Weekly Report 26 (10/22/2024 - 10/28/2024)

Project: Econometric Analysis on stock price vs Market Index (S&P 500)

Introduction

The primary focus of this project is to analyze the relationship between individual stock returns and market movements, particularly as represented by the S&P 500 index. Using a variety of econometric models and time-series techniques, this study aims to provide a comprehensive understanding of stock behavior in relation to market dynamics. By leveraging models such as the Capital Asset Pricing Model (CAPM), ARIMA-GARCH, and Cointegration with Error Correction, the analysis aims to uncover insights into stock volatility, long-term relationships with the market, and optimal trading strategies.

```
merged_data
```

	Date	Stock	Adj Close	Stock_Returns	SP500_Returns
1	2011-01-03	AAPL	9.939775	0.021732	0.011315
2	2011-01-04	AAPL	9.991650	0.005219	-0.001313
3	2011-01-05	AAPL	10.073380	0.008180	0.005007
4	2011-01-06	AAPL	10.065243	-0.000808	-0.002123
5	2011-01-07	AAPL	10.137320	0.007161	-0.001845
...	...	...	...	...	...
34237	2024-09-24	TSLA	254.270004	0.017080	0.002511
34238	2024-09-25	TSLA	257.019989	0.010815	-0.001861
34239	2024-09-26	TSLA	254.220001	-0.010894	0.004039
34240	2024-09-27	TSLA	260.459991	0.024546	-0.001253
34241	2024-09-30	TSLA	261.630005	0.004492	0.004237

34232 rows × 5 columns

Objective

The goal of this project is to:

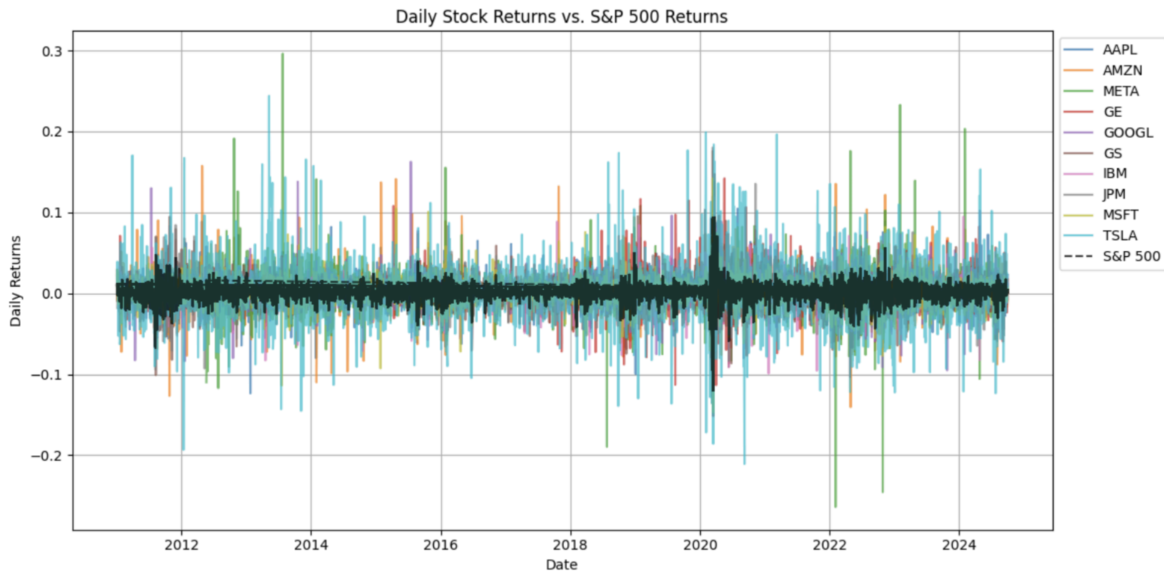
1. Assess how different stocks behave in relation to the broader market, identifying their sensitivity to market changes using CAPM.
2. Model and predict stock returns and volatility over different time horizons using ARIMA-GARCH models.
3. Identify long-term equilibrium relationships between stock prices and the S&P 500 through cointegration analysis.
4. Develop a robust trading strategy based on the insights derived from CAPM, GARCH-based volatility analysis, and cointegration dynamics.
5. Apply additional analysis method including Dynamic Factor, Bayesian VAR and Markov Switching method
6. Evaluate the performance of the trading strategy against market benchmarks to assess its potential in real-world applications.

Weekly Report 26 (10/22/2024 - 10/28/2024)

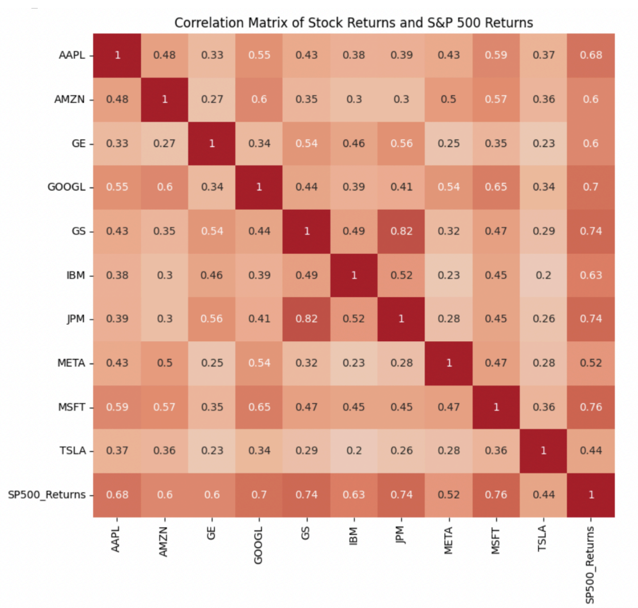
Exploratory Data Analysis (EDA)

Before diving into the modeling phase, an extensive Exploratory Data Analysis (EDA) was conducted to better understand the characteristics of the stock returns and their relationships with the S&P 500. EDA is a crucial step in uncovering patterns and guiding the selection of appropriate models for further analysis. Key methods used include:

- Distribution Analysis: Analyzing the distribution of stock returns helps to identify deviations from normality, skewness, and kurtosis. This analysis highlights stocks that might carry higher risk due to extreme price movements, allowing us to tailor our models accordingly.



- Correlation Analysis: Assessing correlations between stock returns and the S&P 500 revealed stocks that move closely with the market, indicating higher systemic risk, as well as those that are more independent, offering diversification benefits.



Weekly Report 26 (10/22/2024 - 10/28/2024)

CAPM Model Analysis

The Capital Asset Pricing Model (CAPM) was used to assess how individual stocks respond to market movements. The CAPM formula is defined as: $Ri=\alpha+\beta \cdot Rm+\epsilon$

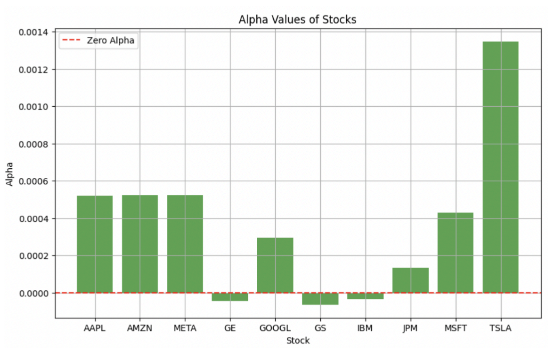
- Ri: Stock returns
- Rm: Market returns (S&P 500)
- α: The stock's ability to generate returns independent of the market (excess returns)
- β: Sensitivity of the stock to market movements
- ϵ: Error term

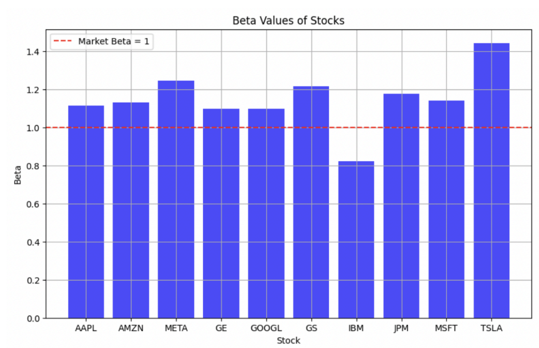
CAPM Results for Multiple Stocks:

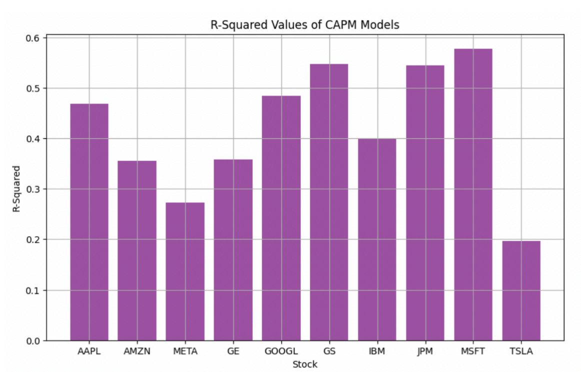
	Stock	Alpha	Beta	R_squared
0	AAPL	0.000519	1.114523	0.468403
1	AMZN	0.000524	1.132112	0.354906
2	META	0.000523	1.244323	0.272882
3	GE	-0.000045	1.099224	0.358031
4	GOOGL	0.000296	1.098812	0.483753
5	GS	-0.000064	1.215449	0.546709
6	IBM	-0.000034	0.821417	0.398504
7	JPM	0.000135	1.175571	0.544304
8	MSFT	0.000430	1.140282	0.577600
9	TSLA	0.001348	1.443690	0.196012

Key insights:

- Alpha Analysis: Stocks with positive alphas, like TSLA, indicate a history of outperforming the market, making them attractive for growth-oriented investors.
- Beta Analysis: Stocks such as AAPL, AMZN, and META have betas greater than 1, indicating that they are more volatile than the market. These stocks can potentially offer higher returns during market upswings but also carry more risk during downturns.
- R-squared Values: Higher R-squared values in stocks like MSFT and JPM suggest that a significant portion of their return variance can be explained by market movements, making CAPM a suitable model for these stocks.





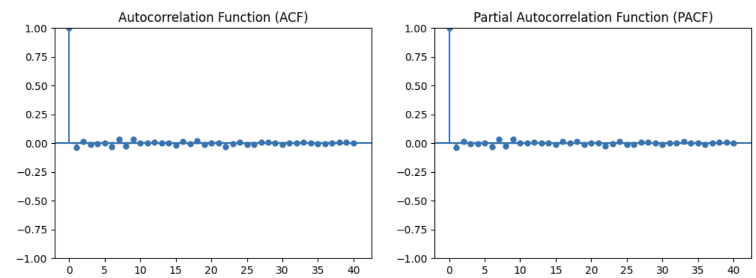


Weekly Report 26 (10/22/2024 - 10/28/2024)

ARIMA-GARCH Model

To better capture the time series dynamics of stock returns, ARIMA-GARCH models were employed:

- ACF and PACF Graph



- ARIMA Model: The AutoRegressive Integrated Moving Average (ARIMA) model is used to account for trends and autocorrelation in time series data. The chosen order of ARIMA(0,0,0) in this context treats the stock returns as stationary without relying on past values for predictions.
- GARCH Model: The Generalized Autoregressive Conditional Heteroskedasticity (GARCH) model is used to model time-varying volatility, capturing periods of heightened risk. The GARCH(1,2) model, which combines past residuals and variances, was particularly effective in modeling volatility clustering.

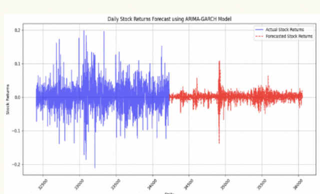
<u>Forecasting Performance</u>

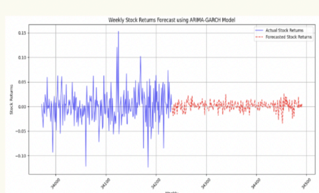
- Daily, Weekly, and Monthly Forecasts: The ARIMA-GARCH model provided reasonable forecasting accuracy, with Root Mean Square Error (RMSE) values decreasing as the forecast period extended from daily to weekly, indicating better performance in smoothing over longer time horizons.

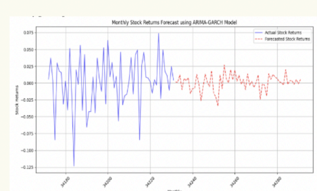
```
Best ARIMA order: (0, 0, 0) with AIC: -182334.19571241707
                               SARIMAX Results
==============================================================================
Dep. Variable:          Stock_Returns   No. Observations:                34232
Model:                          ARIMA   Log Likelihood               91170.098
Date:                Sun, 20 Oct 2024   AIC                        -182334.196
Time:                        08:04:23   BIC                        -182308.873
Sample:                             0   HQIC                       -182326.121
                              - 34232
Covariance Type:                  opg
=================================================================================
                    coef    std err          z      P>|z|      [0.025      0.975]
---------------------------------------------------------------------------------
const             0.0004   9.24e-05      3.865      0.000       0.000       0.001
SP500_Returns     1.1471      0.006    188.855      0.000       1.135       1.159
sigma2            0.0003   6.31e-07    451.329      0.000       0.000       0.000
===================================================================================
Ljung-Box (L1) (Q):                   0.08   Jarque-Bera (JB):            726505.24
Prob(Q):                              0.78   Prob(JB):                         0.00
Heteroskedasticity (H):               1.50   Skew:                             0.75
Prob(H) (two-sided):                  0.00   Kurtosis:                        25.52
===================================================================================
```

```
Best GARCH order: (1, 2) with AIC: -193174.1999564076
                     Constant Mean - GARCH Model Results
==============================================================================
Dep. Variable:                   None   R-squared:                       0.000
Mean Model:             Constant Mean   Adj. R-squared:                  0.000
Vol Model:                      GARCH   Log-Likelihood:                96592.1
Distribution:                  Normal   AIC:                          -193174.
Method:            Maximum Likelihood   BIC:                          -193132.
                                        No. Observations:                34232
Date:                Sun, Oct 20 2024   Df Residuals:                    34231
Time:                        08:04:44   Df Model:                            1
                                 Mean Model
=============================================================================
                 coef    std err          t      P>|t|      95.0% Conf. Int.
-----------------------------------------------------------------------------
mu         4.9934e-06  7.883e-05  6.334e-02      0.949 [-1.495e-04,1.595e-04]
                              Volatility Model
============================================================================
                 coef    std err          t      P>|t|     95.0% Conf. Int.
----------------------------------------------------------------------------
omega      5.6917e-06  1.763e-12  3.229e+06      0.000 [5.692e-06,5.692e-06]
alpha[1]       0.0500  9.203e-04     54.331      0.000 [4.820e-02,5.180e-02]
beta[1]        0.4650      0.121      3.830  1.281e-04     [  0.227,  0.703]
beta[2]        0.4650      0.122      3.800  1.447e-04     [  0.225,  0.705]
============================================================================
```

<u>Forecasting in 5 Year (Daily, Weekly, Monthly) with RMSE</u>







Daily RMSE:
0.034398980025610176

Weekly RMSE:
0.03046020289965815

Monthly RMSE:
0.031244400901975625

Weekly Report 26 (10/22/2024 - 10/28/2024)

Cointegration Analysis with Error Correction Model (ECM)

Cointegration analysis was conducted to identify long-term relationships between individual stock prices and the S&P 500:

- Cointegration Tests: These tests help determine whether pairs of time series (e.g., stock prices and market index) have a stable, long-term equilibrium relationship despite short-term fluctuations.
- Key Results: Stocks such as AAPL, AMZN, META, and TSLA exhibited strong cointegration with the S&P 500, with p-values below the 0.05 threshold, indicating that they tend to move together over time.
- Error Correction Model (ECM): The ECM was applied to model the short-term dynamics that adjust deviations from this long-term equilibrium. By analyzing the residuals of cointegrated series, the ECM provides signals (buy/sell/hold) based on how quickly prices return to their long-term relationship.

```
AAPL Cointegration Test with S&P 500:
Cointegration Test Statistic: -56.133144311630566
P-Value: 0.0
Critical Values (1%, 5%, 10%): [-3.89961084 -3.33789803 -3.04567707]
AAPL and S&P 500 are cointegrated at the 5% significance level.

AMZN Cointegration Test with S&P 500:
Cointegration Test Statistic: -42.900896219777366
P-Value: 0.0
Critical Values (1%, 5%, 10%): [-3.89961084 -3.33789803 -3.04567707]
AMZN and S&P 500 are cointegrated at the 5% significance level.

META Cointegration Test with S&P 500:
Cointegration Test Statistic: -28.37856348578412
P-Value: 0.0
Critical Values (1%, 5%, 10%): [-3.89996611 -3.338096   -3.04581445]
META and S&P 500 are cointegrated at the 5% significance level.

GE Cointegration Test with S&P 500:
Cointegration Test Statistic: -18.021916926586503
P-Value: 1.8379320293840524e-29
Critical Values (1%, 5%, 10%): [-3.89961084 -3.33789803 -3.04567707]
GE and S&P 500 are cointegrated at the 5% significance level.

GOOGL Cointegration Test with S&P 500:
Cointegration Test Statistic: -57.46531795417587
P-Value: 0.0
Critical Values (1%, 5%, 10%): [-3.89961084 -3.33789803 -3.04567707]
GOOGL and S&P 500 are cointegrated at the 5% significance level.

GS Cointegration Test with S&P 500:
Cointegration Test Statistic: -59.94018769145704
P-Value: 0.0
Critical Values (1%, 5%, 10%): [-3.89961084 -3.33789803 -3.04567707]
GS and S&P 500 are cointegrated at the 5% significance level.

IBM Cointegration Test with S&P 500:
Cointegration Test Statistic: -42.32852801606174
P-Value: 0.0
Critical Values (1%, 5%, 10%): [-3.89961084 -3.33789803 -3.04567707]
IBM and S&P 500 are cointegrated at the 5% significance level.

JPM Cointegration Test with S&P 500:
Cointegration Test Statistic: -23.265792406983252
P-Value: 0.0
Critical Values (1%, 5%, 10%): [-3.89961084 -3.33789803 -3.04567707]
JPM and S&P 500 are cointegrated at the 5% significance level.

MSFT Cointegration Test with S&P 500:
Cointegration Test Statistic: -42.9728163277309
P-Value: 0.0
Critical Values (1%, 5%, 10%): [-3.89961084 -3.33789803 -3.04567707]
MSFT and S&P 500 are cointegrated at the 5% significance level.

TSLA Cointegration Test with S&P 500:
Cointegration Test Statistic: -60.05073371958149
P-Value: 0.0
Critical Values (1%, 5%, 10%): [-3.89961084 -3.33789803 -3.04567707]
TSLA and S&P 500 are cointegrated at the 5% significance level.
```

Further Analysis

Bayesian VAR (BVAR)

- Purpose: Bayesian Vector Autoregression (BVAR) models help capture the dynamic interactions between stock returns and S&P 500 returns over time, adding a Bayesian layer to traditional VAR models for better handling of parameter uncertainty.
- Forecasting: A VAR model was fitted using stock and S&P 500 returns data, with an optimal lag order of 2 determined by AIC. The model was then used to forecast returns for the next 5 periods.
- Key Results: The forecasts provide insights into expected short-term movements in stock and market returns, which can be valuable for immediate trading decisions.
- Implications: These forecasts allow investors to adjust their positions based on expected market movements, helping them align their strategies with anticipated short-term trends.

Weekly Report 26 (10/22/2024 - 10/28/2024)

```
[78] #Bayesian VAR (BVAR)

     # Define the data including stock and S&P 500 returns
     var_data = merged_data[['Stock_Returns', 'SP500_Returns']]

     # Fit a VAR model (simplification of BVAR)
     var_model = VAR(var_data)
     bvar_results = var_model.fit(maxlags=2)

     # Forecast using the fitted VAR model for the next 5 periods
     forecast = bvar_results.forecast(var_data.values[-2:], steps=5)
     forecast_df = pd.DataFrame(forecast, columns=['Forecasted_Stock_Returns', 'Forecasted_SP500_Returns'])
     print("Forecasted stock returns:", forecast_df)

     # Add forecast results back to the dataset for analysis
     forecast_dates = pd.date_range(start=merged_data['Date'].iloc[-1] + pd.Timedelta(days=1), periods=5)
     forecast_df['Date'] = forecast_dates
```

```
Forecasted stock returns:    Forecasted_Stock_Returns  Forecasted_SP500_Returns
0                  0.000162                 -0.000115
1                  0.001222                  0.000800
2                  0.000873                  0.000432
3                  0.000963                  0.000529
4                  0.000931                  0.000495
```

Dynamic Factor Models (DFM)

- Purpose: The DFM captures latent factors driving the movements in stock and S&P 500 returns, revealing underlying market sentiment.
- Modeling Approach: A Dynamic Factor Model with one latent factor was applied to stock returns and S&P 500 returns, extracting a common trend that explains their co-movement.
- Key Results: The model output includes a filtered factor representing market sentiment, which was added to the dataset for further analysis. The factor loadings indicated a strong relationship between stock returns and this latent factor.
- Implications: The extracted Market_Sentiment factor can be used as a timing signal, guiding investors on when to increase or decrease exposure to the market. This helps distinguish between market-driven movements and stock-specific performance

```
# Dynamic Factor Models (DFM)

from statsmodels.tsa.statespace.dynamic_factor import DynamicFactor

# Fit a Dynamic Factor Model with stock returns as endogenous variables
dfm_model = DynamicFactor(merged_data[['Stock_Returns', 'SP500_Returns']], k_factors=1, factor_order=1)
dfm_results = dfm_model.fit()
print(dfm_results.summary())

# Extract the common factor and add it to the dataset as a proxy for market sentiment
merged_data['Market_Sentiment'] = dfm_results.factors.filtered[0]

# Plot the extracted market sentiment factor over time
plt.figure(figsize=(10, 6))
plt.plot(merged_data['Date'], merged_data['Market_Sentiment'], label='Market Sentiment Factor', color='green')
plt.title('Extracted Market Sentiment from Dynamic Factor Model')
plt.xlabel('Date')
plt.ylabel('Market Sentiment')
plt.grid(True)
plt.legend()
plt.tight_layout()
plt.show()
```

```
                                   Statespace Model Results
==========================================================================================
Dep. Variable:     ['Stock_Returns', 'SP500_Returns']   No. Observations:            34232
Model:              DynamicFactor(factors=1, order=1)   Log Likelihood          197665.330
Date:                                Thu, 24 Oct 2024   AIC                    -395320.661
Time:                                        03:51:44   BIC                    -395278.456
Sample:                                             0   HQIC                   -395307.204
                                              - 34232
Covariance Type:                                  opg
===================================================================================
Ljung-Box (L1) (Q):            0.01, 21.26   Jarque-Bera (JB):   222433.63, 124108.35
Prob(Q):                        0.93, 0.00   Prob(JB):                     0.00, 0.00
Heteroskedasticity (H):         1.36, 1.16   Skew:                        0.31, -0.41
Prob(H) (two-sided):            0.00, 0.00   Kurtosis:                   15.47, 12.29
                        Results for equation Stock_Returns
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
loading.f1    -0.0124      0.000    -48.739      0.000      -0.013      -0.012
                        Results for equation SP500_Returns
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
loading.f1    -0.0108      0.000    -46.778      0.000      -0.011      -0.010
                         Results for factor equation f1
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
L1.f1         -0.0918      0.005    -19.929      0.000      -0.101      -0.083
                             Error covariance matrix
=====================================================================================
                         coef    std err          z      P>|z|      [0.025      0.975]
-------------------------------------------------------------------------------------
sigma2.Stock_Returns     0.0003   6.48e-06     43.981      0.000       0.000       0.000
sigma2.SP500_Returns  2.192e-09   4.92e-06      0.000      1.000   -9.65e-06    9.65e-06
=====================================================================================
```

.

Weekly Report 26 (10/22/2024 - 10/28/2024)

Value-at-Risk (VaR)

- Purpose: VaR quantifies the potential maximum loss in stock returns over a specified period, given a 95% confidence level, thus aiding in risk assessment.
- Calculation: The historical VaR method was applied, calculating the 5th percentile of stock returns, which indicates the worst expected loss under normal market conditions.
- Key Results: For stocks like AAPL, AMZN, and TSLA, the calculated VaR was around -0.0297, implying a 5% probability that losses could exceed this value during adverse conditions.
- Implications: VaR estimates provide a benchmark for potential downside risk, helping investors set stop-loss limits and maintain adequate capital buffers to manage their exposure to extreme market movements.

```
# Value-at-Risk (VaR) for Risk Management

# Store VaR results for each stock
var_results_dict = {}
confidence_level = 0.95

for stock in stocks:
    # Calculate VaR using historical method
    historical_var = np.percentile(merged_data['Stock_Returns'], (1 - confidence_level) * 100)
    merged_data['VaR'] = abs(historical_var)
    var_results_dict[stock] = historical_var
```

```
[103] var_results_dict
```

```
{'AAPL': -0.029774476771713364,
 'AMZN': -0.029774476771713364,
 'META': -0.029774476771713364,
 'GE': -0.029774476771713364,
 'GOOGL': -0.029774476771713364,
 'GS': -0.029774476771713364,
 'IBM': -0.029774476771713364,
 'JPM': -0.029774476771713364,
 'MSFT': -0.029774476771713364,
 'TSLA': -0.029774476771713364}
```

Markov Switching Models

- Purpose: Markov Switching models capture regime changes in stock returns, distinguishing between periods of high and low volatility, such as bull and bear market phases.
- Modeling Approach: A two-regime Markov Switching model was applied to stock returns data, identifying periods characterized by either high or low volatility.
- Key Results: The analysis identified 26,173 instances of low-volatility regimes and 8,059 instances of high-volatility regimes. This provides insights into when the market is more stable versus when it is subject to higher risk.
- Implications: Understanding regime shifts helps investors adjust their trading strategies, increasing their risk tolerance during stable periods and adopting more conservative strategies during high-volatility phases, thus optimizing returns and managing risks effectively.

```
# Markov Switching Models

from statsmodels.tsa.regime_switching.markov_regression import MarkovRegression

# Store Markov results for each stock
markov_results_dict = {}

for stock in stocks:
    # Fit a Markov Switching model with 2 regimes
    markov_model = MarkovRegression(merged_data['Stock_Returns'], k_regimes=2, trend='c', switching_variance=True)
    markov_results = markov_model.fit()
    markov_results_dict[stock] = markov_results

    # Add regime information to the dataset
    merged_data['Regime'] = np.where(markov_results.smoothed_marginal_probabilities[1] > 0.5,
                                     'High Volatility', 'Low Volatility')
```

```
[77] merged_data['Regime']
```

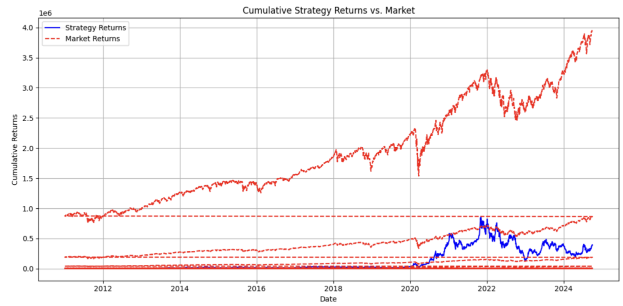
	count
Regime	
Low Volatility	26173
High Volatility	8059

dtype: int64

Weekly Report 26 (10/22/2024 - 10/28/2024)

Analysis of the Strategy vs. Market Performance Graph

```python
def trading_strategy_with_capm(data, capm_df, ecm_signals, garch_vol_dict, var_results_dict):
    """
    A trading strategy that uses CAPM alpha and beta, ECM signals, and GARCH volatility.
    """
    positions = []
    position_sizes = []

    for index, row in data.iterrows():
        stock = row['Stock']
        date = row['Date']
        alpha = capm_df.loc[capm_df['Stock'] == stock, 'Alpha'].values[0]
        beta = capm_df.loc[capm_df['Stock'] == stock, 'Beta'].values[0]
        ecm_signal = ecm_signals.get(date, 'hold')
        garch_volatility = garch_vol_dict.get(stock, pd.Series()).get(date, 0.02)
        market_var = var_results_dict.get(stock, 0.02)

        # Determine the position based on CAPM alpha and ECM signals
        if ecm_signal == 'buy' and alpha > 0:
            position = 1  # Buy if ECM indicates a buy and alpha is positive
        elif ecm_signal == 'sell' or alpha < 0:
            position = -1  # Sell if ECM indicates a sell or alpha is negative
        else:
            position = 0  # Hold if no clear signal

        # Adjust position size based on beta and GARCH volatility
        if beta > 0:
            position_size = 1000 / (1 + garch_volatility) * alpha * (1 - market_var)
        else:
            position_size = 500 / (1 + garch_volatility)

        positions.append(position)
        position_sizes.append(position_size)

    data['Position'] = positions
    data['Position_Size'] = position_sizes
    return data

# Apply the strategy to each stock
for ticker in tickers:
    stock_data = merged_data[merged_data['Stock'] == ticker].copy()

    # Apply the strategy with CAPM, ECM signals, and GARCH volatility
    stock_data = trading_strategy_with_capm(
        merged_data,
    capm_results,
    merged_data['GARCH_Volatility'],
    merged_data['Market_Sentiment'],
    merged_data['ECM_Signal'])

    # Store the results back into the merged_data
    merged_data.loc[stock_data.index, ['Position', 'Position_Size']] = stock_data[['Position', 'Position_Size']]
```




```
Strategy Sharpe Ratio: 0.5004722772294854
Strategy Maximum Drawdown: 724902.5661275974
```

1. Strategy and Market Comparison:
    - The graph shows the cumulative returns of the implemented trading strategy (blue line) versus the S&P 500 market returns (red line).
    - The S&P 500 (red line) displays a consistent upward trend, which is typical for a long-term market index, despite the noticeable drops around periods like early 2020 (likely due to the COVID-19 pandemic).
    - The strategy's performance (blue line) seems to have periods where it closely tracks the market but with lower absolute return values. This is evident especially in the more recent years, where the blue line shows increased volatility compared to the market.
2. Strategy Sharpe Ratio:
    - The reported Sharpe Ratio of approximately 0.50 indicates that for each unit of risk taken, the strategy delivers a return that is half a unit above the risk-free rate.
    - A Sharpe Ratio of 0.50 is moderate and suggests that while the strategy is not overly risky, it doesn't outperform the market in terms of return-to-risk ratio. Higher Sharpe Ratios (e.g., above 1) are typically desired for a strategy that provides a good balance of risk and reward.
3. Strategy Maximum Drawdown:
    - The strategy's maximum drawdown is noted as around 724,902. Drawdown measures the peak-to-trough decline during a specific period before a new peak is achieved.
    - This value suggests a substantial decrease in portfolio value from its peak, highlighting periods when the strategy faced significant losses before recovering.
    - It suggests that during market downturns or periods of high volatility, the strategy may not have been able to adjust effectively, leading to larger losses before stabilization.

Weekly Report 26 (10/22/2024 - 10/28/2024)

4. Market Outperformance:
    - The market's upward trend (red line) indicates that a buy-and-hold strategy in the S&P 500 would have yielded higher cumulative returns compared to the trading strategy.
    - The lower cumulative returns of the strategy suggest that while it may be able to manage risk and adjust positions based on various factors (such as ECM signals, GARCH volatility, CAPM beta, etc.), it may lack the same growth potential as the market during bullish phases.
5. Risk Management and Volatility:
    - The blue line's movements appear more erratic compared to the steady rise of the market, reflecting adjustments in positions based on GARCH volatility and ECM signals. This suggests that the strategy is more responsive to short-term market fluctuations.
    - This responsiveness can be beneficial during market downturns as the strategy might reduce exposure, but it may also lead to underperformance in steady bull markets due to a more conservative approach.

In summary, the trading strategy appears to offer better risk management and adaptiveness to market conditions through mechanisms like ECM signals and GARCH volatility. However, it sacrifices some of the market's long-term gains, as evidenced by the lower cumulative returns and Sharpe Ratio. The strategy could benefit from further optimization, such as better calibration of position sizes or improved timing of entry and exit signals to enhance overall returns.