to August 2025. The cumulative performance chart shows that the ML strategy remains consistently above the equal-weight benchmark through most of the test period, ending with a portfolio value of 1.54 versus 1.39 for the equal-weight approach.




Performance statistics lead to the same conclusion. The ML strategy generates a higher total return (54.0% vs. 39.4%) and a higher annualized return (8.17% vs. 6.23%). Although annualized volatility increases slightly, the ML portfolio achieves a superior Sharpe ratio (0.94 vs. 0.86) and a higher win rate (63.6% vs. 54.6%). While maximum drawdown is marginally higher, the additional drawdown is compensated by stronger returns and better risk-adjusted performance. Additionally, the PnL curve shows that the strategy earned most of its profits between 2020 and June 2022. Afterward, it experienced slightly higher volatility than the baseline but managed to hold on to the gains made earlier.

## Interpretation — Why ML Performs Differently

The equal-weight benchmark assumes all factors are equally attractive at all times, ignoring changes in market dynamics. In contrast, our ML strategy adjusts factor exposures based on predicted relative strength and captures factor rotation. By allocating more weights to factors that are predicted to outperform, the model takes advantage of regime-dependent factor behavior. The backtest results show that this adaptive allocation leads to higher returns and higher Sharpe ratios, confirming that incorporating machine learning and regime awareness adds value beyond a static diversification approach.

# Conclusion

Our backtesting results support our working hypothesis that a factor rotation strategy based on predicted market regimes and machine learning models achieves significantly better returns than the baseline. In addition, even when we only consider the first stage of the model, accurately predicting the next market style can provide useful references and additional information for risk management.

Moreover, this model shows promising flexibility. We can freely combine different classification models in both Stage One and Stage Two to better adapt to specific data characteristics and improve robustness. We can also include different styles of factors in Stage Two to find strategies suitable for more specific market segments.

However, the model also has some clear limitations. The first stage relies heavily on macroeconomic data to predict the next market regime, but most macroeconomic indicators are released monthly, quarterly, or even annually. This low data frequency leads to serious lag effects within the strategy itself. As a result, the model may perform poorly during periods of rapid market-style shifts or when regime changes happen too quickly.

# Appendix

## Machine Learning Models and Techniques

### K Means Clustering (Unsupervised Learning, Stage 1)

The component of Stage 1 is to determine the types of Market Regimes. This was used by implementing K-means clustering with elbow method optimization. K means clustering is an unsupervised model because the machine learning algorithm uses unlabeled training data to assign the labels, which in our case is different market regimes, and groups the data by itself. The algorithm is shown below:

$$minimize \sum_{i} \sum_{j} ||X_i - u_j||^2$$

where $||X_i - u_j||$ is the euclidean distance between data point $X_i$ and centroid $u_j$

### Elbow Method (Optimization Technique, Stage 1)

This is the optimization method used for K-means clustering. The optimization is done by running K means clustering with several ranges of K values. Then it calculates the Within-Cluster Sum of Squares (WCCS) for each value K which is shown below:

$$WCCS = \sum_{i=1}^{k} \sum_{x\ in\ cluster\ i} distance(x, centroid_i)^2$$

WCCS is the sum of the squared distances between each data point $x$ and the centroid of the cluster. Lower WCCS would mean tighter, more coherent clusters. As the number of clusters K increases, the WCCS will decrease but with diminishing returns. Plotting the WCCS for a range of K values will show an initial sharp decrease in the first few values of K but will have increasingly flatter slopes. The point of optimal clusters, the elbow, is where there is the greatest change in slope. The name is due to the fact that the WCCS plot somewhat resembles a slightly bent human arm with an elbow point.

### Stacking Ensemble Classification (Supervised Learning, Stage 1)

After determining the market regimes, the prediction of market regimes was done by the stacking model, which contains 2 layers. The first layer consists of an ensemble of 3 different classifiers: Random forest (RF), Support Vector Classifier (SVC), and Gaussian Naive Bayes (GNB). The reason for using an ensemble is because in machine learning, a mixture of different "weaker" models can often outperform a single "strong" model for prediction or classification tasks. The output of the 3 ensemble model was then used as an input to the second layer, which is the Logistic Regression. The Logistic Regression natively outputs a probability which suits the purpose of predicting the market regime task for stage 1. Logistic Regression learns the optimal weights of the inputs from the 3 ensembles and then minimizes the cross entropy loss by the following equation:

$$L(w) =- \sum[y_i * log(y_{pred_i}) + (1 - y_i) * log(1 - y_{pred_i})$$

$y_i$ is the true label of the i-th sample $y_{pred}$ is the predicted probability.
The final prediction probability is calculated as the following:

$$y_{pred} = \sigma(w_1 * RF(x) + w_2 * SVC(x) + w3 * GNB(x)$$

$\sigma$ is the sigmoid function which maps the weighted sum of the 3 ensemble classifiers into a probability output. By utilizing the ensemble with Logistic Regression, we are able to capture linear and non linear complex relationships to better predict the market regime.

## Principal Component Analysis (Dimension Reduction Technique, Stage 1)

When working with a large set of data and features, it is important to be aware of features with complete or near complete linear relationships. This is because multiple features with extremely high correlation (not to be confused with autocorrelation) may cause multicollinearity, which negatively impacts model performance especially in regression models. When variables are highly correlated, small changes in the data may cause significant fluctuations in the regression coefficients, meaning unstable coefficients which result in reduced interpretability of the model. Moreover, fluctuations in the coefficients may also result in unreliable model performance and predictions. Multicollinearity can be detected using the Variance Inflation Factor:

$$VIF_i = \frac{1}{1-R_i^2}$$

where $R^2$ is the coefficient of the determination of the variable. Principal Component Analysis resolves the issue of multicolinearity by creating new uncorrelated variables called principal components and used in the regression model, resulting in reduced dimensionality and multicolinearity which also allows for faster computation due to a smaller dimension for calculation.

## Random Forest Classification (Supervised Learning, Stage 2)

In Stage 2, given a target of the 6 equity factors, Random Forest was used to output the winning equity factor. The dataset used in Stage 1, was divided into subdata corresponding to each market regime identified in Stage 1. This was done in order to conduct regime specific model training. Each subdata was an input to a separate Random Forest Classification model in order to determine the winning equity factor for each market regime. Random Forest is an ensemble of decision trees. An ensemble of decision trees, compared to using a single very deep decision tree, prevents overfitting with the given data which allows the model to perform as well in out of sample data and prediction. A random forest model works by reducing the purity of a node. The purity is a measure of how well the node represents a single class. The Gini Impurity is frequently used to determine the purity and is the following:

$$G = 1 - \sum_{i=1}^{C} p_i^2$$

C is the number of classes and $p_i$ is the proportion of elements belonging to class i in the dataset. The dataset utilized is imbalance with certain market regimes occurring rarer than others. Therefore, balanced weights were used in the Random Forest implementation where class weights calculated for the market regimes were inversely proportional to the frequency of the market regimes. This is typically done when dealing with imbalanced datasets.

## Lagging Features (Stage 1, 2)

Lagged Features are created by taking the value of a variable from a previous time period and using it as a predictive feature in the current period. This is essential for times series data as using a factor from a future data to predict a future outcome is essentially cheating. Moreover, lagging features also help with autocorrelation, which is the following:

$$p_k = \frac{Cov(X_t, X_{t-k})}{\sqrt{Var(X_t)*Var(X_{t-k})}}$$

Autocorrelation is the correlation of a time series with its lagged versions. In other words, autocorrelation essentially measures the relationship between a variable's current value and its own past values. A high autocorrelation means what happened recently would be likely to continue into the future. Therefore, a lagged feature with high autocorrelation would indicate a potential to be a good predictor. For example, financial turbulence and market drawdown shows strong autocorrelation in Stage 1 and thus helpful for identifying and predicting market regimes. Also, seasonal autocorrelation can be utilized to determine the winning equity factor for different market regimes due to economic cycles such as shown in Stage 2 of the paper, thus highlighting the cruciality of lagged features.

# Reference

DiCiurcio, K. J., Wu, B., Xu, F., Rodemer, S., & Wang, Q. (2024). Equity Factor Timing: A Two-Stage Machine Learning Approach. *Journal of Portfolio Management, 50* (3), 132–148.

FAMA, E. F., & FRENCH, K. R. (1992). The Cross-Section of Expected Stock Returns. *The Journal of Finance (New York)*, *47*(2), 427–465. https://doi.org/10.1111/j.1540-6261.1992.tb04398.x

Kritzman, M., & Li, Y. (2010). Skulls, Financial Turbulence, and Risk Management. *Financial Analysts Journal*, *66*(5), 30–41. https://doi.org/10.2469/faj.v66.n5.3

Sharpe, W. F. (1964). Capital Asset Prices: A Theory of Market Equilibrium under Conditions of Risk. *The Journal of Finance (New York)*, *19*(3), 425. https://doi.org/10.2307/2977928