Black-Litterman Models in Portfolio Optimization

Rui Zong

1. Abstract

In this report, we'd like to explore how to build a portfolio with Black-Litterman model combined with sentiment analysis, how well it performs, and differences between Mean-Variance model and BL model optimization outcomes. First, we use 2 different methods to select assets from 2019-01-01 to 2021-09-30. Then, calculate 1-month assets' sentiment scores from a predeterminate financial website. Next, build two models, test each model's outcome in 1 million dollars with those assets' scores. Finally, back test the portfolio and show its risk metrics and performing result from 2021-10-01 to 2022-10-01.

2.Data Collection and Asset Selection

Asset Selection (Method I)

We collect monthly prices of 101 companies together with S&P500 index from 2019-01-01 to 2021-09-30. Dropping all NA values, we calculate monthly returns from percentage change.

Date	AAL	AAPL	ABBV	ABT	AIV	AMAT	AMD	AMZN	APA	ATVI	...
2019-02-01	-0.003914	0.040315	-0.000973	0.068511	-0.011914	-0.018935	-0.036051	-0.045906	0.019008	-0.107959	...
2019-03-01	-0.106186	0.101731	0.017037	0.029889	0.036015	0.039720	0.084573	0.085936	0.044605	0.080446	...
2019-04-01	0.076196	0.056436	-0.014890	-0.004754	-0.018493	0.111195	0.082680	0.081859	-0.050491	0.067368	...
2019-05-01	-0.203335	-0.127572	-0.020946	-0.039196	0.011953	-0.122078	-0.007962	-0.078613	-0.202427	-0.100394	...
2019-06-01	0.201032	0.134873	-0.052014	0.104689	0.011280	0.166702	0.107990	0.066792	0.111239	0.088310	...

5 rows × 102 columns

Figure 1: Stock returns from Yahoo Finance

To select 10 valuable stocks as our portfolio asset, we use two methods to do the selection separately and then chose the intersection. First, we select stocks by comparing its returns with returns of SPY. Based on the data frame of returns of all stocks and SPY, we label the return as 1 if it is larger than the return of SPY in the same month and label it as 0 otherwise. This is done on all stocks with all monthly returns so that we got a data frame of binary data. And then, we count the occurrence of 1 for each stock, which means the occurrence that the return of stocks being larger than that of SPY. Sorting the stocks, we choose the top 20 stocks with the largest occurrence beating SPY as our portfolio asset.

	NVDA	AAPL	AMAT	PYPL	MSFT	SYF	MS	RF	KEY	V	...	BMY	MO	CNP	HPE	MDLZ	DIS	COP	T	WBA	VZ
1	23	22	22	21	21	20	19	19	19	19	...	11	11	11	11	10	10	10	10	9	9
0	9	10	10	11	11	12	13	13	13	13	...	21	21	21	21	22	22	22	22	23	23

2 rows × 101 columns

Figure 2: Occurrence of beating SPY for each stock

```
['NVDA' 'AAPL' 'AMAT' 'PYPL' 'MSFT' 'SYF' 'MS' 'RF' 'KEY' 'V' 'QCOM' 'APA'
 'HPQ' 'AMD' 'CMCSA' 'UAA' 'DVN' 'USB' 'NCLH' 'NKE']
```

Figure 3: Stock selection results

Asset Selection (Method II)

The second method involves calculating the cumulative returns and Sharpe ratio for the 101 companies mentioned previously. These 101 companies have the largest trade volume out of the 500 companies listed on the S&P 500. The result is a table and graph listed below. We select the top 20 companies with largest cumulative return. These companies generally have a Sharpe ratio > 1 too. Every dot on the scatter plot represents a single company.

COMPANY	CUMULATIVE_RETURNS	SHARPE_RATIO
APA	2.022815	0.743925
RRC	1.995572	0.173573
NVDA	1.976876	1.452393
AMD	1.665476	1.482620
FCX	1.452297	0.774905
AMAT	1.416472	1.255813
AAPL	1.399127	1.602450
QCOM	1.279928	2.117696
DVN	[illegible]	[illegible]

Table 1: The table above sorts the cumulative returns, from largest to smallest.

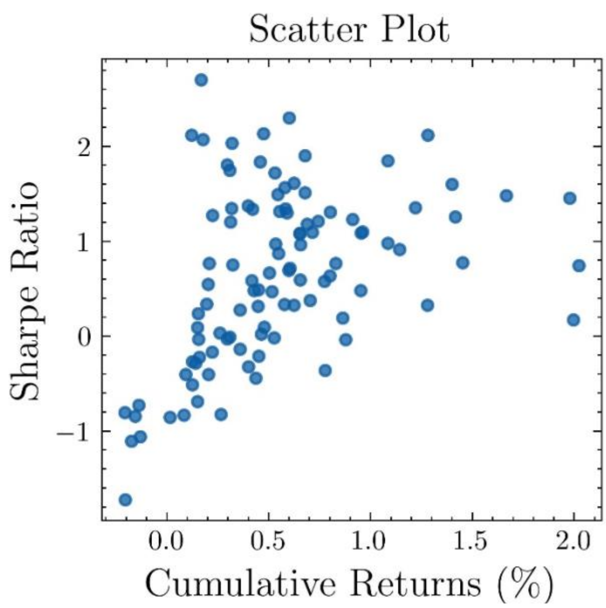


Figure 4: The values from the table are being plotted.

Finally, we compared the selection results from both methods and chose the 10 stocks shown in both as our portfolio asset, consisting of 'NVDA', 'AAPL', 'AMAT', 'PYPL', 'MSFT', 'SYF', 'MS', 'QCOM', 'APA', 'AMD' and 'DVN'.

What's more, before moving to sentiment analysis, we computed and tested the existence of correlation and clustering in our stocks. The 'heatmap' method in python helped to visually show the correlation among the 10 stocks selected. From the figure 4 shown below, we can tell that stocks were positively correlated with each other and some correlation coefficient were high, for example 'MSFT' and 'APPL'. Thereofore, we further used hierarchical clustering to see whther there existed clusters. Fortunately, from the dendrograms in figure 5, we can see that there was no obviously distant clusters and so we can use these stocks to do the analysis in the following.

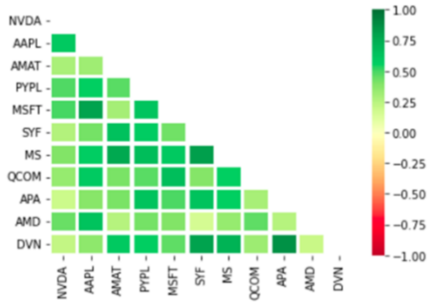


Figure 5: Correlation heatmap of stocks

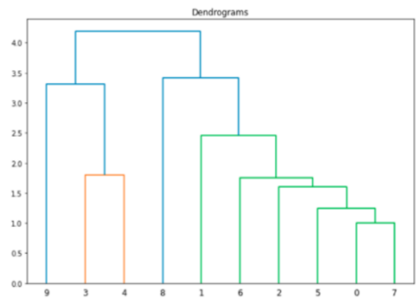


Figure 6: Dendrograms of Hierarchical clustering

3.Sentiment Analysis

In this chapter, we will use these assets we chose to do sentiment analysis. After we got the top10 cumulative returns, 'NVDA', 'AAPL', 'AMAT', 'PYPL', 'MSFT', 'MS', 'QCOM', 'APA', 'AMD' and 'DVN' (See Table 1), we need to do sentiment analysis for each stock. We decide to capture the relevant headline of news in website: FinViz as our source or input, figure 6 shows the information we will capture, and the figure 7 shows that through python code we capture the input in sentiment analysis.

NVDA	AAPL	AMAT	PYPL	MSFT
1.9769	**1.3391**	**1.4165**	**1.2200**	**1.0842**
MS	QCOM	APA	AMD	DVN
1.0851	**1.2800**	**2.0228**	**1.6655**	**1.2791**

Table 2: Top 10 stocks with highest cumulative returns

Dec-06-22 04:55PM	The Chip War Cant Be Solved by Taiwan Semis $40 Billion	Barrons.com
04:43PM	iPhone Pros Wait Times Are Improving. What That Means for Apple Stock.	Barrons.com
04:40PM	TSMC expects $10 billion in annual revenue from Arizona fabs	Reuters
04:32PM	TSMC expects $10 billion in annual revenue from Arizona fabs	Reuters
03:49PM	RPT-Apple is sued by women who say AirTag lets stalkers track victims	Reuters

Figure 7: Information in website: FinViz

```
[['AAPL',
  'Dec-06-22',
  '01:52PM',
  'Dec-06-22 01:52PMApple Scales Back Self-Driving Car and Delays Debut Until 2026 Bloomberg'],
 ['AAPL',
  'Dec-06-22',
  '01:39PM',
  '01:39PMThe Chip War Wont Be Solved by Taiwan Semi Spending $40 Billion in the U.S. Barrons.com'],
 ['AAPL',
  'Dec-06-22',
  '01:11PM',
  '01:11PMTech IPOs could make a comeback in 2023, analyst says Yahoo Finance'],
 ['AAPL',
  'Dec-06-22',
  '01:00PM',
  '01:00PMApple Announces Biggest Upgrade to App Store Pricing, Adding 700 New Price Points Business Wire'],
```

Figure 8: Information we capture from the website

The next step that we will use input to do sentiment analysis. For this step, we will python package named 'Vader' to do analysis, the result shows in figure 8. Because of the large numbers, we visualize these number as figure 9.

	ticker	date	compound
0	AAPL	2022-12-06	-0.4404
1	AAPL	2022-12-06	0.5563
2	AAPL	2022-12-06	-0.6921
3	AAPL	2022-12-06	0.4215
4	AAPL	2022-12-06	0.0000
...	...	...	...
98	AAPL	2022-12-02	-0.2732
99	AAPL	2022-12-02	0.2732
100	AAPL	2022-12-02	0.4019
101	AAPL	2022-12-02	-0.6808
102	AAPL	2022-12-02	-0.1531

	ticker	date	compound
103	NVDA	2022-12-08	0.1027
104	NVDA	2022-12-08	-0.7269
105	NVDA	2022-12-07	0.5423
106	NVDA	2022-12-07	-0.4404
107	NVDA	2022-12-07	-0.7430
...	...	...	...
201	NVDA	2022-11-17	-0.4404
202	NVDA	2022-11-17	0.0772
203	NVDA	2022-11-17	0.0000
204	NVDA	2022-11-17	-0.3182
205	NVDA	2022-11-17	-0.1531

	ticker	date	compound
927	AMD	2022-12-08	0.4939
928	AMD	2022-12-08	-0.4404
929	AMD	2022-12-08	-0.4404
930	AMD	2022-12-08	-0.7269
931	AMD	2022-12-07	-0.4404
...	...	...	...
1025	AMD	2022-11-14	0.3291
1026	AMD	2022-11-14	0.0000
1027	AMD	2022-11-14	0.0000
1028	AMD	2022-11-14	0.5994
1029	AMD	2022-11-14	0.0000

Figure 9: Sentiment Score in AAPL, NVDA and AMD

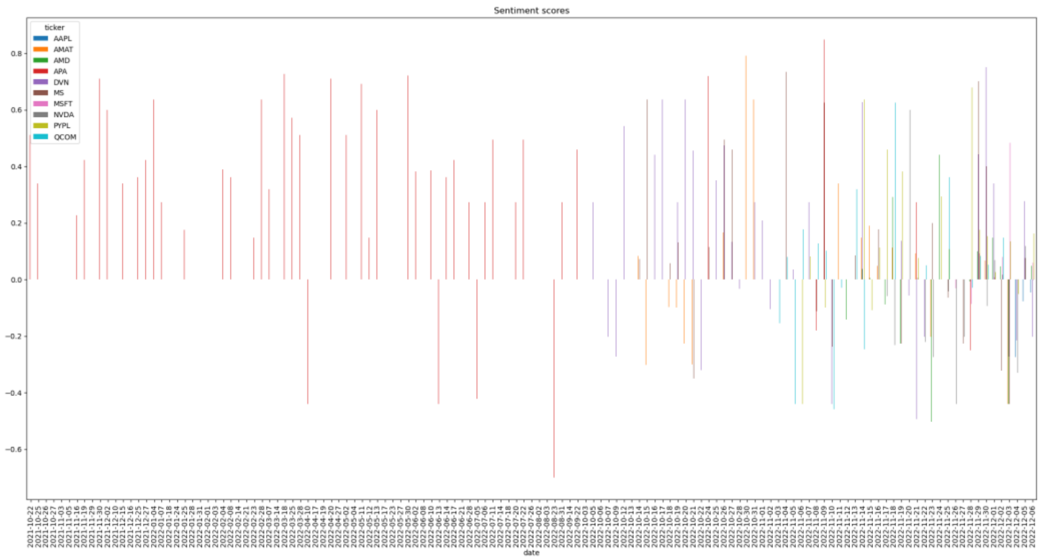


Figure 10: Sentiment Scores

From the figure 9 and numbers, the ‘compound’ column gives the sentiment scores. For positive scores, the higher the value, the more positive the sentiment is. Similarly for negative scores, the more negative the value, the more negative the sentiment is. The scores range from -1 to 1.

From now, we have the sentiment scores for each stock, however we want to verify that the scores are confident for our analysis. Based on that, we download the stock prices for each stock we need, and then calculate the log-return combining with our sentiment scores. Finally, we can compute the correlation between the log-returns and sentiment scores. Table 2 shows the results which will be used as the input of views in Black Litterman model.

NVDA	AAPL	AMAT	PYPL	MSFT
0.3301	**0.0112**	**-0.1490**	**-0.2367**	**0.2367**
MS	QCOM	APA	AMD	DVN
0.5801	**-0.1076**	**0.3047**	**0.0776**	**-0.2788**

Table 3: Correlation between log-return and sentiment scores

4.Black Litterman Model

The idea for this chapter is that we try to set the sentiment correlation scores we calculated before as investors' views for those assets. We use training set (2019-01-01 to 2021-09-30) to calculate 10 weights for those stocks with Original Mean-Variance Method and BL Model, then employ those weights for two portfolios' returns on the testing set (2021-10-01 to 2022-10-01).

First, let's look at those assets' returns in training set and testing set. Based on Training set period, we could figure out the cumulative returns for these assets performed well. However, some of these assets' prices plummet in the testing set.

Figure 11: Returns in Training Set & Testing Set

Then, we develop the Original Portfolio Optimization for ten assets, we could buy or sell these assets (-1,1) with boundary, have ten weights.

Figure 12: Original Mean-Variance Optimization

```
# The constraints
cons = (# The weights must sum up to one.
        {"type":"eq", "fun": lambda x: np.sum(x)-1},
        # This constraint says that the inequalities (ineq) must be non-negative.
        # The expected daily return of our portfolio and we want to be at greater than 0.001
        {"type": "ineq", "fun": lambda x: np.sum(returns_train.mean()*x)})
# the objective function is to minimize the portfolio risk
def objective(weights):
    weights = np.array(weights)
    return weights.dot(cov_matrix_train).dot(weights.T)
# Every stock can get any weight from -1 to 1
bounds = tuple((-1,1) for x in range(returns_train.shape[1]))
guess = [0.1 for x in range(returns_train.shape[1])]
optimized_results = minimize(objective, guess, method = "SLSQP", bounds=bounds, constraints=cons)
optimized_results
```

```
     fun: 0.08344179733195195
     jac: array([0.1671298 , 0.1669025 , 0.16690387, 0.1667699 , 0.1667854 ,
       0.16659689, 0.16645367, 0.16697125, 0.16707404, 0.16676096])
 message: 'Optimization terminated successfully'
    nfev: 187
     nit: 17
    njev: 17
  status: 0
 success: True
       x: array([ 0.18083168, -0.18910473, -0.12601285,  0.01060402, -0.01864123,
        0.22822299,  0.11002038,  0.6742593 ,  0.09753804,  0.0322824 ])
```

Next, we import our optimization model with BL Model and those sentiment scores.

Figure 13: Black-Litterman Optimization

```
viewdict = {'AAPL':0.0112,'NVDA':0.3301,'AMAT':-0.1490,'DVN':-0.2788,'APA':0.3047,'MS':0.5801,
            'QCOM':-0.1076,'MSFT':0.2366,'PYPL':-0.2367,'AMD':0.0776 }
bl = BlackLittermanModel(cov_matrix_train, absolute_views=viewdict)
a=bl.bl_weights()
df = pd.DataFrame({'Ticker':a.keys(), 'Weights':a.values()})
df
```

As shown, the Mean-Variance weights are in the left side, and the BL weights are in the right side. Totally, 7 assets' weights are changed because of their sentiment scores, including AAPL, NVDA, DVN, APA, QCOM, PYPL, AMD.

Figure 13: Mean-Variance Weights (left) & Black-Litterman Weights(right)

	Symbol	Weight
0	AAPL	0.180832
1	NVDA	-0.189105
2	AMAT	-0.126013
3	DVN	0.010604
4	APA	-0.018641
5	MS	0.228223
6	QCOM	0.110020
7	MSFT	0.674259
8	PYPL	0.097538
9	AMD	0.032282

	Ticker	Weights
0	AAPL	-0.381544
1	NVDA	0.825945
2	AMAT	-0.690233
3	DVN	-0.621157
4	APA	0.221615
5	MS	2.072790
6	QCOM	-0.546922
7	MSFT	1.395214
8	PYPL	-1.267650
9	AMD	-0.008058

Now, we back test two portfolio optimization in training set and testing set. In training set, we get a 0.14%, 8% Sharpe-Ratio.
Based on training set Mean-Variance Model, we get a negative mean return -0.05% mean return and -3.39%.
Fortunately, we have a1.36% mean return and 2.89% Sharpe-Ratio in our portfolio.

Figure 14: Training & Testing in Mean-Variance Model and BL Model

```
opt_result = np.matmul(np.matrix(returns_train), np.array(optimized_results.x))
train_mean=opt_result.mean()
train_sharpe=opt_result.mean()/opt_result.std()
```

```
train_mean
```

```
0.001497244548090308
```

```
train_sharpe
```

```
0.08234082408487556
```

```
opt_result_newret = np.matmul(np.matrix(returns_test),np.array(optimized_results.x))
unstrained_sharpe=opt_result_newret.mean()/opt_result_newret.std()
opt_result_newret.mean()
```

```
-0.0005506728972510742
```

```
unstrained_sharpe
```

```
-0.03399663432545509
```

```
opt_result_newret = np.matmul(np.matrix(returns_test),weights.reshape(-1, 1))
strained_sharpe=opt_result_newret.mean()/opt_result_newret.std()
opt_result_newret.mean()
```

```
0.0013631209931548134
```

```
strained_sharpe
```

```
0.02892785369029126
```

Finally, this is our portfolio returns versus SPY. It performs higher than SPY in most of the testing period.

Figure 15: Portfolio returns & SPY returns

5.Risk Analysis

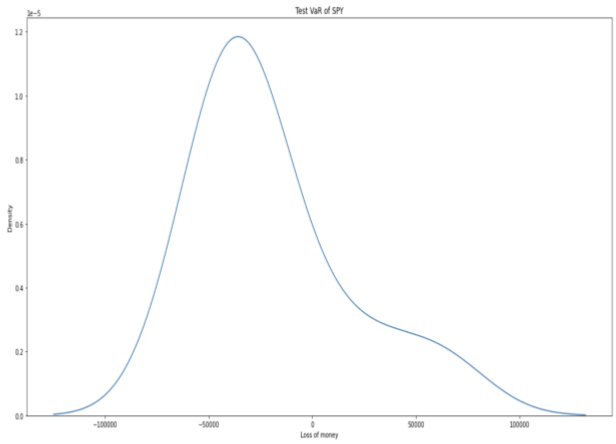
In this part, we develop VaR calculation for our portfolio, SPY, and 10 assets we selected. The main idea is to use historical method to visualize 5% VaR for investing 1 million dollars.

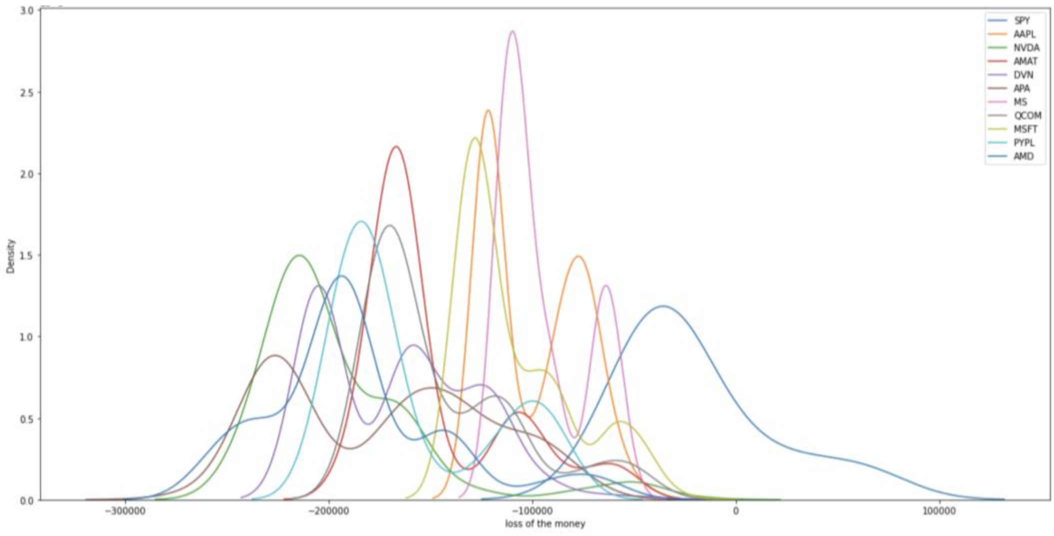
The VaR for SPY and for each asset show a large density of losing situation. And our portfolio's VaR plot seems like a normal distribution with mean value close to 0.

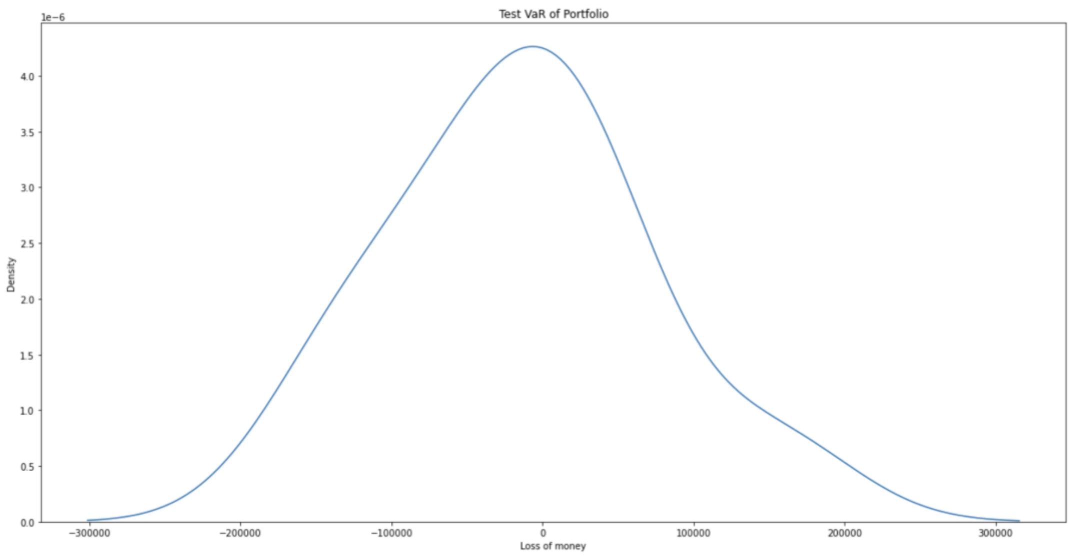
Figure 16: VaR for investing 1 million in SPY & Portfolio & each asset.

```
# 10day VaR = sqrt(10)* 1day VaR
roll = 10
a = 0.05
bf_day = 100
seq = int(a*bf_day)
len_day=len(data['spy'])-roll
var_1 = np.zeros(len_day)
var_1.shape
for i in range(len_day):
    td_var_1 = data['spy'].iloc[i:i+10].sort_values()
    var_1 = td_var_1
    var_10 = var_1*np.sqrt(10)

plt.figure(figsize=(20,10))
sns.kdeplot(var_10*1000000)
plt.title("Test VaR of SPY")
plt.xlabel("Loss of money")
```







6.Conclusions &Improvements:

We try sentiment correlation scores as investors' views for each asset on the training set and testing set.
This proves sentiment analysis do have positive influences on portfolio optimization.

Improvements if we have more time:
1. Try to use GANs (Generative Adversarial Networks) to simulate new test data (similar distribution with previous training data) for VaR calculation.
2. Combine with more Financial website Information, such as announcements from the Federal Reserve or Twitter. Change this calculate method for investment scores.
3. Employ this strategy with different kind of asset, such as futures.
4.Use different distribution such as Gram-Charlier distribution to measure the portfolio tail risk.

7.Reference

1.Algorithmic Trading and Quantitative Strategies, Raja Velu.

2.Machine Learning for Algorithmic Trading, Stefan Jansen

3.The Black-Litterman Approach: Original Model and Extensions, Attilio Meucci (2010).