REPORT

TASK 1

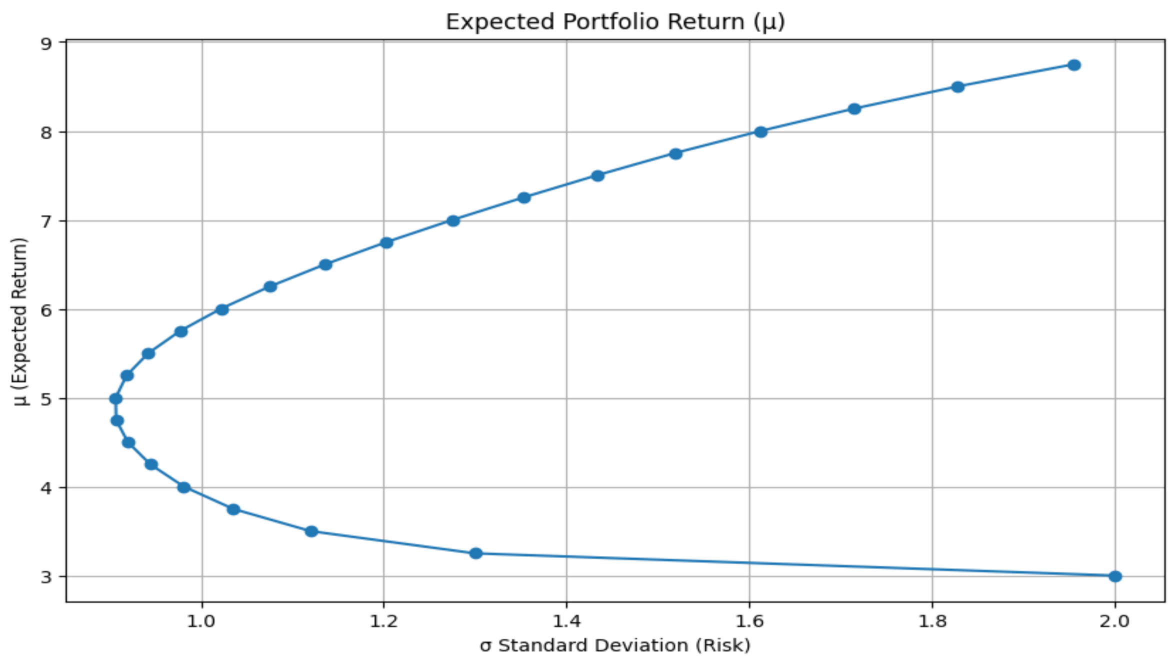
Data Generation:

The “generate_information()” function is designed to generate random data for an array of properties, in this case 10. The correlation matrix is initialized to zero and filled with a pattern based on the differences between the properties. This function produces the sig ma value (standard deviation) and mu value (expected return) for each asset. These val uesare used to generate the diagonal matrix, which is combined with the connected mat rix to calculate the C2 matrix. The final covariance matrix is
obtained by averaging the C2 matrix and its transpose.

Portfolio Optimization:

Return values (`r_values`) are defined as factors from 2.00 to 9.00 in 0.25 increments. This leads to a line that captures the risk (standard deviation) and expected return values for the improved portfolio. Constructed data for mean income (`mu_values`), covariance matrix (`C`), and number of attributes (`n`) are obtained by calling the `generate_information()` function. For each useful result in `r_values`, a new Gurobi optimization model is created. A variable is added that represents the weight, between 0 and 1. The objective function is set to minimize the variance of the portfolio. A bonus is added to ensure that the fund receives a return of 1 money. The model has been modified.If the optimization is successful, the optimal asset weights are found and the financial risk (standard deviation) and expected return are calculated. These products will be added to that list. If the return optimization fails, it prints an error message.

Plot Analysis :



The graph illustrates the Markowitz efficient frontier, which describes the relationship between risk (shown as standard deviation on the x-axis) and the portfolio's expected return (y-axis). This curve shows the optimal portfolios that provide the highest expected return for a given level of risk. Initially, the curve rises sharply, which means a considerable return from a relatively small increase in risk. As the curve moves to the right, the increase in return from additional risk begins to level off significantly. This shows that above a certain risk threshold, the return of higher risk reduces the return, a key concept of portfolio optimization. The efficient frontier is critical for investors because it helps them choose portfolios that meet their risk appetite and return goals, effectively showing the trade-off between risk and potential return.

TASK 2:

Data Generation:

Initially, the `generate_information()` function generates the required data, including the mean return (`mean_return`) and the covariance matrix (`covariance_matrix`) for the number of assets (`num_assets`). This function fills correlation and covariance matrices using log numbers, random values, and predefined patterns.

Portfolio Optimization :

This code sets the return rate (`r_vals`) from 2.00 to 9.00 in 0.25 increments. The array is initialized to store the risk (`risk_values_sigma`) and return values (`return_values_mu`) for the optimized portfolio.
For each useful result in `r_vals`, a new Gurobi optimization model is created. Variables representing asset weights (`x_vars`) are added with bounds between 0 and 1. The objective function is set to minimize portfolio variance. Added constraints to the portfolio so that the return (`return_constraint') and total constraints (`budget_constraint') are less than or equal to 1. The formula has been modified.
If the optimization is successful (checked using `model.status`), we get the optimal weights (`x_optimal`). The risk (standard deviation) of the portfolio is calculated using the covariance matrix and optimal weights. Expected returns are calculated using average returns and optimal weights. These products will be added to that list. Failure to optimize return rates can lead to errors. This code sets the return rate (`r_vals`) from 2.00 to 9.00 in 0.25 increments. The array is initialized to store the risk (`risk_values_sigma`) and return values (`return_values_mu`) for the optimized portfolio.
For each useful result in `r_vals`, a new Gurobi optimization model is created. Variables representing attribute weights (`x_vars`) are added with bounds between 0 and 1. The objective function is set to minimize portfolio variance. Constraints are added to the folder

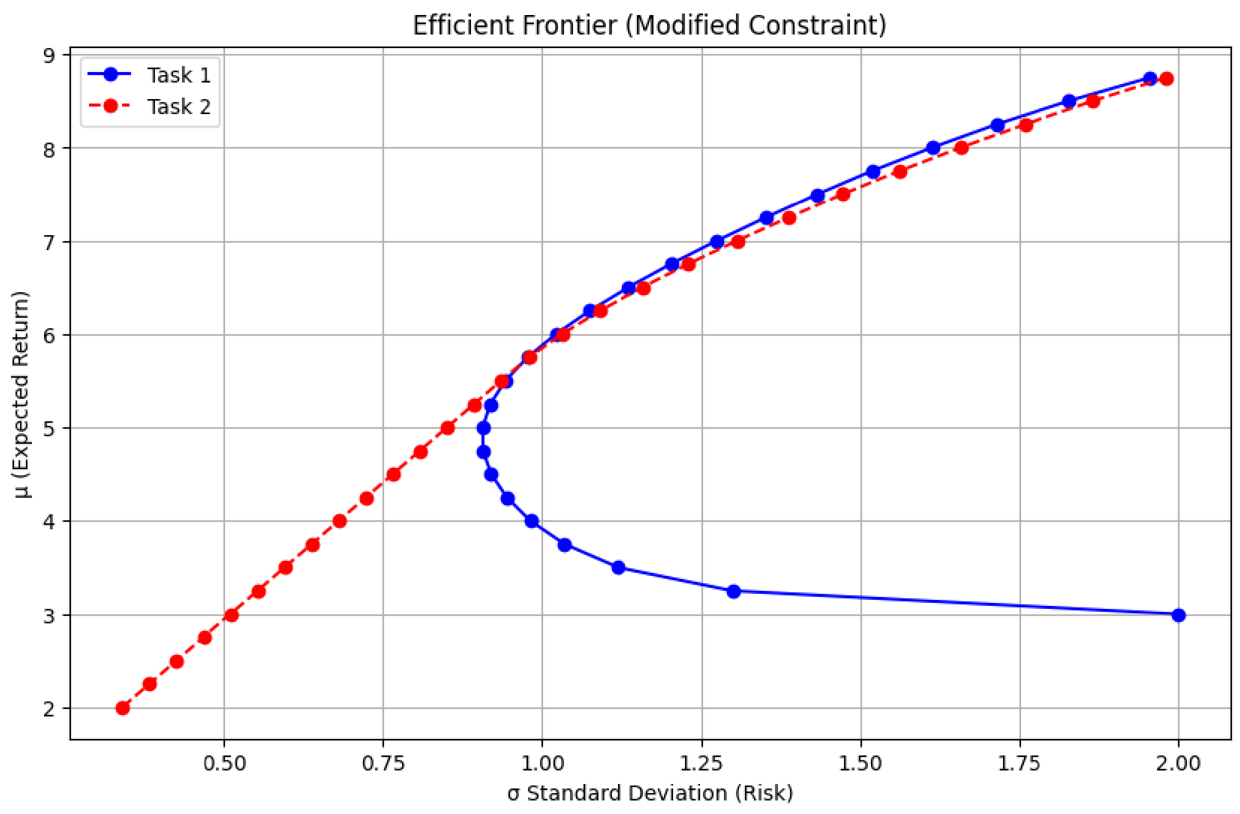
so that the return (`return_constraint') and the total constraint (`budget_constraint') are less than or equal to 1. The formula has been changed.

If the optimization is successful (marked in `model.status`), the optimal weights (`x_optimal`) are obtained. The risk (standard deviation) of the portfolio is calculated using the covariance matrix and optimal weights. Expected returns are calculated using average returns and optimal weights. These products will be added to that list. If the optimization fails on a callback, an error message is printed indicating which callback failed.

Finally, the list of risk values is converted and returned as a numpy array, which can be used for further analysis and drawing optimal bounds. This plan helps investors understand the potential performance of investments at different levels of risk and shows the balance between risk and return for an optimized portfolio.Print a message indicating which response failed.

Finally, the list of risk values is converted and returned as a numpy array, which can be used for further analysis, to filter optimal bounds. This plan helps investors understand the potential performance of investments at different levels of risk and shows the balance between risk and return for an optimized portfolio.

PLOT ANALYSIS:



The graph presents the Markowitz efficient frontier for two investment tasks, Task 1 (blue line) and Task 2 (dashed red line), highlighting the optimal balance between risk (standard deviation) and expected return for different portfolio configurations. Task 1 presents a wide range of risk-return profiles, starting with lower risk and generating progressively higher returns as risk increases. Task 2 follows a similar trajectory but differs from Task 1 by having a lower level of risk, suggesting that it initially uses a slightly less risky strategy. However, both tasks approach a higher level of risk, suggesting similar strategies or asset allocations given higher risks. This efficient limit shows the highest possible return that can be achieved at a given level of risk with different limits or investment strategies and provides investors with valuable information about where they can maximize their return relative to the risk they are willing to take.

TASK 3

Information Arrangement and Initialization:

The script starts by characterizing an cluster, r_vals, which indicates a arrangement of target returns extending from 2.00 to 9.00 in increases of 0.25. This extend speaks to the different return targets for which the portfolio ought to be optimized. Moreover, records risk_vals__si_task3 and return_vals_mu_task3 are initialized to store calculated values for the chance (standard deviation) and anticipated returns of the optimized portfolios, separately. The script at that point calls the generate_information() work to form vectors for the cruel returns (mean_return), the covariance network (covariance_matrix), and the number of resources (num_assets). This information is crucial because it gives the elemental parameters required for portfolio optimization.

Portfolio Optimization :

For each target return indicated in r_vals, the script sets up a portfolio optimization demonstrate utilizing the Gurobi optimizer. It arranges choice factors (x_vars), which speak to the division of add up to capital to be apportioned to each asset, constrained between (no speculation) and 1 (full speculation). The model's objective is to play down the portfolio's add up to chance, which is calculated as the quadratic entirety of the items of the choice factors and the covariance network. Furthermore, the demonstrate forces two imperatives: the return limitation guarantees that the weighted entirety of the anticipated returns of the resources meets or surpasses the target return, and the budget limitation requires that the whole of the resource weights rises to one, showing full assignment of accessible capital.

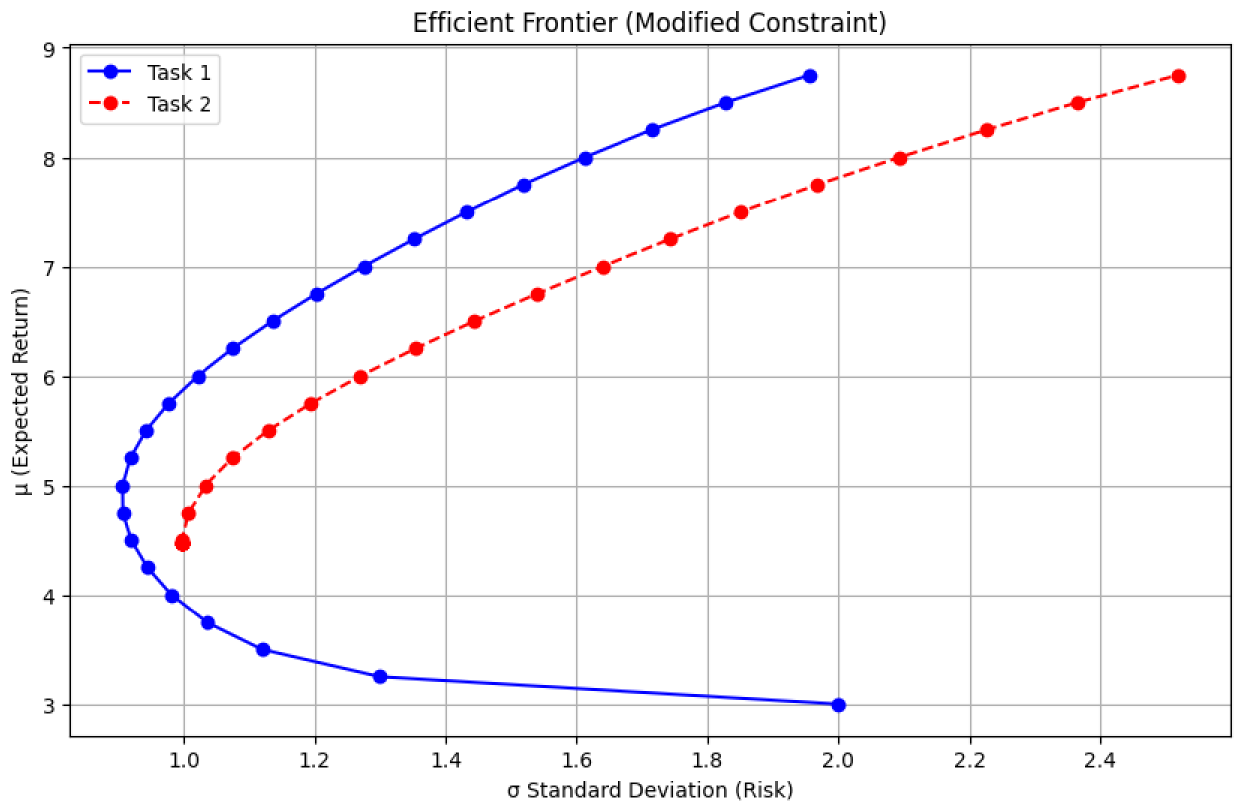
Optimization Execution and Result Dealing with:

The optimization show is executed for each target return. On the off chance that the optimization is effective (status GRB.OPTIMAL), the script calculates the optimal asset weights (x_optimal), from which it infers the portfolio's hazard and anticipated return. These values are at that point put away within the individual records for encourage examination. In case the optimization comes up short to discover a attainable arrangement, an mistake message is printed, indicating the target return that caused the disappointment.

Information Change for Visualization:

After completing the optimizations over all target returns, the script changes over the records containing the computed chance and return information into NumPy clusters (risk_vals__si_task3 and return_vals_mu_task3). This change encourages simpler taking care of and visualization, especially for plotting the proficient wilderness. This plot empowers speculators and examiners to outwardly survey and compare the risk-return profiles of different portfolio arrangements, in this manner helping in vital venture decision-making based on quantitative investigation.

PLOT ANALYSIS:



The graph shows the Markowitz efficient frontier for two different tasks (task 1 in blue, task 2 in red) and shows the optimal trade-offs between risk (standard deviation on the x-axis) and expected return (y-axis). Both problems describe portfolios that maximize expected return for a given level of risk. The curves show that Task 1 starts with lower risk and provides a steady and steady increase in performance as risk increases. In contrast, task 3 starts with a slightly higher risk level and closely follows the trajectory of task 1, then expanding to an even higher expected return at the same risk level, suggesting that task 3 may use a slightly more aggressive investment strategy or different constraints. which allow higher returns at the same risk level. The overlap between the two curves indicates areas where both positions achieve similar returns at similar risk levels, while the difference at higher risk points highlights differences in risk management and return optimization strategies for the two positions. This visualization is crucial for comparing how different portfolio strategies perform under different risk scenarios, helping investors choose the strategy that best matches their risk tolerance and return expectations

Task 4:

Initialization and Information Setup:

The script starts by characterizing an cluster r_vals, which indicates a grouping of target returns extending from 2.00 to 9.00 in increases of 0.25. This cluster sets the anticipated returns that the portfolio optimization demonstrate points to realize. To encourage the optimization handle, the script moreover makes records risk_vals_si_task4 and return_vals_mu_task4 to hold the calculated hazard (as standard deviation) and anticipated returns of the optimized portfolios, separately. Also, the generate_information() work is conjured to create vectors for cruel returns (mean_return), a covariance framework (covariance_matrix), and the number of resources (num_assets). These inputs are basic as they layout the anticipated execution of each resource and the hazard connections between the resources.

Portfolio Optimization Arrangement:

Inside the essential circle, for each indicated target return in r_vals, a unused Gurobi demonstrate is arranged for portfolio optimization. Choice factors (x_vars) are set up for each resource to represent the extent of the whole venture designated to each. These factors are bound to remain continuous values between and 1. The models objective is to play down the overall portfolio hazard, which is measured as the quadratic entirety of the item of these choice factors with the covariance lattice. The show incorporates two basic limitations: a return limitation that guarantees the weighted whole of anticipated returns from the resources meets the desired target return r, and a budget imperative that requires the whole of the speculation divisions to rise to 1, guaranteeing total assignment of accessible capital.

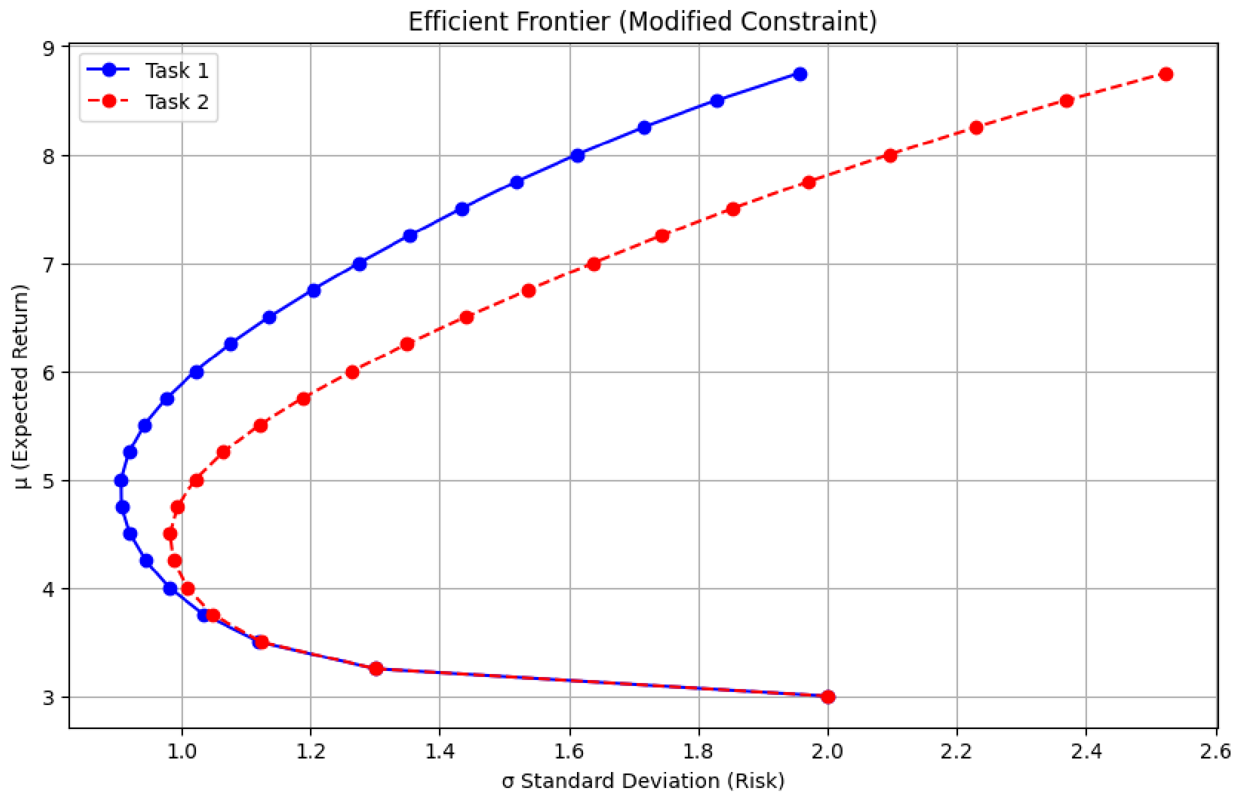
Optimization Execution and Comes about Administration:

The optimization handle is carried out utilizing the model.optimize() strategy. On the off chance that the demonstrate effectively finds an ideal arrangement (status GRB.OPTIMAL), the ideal resource assignments (x_optimal) are recovered. These allotments are at that point utilized to compute the portfolios hazard and anticipated return, which are in this way included to their particular records. In the event that the optimization comes up short, an mistake message is shown, pinpointing the particular target return that caused the issue.

Information Change for Visualization:

After all target returns have been handled, the script changes over the records containing the hazard and return information into NumPy clusters (risk_vals_si_task4 and return_vals_mu_task4). This transformation disentangles dealing with and visualization, especially for plotting the effective wilderness. This plot outwardly illustrates the relationship between chance and return over the optimized portfolios, giving important experiences to financial specialists approximately the ideal risk-return profiles beneath different venture methodologies.

PLOT ANALYSIS:



This graph shows the Markowitz efficient frontier for two distinct investment strategies or tasks, shown as Task 1 (blue solid line) and Task 2 (red dashed line). The efficient frontier depicts the optimal trade-offs of the portfolios between risk (standard deviation, plotted on the x-axis) and expected return (y-axis). The curve in Problem 1 starts at lower risk and rises sharply, indicating that incremental increases in risk are significantly associated with increases in expected return, especially at lower levels of risk. As the curve progresses, the increase in return from additional risk, which is typical of efficient cross-border behavior, decreases. In contrast, task 2 starts with higher risk and initially lower returns than task 1 at similar levels of risk, but then approaches the limit of task 1, suggesting that both tasks may use similar investment strategies or asset allocations with higher levels of risk . The difference in lower risk levels may indicate differences in initial asset selection or risk management methods between the two tasks, with Task 1 perhaps initially focusing on lower-risk, low-return assets than Task 1.