Sector Allocation via the Black-Litterman Model Using a Sentiment Signal

Abhi Yadagiri

Supervised by Ravi Koka

August 2024

Abstract

This project applies the Black-Litterman portfolio allocation model to dynamically allocate capital across 11 sector-level ETFs using both historical market data and sentiment-derived investor views. By incorporating a Bayesian framework, the model blends equilibrium market returns with forward-looking views calibrated by confidence levels using the StockSnips news sentiment signal. The strategy was implemented using a rolling 900-day "training" window of historical data, with daily portfolio rebalancing performed over a backtesting period from January 3, 2023 to June 28, 2024. The results show that the Black-Litterman portfolio outperformed the S&P 500 benchmark over this period in terms of annual return and the Sharpe ratio, while also maintaining lower volatility. The use of sentiment signals to inform the Black-Litterman investor views and confidence levels demonstrates the potential of this allocation strategy to adapt to changing market conditions and generate robust, risk-adjusted returns.

1 Introduction

This project revolved around optimizing the allocation of funds to 11 sector-level ETFs using the **Black-Litterman model**. For each ETF, we had the following data from January 25, 2017 to June 28, 2024.

- Adjusted closing price on a given day
- Next day ticker prediction as provided by StockSnips using its sentiment model
- Next day ticker prediction confidence level as provided by StockSnips
- Daily return on a given day
- Total market cap on a given day
- Historical risk-free rates taken from 3-month T-bill data

The **S&P 500 closing prices and daily returns were used as the market benchmark** against each ETF and the Black-Litterman allocation portfolio.

2 Objectives

2.1 Model for Allocation

The objective of this project was to maximize the return of the portfolio containing the 11 ETFs over a "test period," chosen to be from January 3, 2023 to June 28, 2024, via the **Black-Litterman model**, using historical data from a rolling "training period" of 900 trading days. This meant that the model would be backtested using 900 trading days prior to each day from January 3, 2023 to June 28, 2024. Views and confidences were derived from the sentiment prediction data provided by StockSnips and then incorporated into the Black-Litterman model over the backtesting period to generate the optimal portfolio allocations.

2.2 Background of Black-Litterman Model

The Black-Litterman model is an analytical tool used by portfolio managers to optimize asset allocation within an investor's risk tolerance and market views. Developed by Fischer Black and Robert Litterman in 1990 at Goldman Sachs, the algorithm takes a **Bayesian** approach to asset allocation. This means that the model optimizes the allocation weights by combining a prior estimate of the returns (derived from the "training period" in our case) and incorporating the unique expectations of the investor for future returns. The model essentially computes a weighted average of the prior estimate of returns and particular views (derived from sentiment data in our case) held by the investor for each asset.

The weighting is dictated by the confidence levels (derived from sentiment data in our case) given by the investor for each of their views, which allows for a more personalized investment strategy. The model uses ideas from the **Capital Asset Pricing Model (CAPM)** and builds on the **Markovitz mean-variance optimization model**, in which the return is maximized for a given level of risk. Mean-variance optimization is considered limited since it only incorporates historical market data and then assumes those same returns going forward. This results in problems such as highly concentrated portfolios, input sensitivity, and maximization of estimation errors.

The Black-Litterman model uses "equilibrium" returns as a neutral starting point, where these "equilibrium" returns are the set of returns that clear the market, i.e., the market equilibrium. The aggregate of all portfolios in the market is assumed to be optimal, which reduces overreliance on historical data. The equilibrium returns are derived using a **reverse optimization** method in which the vector of implied excess equilibrium returns is extracted from known information. These can also be thought of as the implied market returns derived from the CAPM. If these implied returns appeal to investors, they can use the neutral weights given by the Black-Litterman model to develop their optimal portfolio. Otherwise, they should apply the Black-Litterman model, which adjusts the neutral weights according to the investor's views.

There are two key assumptions behind the Black-Litterman model:

1. All asset returns follow the **same probability distribution**.
2. The variance of the prior and the conditional distribution about the true means of the assets and the views of the investors are **unknown**.

3 Methodology

3.1 Theoretical Approach

Determining the prior estimate of returns involves relying on the market expectations, which are reflected in the market cap of the asset. First, we estimate the level of risk aversion among market participants, represented by the parameter δ. We compute δ by dividing the excess market return by the market variance, where the excess market return is the risk-free rate of return subtracted from the expected market portfolio rate of return:

$$\delta = \frac{E(R_m) - R_f}{\sigma^2}$$

Here, $E(R_m)$ is the expected market portfolio rate of return, R_f is the risk-free market rate of return, and σ^2 is the variance of the market portfolio. Now, we can calculate the Nx1 vector (where N is the number of assets) of **prior expected returns**, represented by Π. We use the risk aversion parameter δ, the NxN covariance matrix of asset returns represented by Σ, and the market cap weights of all the assets represented by the Nx1 vector w_{mkt} (calculated by dividing the market cap of each asset by the sum of the market caps of all assets) in the following formula:

$$\Pi = \delta \Sigma w_{mkt}$$

Next, we add the views of the investor. The Black-Litterman model allows users to provide either **absolute** or **relative** views. Absolute views are statements such as "Asset X will return 10%" or "Asset Y will drop 40%", while relative views are statements such as "Asset X will outperform asset Y by 3%". In our case, we will only consider absolute views. These views are specified in a vector represented by Q. These views are then mapped to the assets via a picking matrix represented by P. Each view has a corresponding row in the picking matrix (in which order matters), and the (absolute) views have a single 1 in the column corresponding to the ticker's order in the universe.

We also must have a KxK uncertainty matrix of views, which is represented by Ω. This is a diagonal covariance matrix that contains variances for each of the asset views. We can also specify view certainties as percentage confidences, which we will do in our approach for convenience, since we have access to this data.

Next, we can finally calculate the Nx1 vector of **expected posterior returns**, $E(R)$, estimated by the Black-Litterman model using the following formula:

$$E(R) = \left[(\tau\Sigma)^{-1} + P^T\Omega^{-1}P\right]^{-1}\left[(\tau\Sigma)^{-1}\Pi + P^T\Omega^{-1}Q\right]$$

Here, τ is a scalar number indicating the uncertainty of the CAPM distribution, which is usually within the interval $[0.025, 0.05]$. The above computation can be thought of as the weighted average between the prior estimate of returns and the investor's views, since the Black-Litterman model expresses the investor's views and the market equilibrium in terms of probability distributions. The weighting is determined by the confidences in the respective views and the parameter τ.

3.2 Software

To implement the Black-Litterman model, we used `PyPortfolioOpt` – a Python library that implements various portfolio optimization techniques, including Black-Litterman allocation. The `black_litterman` module within this library houses the `BlackLittermanModel` class, which generates posterior estimates of expected returns given a prior estimate and user-supplied views. In addition, two utility functions are defined:

1. `market_implied_risk_aversion`: This function calculates the risk aversion parameter δ and takes in the following as inputs:
    - Market prices (S&P 500 prices)
    - Frequency (252 trading days in a year)
    - Annualized risk-free rate (period must correspond to frequency)
2. `market_implied_prior_returns`: This function calculates the vector of prior expected returns Π and takes in the following as inputs:
    - Market caps
    - Risk aversion parameter δ
    - Covariance matrix of asset returns Σ
    - Annualized risk-free rate (period must correspond to frequency)

A `BlackLittermanModel` object requires a specific input format, specifying the prior, the views, the uncertainty in views, and a picking matrix to map the views to the asset universe. The input format we used implemented **Idzorek's method** [1], which allows us to specify our view certainties as percentage confidences, which we had as part of our data. This method essentially incorporates the default estimate of Ω, and due to this, the value of the parameter τ does not matter, since τ cancels out in the matrix multiplications.

The picking matrix P was not used; the `BlackLittermanModel` function only requires us to input the covariance matrix, prior returns, absolute views, and percentage confidences into the `view_confidences` parameter, while specifying the `omega` parameter as `"idzorek"`, i.e. `omega="idzorek"`. This function returns the expected posterior returns of each asset on a given day.

Functions included in the `EfficientFrontier` module within `PyPortfolioOpt` were then used to "clean" the "raw" weights, i.e. making sure the weights were nonnegative and negligible weights were minimized. Additionally, a constraint was placed in which the weight of an asset on a given day was at least 20% of the market weight of that asset on the previous day. The weights were then optimized to maximize the Sharpe ratio of our portfolio on the given day.

3.3 Implementation

Combining the theoretical approach of the Black-Litterman model with the features of the PyPortfolioOpt library, Python was used to apply the model to the data. After loading all historical data, including historical daily returns, S&P 500 prices, absolute views, percentage confidences, and risk-free rates, a loop was created with the dates from January 3, 2023 to June 28, 2024, where the "test start date" was set to January 3, 2023. We set our "training period" parameter at 900 days, where we would look back 900 days to the day before the desired date in each iteration of the loop (for estimation of the various inputs to the Black-Litterman model). This period of 900 days was also used to generate the sentiment signal to predict the next day return for each ETF in the test period. The absolute views and percentage confidences for the desired date were used in each iteration of the loop to calculate the expected posterior returns and posterior weights given by the Black-Litterman model for that day. For each day from January 3, 2023 to June 28, 2024, the output of our code was the following for each ETF:

- Expected posterior return
- Posterior weight according to Black-Litterman allocation
- Market (S&P 500) weight
- Closing price
- Posterior return view
- View confidence percentage
- Actual daily return

The output also included the following daily computations for our portfolio containing all of the assets:

- Daily risk-free rate
- Daily portfolio return (based on optimal Black-Litterman allocation weights)
- Daily market (S&P 500) return
- Daily excess return (portfolio return minus risk-free rate)
- Sharpe ratio
- Volatility

The following statistics were calculated for the portfolio over the test period:

- Mean daily Black-Litterman allocation weight for each ETF

- Mean daily market weight for each ETF
- Cumulative portfolio return
- Cumulative market (S&P 500) return
- Sharpe ratio
- Volatility
- Alpha
- Beta

4 Results

We will now discuss the results of dynamically applying the Black-Litterman model in conjunction with sentiment-driven investor views to generate daily optimal ETF allocations over the test period from January 3, 2023 to June 28, 2024. Figure 1 illustrates the differences between the portfolio's mean allocation weights and the corresponding market capitalization weights across all ETFs.

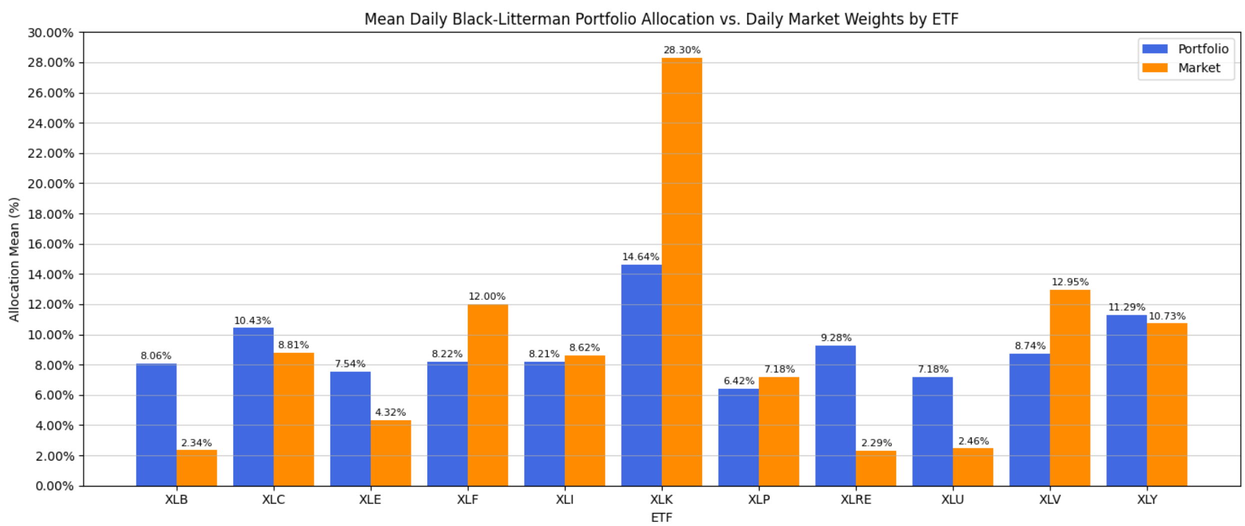

Figure 1: Comparison of Mean Portfolio and Market Allocation Weights by ETF

We see that XLK shows the highest allocation (14.64% on average over the test period), followed by XLY (11.29%) and XLC (10.43%). The XLK allocation average is noticeably lower than the average market weight of 28.30% over the test period. This is likely due to the higher beta for this ETF and the higher expected posterior return used for predicted "negative" return days. This might be resulting in a bias that could be mitigated if we were to use different magnitudes, rather than identical, for each asset's "up" and "down" posterior view predictions on a given day. In Figure 2 below, we can observe the variation in the average allocation by month for each ETF.

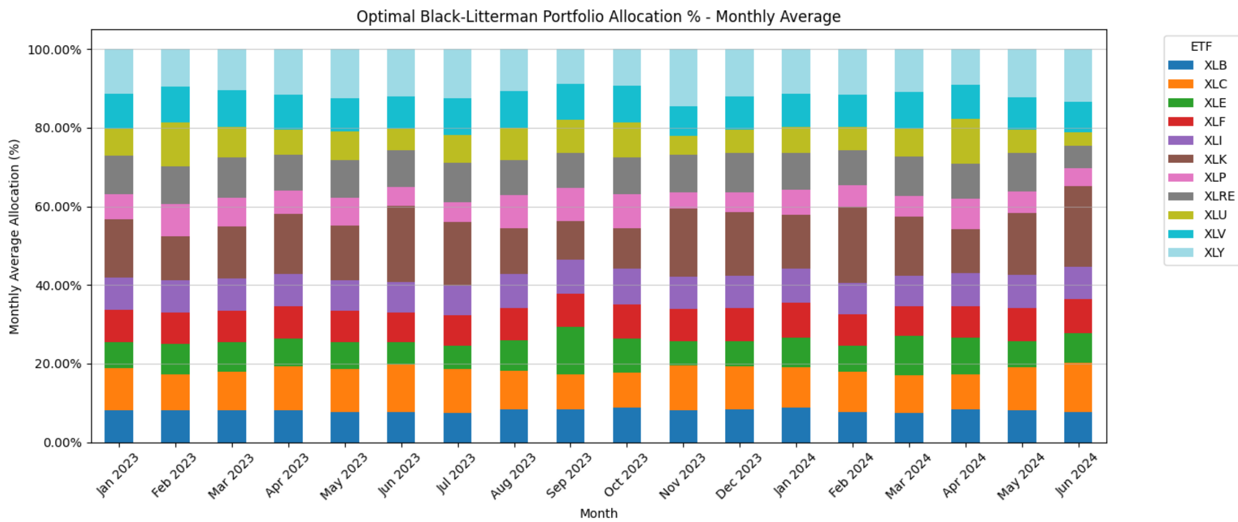

Figure 2: Average ETF Allocation Weight by Month

Figure 3 below depicts the standard deviation of the daily allocation weights by ETF.

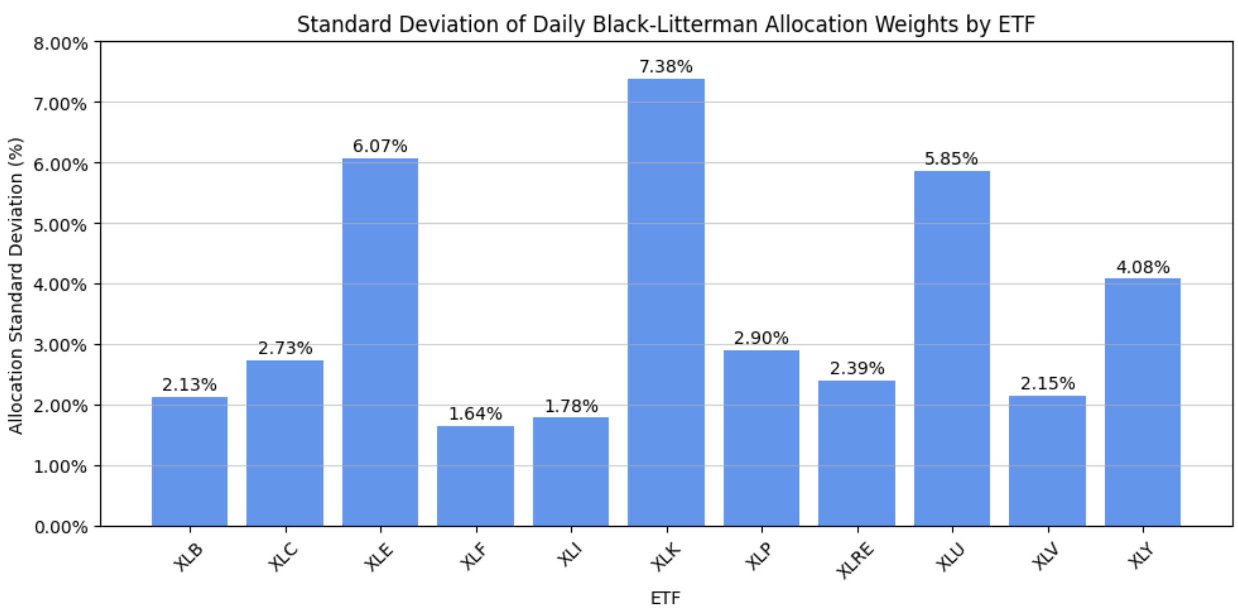

Figure 3: Standard Deviation of Portfolio Allocation Weights by ETF

The bar graph above indicates higher allocation volatility for the XLE (Energy), XLK (Technology), XLU (Utilities), and XLY (Consumer Discretionary) sectors. The model is clearly changing the allocations for these ETFs based on the sentiment prediction signal, which demonstrates the importance of these sectors in achieving optimal portfolio allocation.

The daily ETF allocation weights were then used to calculate the portfolio's cumulative return. Figure 4 below compares the performance of the portfolio to that of the S&P 500 over the test period.

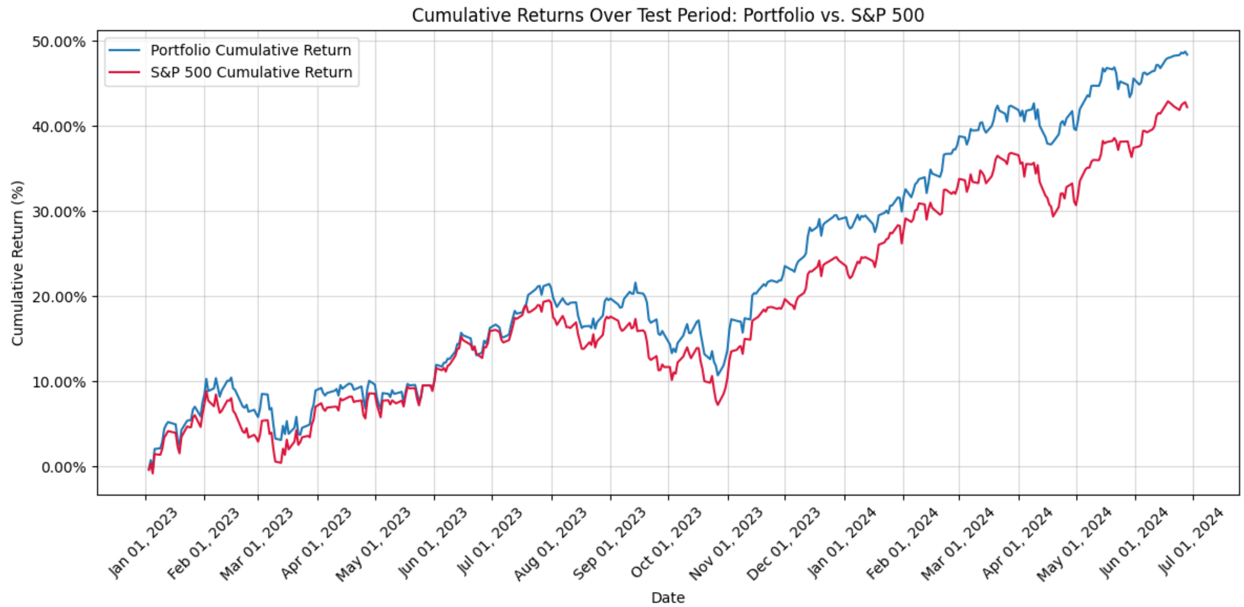

Figure 4: Comparison of Portfolio and Market Cumulative Returns Over Test Period

The graph above shows that over the test period, the total return for the portfolio was **48.39%**, while that for the S&P 500 was **42.22%**, indicating that **the portfolio outperformed the S&P 500 by 6.17%** over that time period. Moreover, we observe slightly **lower volatility in the portfolio** compared to the benchmark, especially noticeable during April 2024 when the portfolio drawdown was much lower than that of the S&P 500. Overall, Figure 4 indicates that for the test period, the Black-Litterman allocation portfolio yielded better results than the market benchmark.

The following performance and risk statistics were computed for the portfolio over the test period:

- **Beta: 0.924** – which indicates that the portfolio is slightly less volatile than the S&P 500 ($0.924 < 1$).
- **Alpha: 4.81%** – which indicates that the portfolio outperformed the S&P 500 by 4.81% annually on a risk-adjusted basis.
- **Sharpe ratio: 1.84** – which indicates a strong risk-adjusted performance.

These summary statistics show that using sentiment-driven posterior views in conjunction with the Black-Litterman model yielded robust returns relative to the market, both in absolute terms and on a risk-adjusted basis, over the backtesting period.

5 Conclusion

This project demonstrated the effectiveness of using the Black-Litterman model with predicted investor views generated via the StockSnips sentiment signal to construct a robust and adaptive ETF portfolio. In particular, the lower beta of 0.924 and high Sharpe ratio of 1.84 demonstrate the strong risk-adjusted performance potential of this allocation strategy, which indeed generated a positive alpha of 4.81% over our chosen backtesting period. Our portfolio delivered nearly 5% more return annually than the CAPM would predict given its level of market risk, as determined by its beta. Furthermore, rebalancing sector allocations daily appear to result in lower volatility during high market drawdown periods. Our strategy took on less systematic risk, yet outperformed the market benchmark, which suggests the Black-Litterman model succeeded in finding superior risk-adjusted positions during the backtesting period.

However, the use of the same posterior return magnitude for both the "up" and "down" signals is likely biasing the model against ETFs with a higher overall average return (most notably XLK being significantly underweighted) – which needs to be further investigated.

Overall, this project showed that using a sentiment signal as input to the Black-Litterman model has the potential for positive alpha generation and yielding robust returns relative to the market.

6 References

[1] Idzorek, T. "A Step-By-Step Guide to the Black-Litterman Model." `https://people.duke.edu/~charvey/Teaching/BA453_2006/Idzorek_onBL.pdf`

[2] Berkeley Black-Litterman Notes. `https://www.stat.berkeley.edu/~nolan/vigre/reports/Black-Litterman.pdf`

[3] Hudson & Thames: Bayesian Portfolio Optimization. `https://hudsonthames.org/bayesian-portfolio-optimisation-the-black-litterman-model/`

[4] PyPortfolioOpt Documentation. `https://pyportfolioopt.readthedocs.io/en/latest/BlackLitterman.html`