INTELLIGENT REASONING SYSTEM

PROJECT MODULE

PORTFOLIO MANAGEMENT SYSTEM

DHARSHINI CHELLAPPA CHETTY RAJAN

[E1391113@u.nus.edu](mailto:E1391113@u.nus.edu)

A0307202X

Table of Contents

[Portfolio Management System 3](#_Toc180928204)

[1. Introduction 3](#_Toc180928205)

[2. Data Preprocessing and Feature Engineering 4](#_Toc180928206)

[Data Loading: 4](#_Toc180928207)

[Data Aggregation: 4](#_Toc180928208)

[Data Normalization: 4](#_Toc180928209)

[3. Pivot Table for Daily Returns 4](#_Toc180928210)

[4. K-means Clustering for Portfolio Segmentation 4](#_Toc180928211)

[Elbow Method: 4](#_Toc180928212)

[Silhouette Score: 4](#_Toc180928213)

[Clustering and Results: 5](#_Toc180928214)

[5. Stock Selection Based on Risk and Return 5](#_Toc180928215)

[Low-Risk Stock Selection: 5](#_Toc180928216)

[High-Return Stock Selection: 5](#_Toc180928217)

[Stock DataFrames: 5](#_Toc180928218)

[6. Portfolio Weight Allocation Methods 5](#_Toc180928219)

[Equal Weight (EW) Portfolio: 5](#_Toc180928220)

[Market-Cap-Weighted (MCW) Portfolio: 5](#_Toc180928221)

[7. Portfolio Optimization 5](#_Toc180928222)

[Global Minimum Variance (GMV) Portfolio: 5](#_Toc180928223)

[Maximum Sharpe Ratio (MSR) Portfolio: 5](#_Toc180928224)

[8. Performance Evaluation of Portfolio Types 6](#_Toc180928225)

[Equal Weight Portfolio: 6](#_Toc180928226)

[Market-Cap-Weighted Portfolio: 6](#_Toc180928227)

[Global Minimum Variance Portfolio: 6](#_Toc180928228)

[Maximum Sharpe Ratio Portfolio: 6](#_Toc180928229)

[9. LSTM Model for Portfolio Weight Prediction 6](#_Toc180928230)

[Data Preparation and Scaling 6](#_Toc180928231)

[LSTM Model Architecture 7](#_Toc180928232)

[Evaluation and Metrics 7](#_Toc180928233)

[Further Training with Additional Epochs 8](#_Toc180928234)

[10. Conclusion and Next Steps 8](#_Toc180928235)

# Portfolio Management System

## 1. Introduction

This report provides a detailed analysis of portfolio construction using machine learning and optimization techniques. The primary focus is to identify optimal stocks based on risk and return metrics, then cluster and weight these stocks to create a diversified portfolio. The methods applied include k-means clustering, portfolio weight allocation strategies (equal weight, market-cap-weighted, global minimum variance, maximum Sharpe ratio), and followed by optimization for the best portfolios.

## 2. Data Preprocessing and Feature Engineering

### Data Loading:

* Loaded two datasets: final\_preprocessed\_data.csv and financial\_input\_data.csv.
* The main dataset, final\_preprocessed\_data.csv, contains financial metrics for different stocks, while financial\_input\_data.csv provided additional date-specific information.

### Data Aggregation:

* Aggregated dataset1 by the ticker symbol using mean values for key metrics including Daily\_Return, Volume, Adj Close, Market Cap, Beta, P/E Ratio, PC1, PC2, and PC3.

### Data Normalization:

* Excluded Ticker from normalization.
* Standardized selected columns using StandardScaler, enabling fair comparisons across metrics.

## 3. Pivot Table for Daily Returns

* Constructed a pivot table for Daily\_Return, setting Date as the index and Ticker as columns.
* This pivot table helped organize and compute covariance matrices and mean returns for stock returns, facilitating portfolio variance calculations.

## 4. K-means Clustering for Portfolio Segmentation

### Elbow Method:

* To determine the optimal number of clusters (K), evaluated inertia values across a range of K values (2 to 10).
* Observed the "elbow" at K = 3 and K = 4, suggesting these as potentially optimal values for clustering.

### Silhouette Score:

* Calculated silhouette scores for K values from 2 to 10.
* Highest scores observed at K = 3 and K = 4, indicating well-defined clusters.

### Clustering and Results:

* Applied clustering models with K = 3, K = 4, and K = 5 clusters.
* Assigned each stock in aggregated\_dataset1 to its respective cluster, creating new columns (Cluster3, Cluster4, and Cluster5).

## 5. Stock Selection Based on Risk and Return

### Low-Risk Stock Selection:

* Defined the 10 stocks with the lowest Beta values, identifying stocks with minimized volatility.

### High-Return Stock Selection:

* Selected the 10 stocks with the highest Daily\_Return values, focusing on maximizing returns.

### Stock DataFrames:

* Created two data frames: min\_risk\_10stock\_df (low risk) and max\_return\_10stock\_df (high return).

## 6. Portfolio Weight Allocation Methods

### Equal Weight (EW) Portfolio:

* Distributed equal weights across 10 selected stocks (1/10 for each stock).

### Market-Cap-Weighted (MCW) Portfolio:

* Calculated weights proportional to each stock's Market Cap, assigning higher weights to stocks with larger market capitalizations.

### Global Minimum Variance (GMV) Portfolio:

* Minimizing the portfolio volatility while maintaining a sum of weights equal to 1

Maximum Sharpe Ratio (MSR) Portfolio:

* Maximizing the Sharpe ratio, using the risk-free rate of 0.01 in Sharpe ratio calculation.

## 7. Portfolio Optimization

As the GMV and MSR portfolios have high return and low risk, optimizing only the same.

### Global Minimum Variance (GMV) Portfolio:

* Applied the minimize function from scipy.optimize with constraints and bounds.
* Calculated optimal weights for low-risk and high-return stocks (gmv\_weights\_min and gmv\_weights\_max).

### Maximum Sharpe Ratio (MSR) Portfolio:

* Applied the minimize function to maximize portfolio return relative to its risk.
* Obtained optimal weights for MSR portfolios for both risk-minimized and return-maximized stock groups (msr\_weights\_min and msr\_weights\_max).

## 8. Performance Evaluation of Portfolio Types

Defined a portfolio\_performance function to calculate the expected return and volatility for each portfolio type.

### Equal Weight Portfolio:

* Calculated return (ew\_ret\_min, ew\_ret\_max) and volatility (ew\_vol\_min, ew\_vol\_max) for both low-risk and high-return stocks.

### Market-Cap-Weighted Portfolio:

* Calculated return (mcw\_ret\_min, mcw\_ret\_max) and volatility (mcw\_vol\_min, mcw\_vol\_max).

### Global Minimum Variance Portfolio:

* Calculated return (gmv\_ret\_min, gmv\_ret\_max) and volatility (gmv\_vol\_min, gmv\_vol\_max).

### Maximum Sharpe Ratio Portfolio:

* Calculated return (msr\_ret\_min, msr\_ret\_max) and volatility (msr\_vol\_min, msr\_vol\_max).

## 9. LSTM Model for Portfolio Weight Prediction

To forecast portfolio weights based on historical data, we implemented an LSTM model designed to capture temporal dependencies in the portfolio weights for two portfolios: the minimum risk portfolio and the maximum return portfolio. The LSTM model provides predictive insights, enabling future weight allocation adjustments based on trends in historical data.

### Data Preparation and Scaling

#### Historical Weight Data:

* Generated sample historical weights as a placeholder; actual historical data should replace this sample data for production use.
* Scaled the weights between 0 and 1 using MinMaxScaler to normalize the data and improve LSTM performance.

#### Dataset Creation for LSTM:

* Developed a create\_dataset function to organize the data into sequences. This function produces input-output pairs where each output corresponds to the weight allocation following a given input sequence of time\_step length.
* Set time\_step to 5, which refers to the number of previous time steps considered for predicting future weights.

#### Train-Test Split:

* Split the scaled data into training and testing sets using an 80/20 ratio, maintaining consistency in the model evaluation process.

### LSTM Model Architecture

#### Model Layers:

* Built an LSTM model with two LSTM layers (each with 50 units) to capture the time-series patterns in the weight data. The first LSTM layer returns sequences to allow stacking of layers, while the second LSTM layer outputs the final sequence.
* The output layer size matches the number of assets in the portfolio, allowing the model to predict a weight for each asset.

#### Compilation:

* Compiled the model using the adam optimizer with mean\_squared\_error as the loss function and added accuracy as a metric to monitor training performance.

#### Training:

* Trained the model on the training set for 100 epochs with a batch size of 32, observing the progress through training logs. A validation split of 20% was used for assessing model performance during training.

### Evaluation and Metrics

#### Prediction of Portfolio Weights:

* Used the trained model to predict portfolio weights on the test set, which were subsequently inverse scaled to retrieve them in their original range.

#### Performance Metrics:

* Evaluated the model's predictions using three key metrics:
  + **Mean Absolute Error (MAE)**: Measures the average absolute difference between the predicted and actual weights.
  + **Mean Squared Error (MSE)**: Highlights prediction variance, with higher penalties for larger errors.
  + **R-squared (R²)**: Represents the proportion of variance explained by the model; a higher value indicates better performance.

#### Results:

* Minimum Risk Portfolio:
  + **MAE**: Value indicating the average prediction error in terms of absolute weight difference.
  + **MSE**: Reflects variance in prediction errors.
  + **R-squared**: Value close to 1 would indicate good predictive capability.
* Maximum Return Portfolio:
  + Similarly evaluated, showcasing the model's predictive performance for the high-return portfolio.

### Further Training with Additional Epochs

#### Model Fine-tuning:

* Compiled the model with an extended set of metrics: mean\_absolute\_error and mean\_squared\_error for finer evaluation.
* Fine-tuned the model by training for an additional 10 epochs on the training set, validating with 20% of the data to observe improvements in prediction accuracy.

## 10. Conclusion and Next Steps

The implementation of the LSTM model for predicting portfolio weights demonstrates its potential in time-series analysis of asset allocations. Through rigorous data preparation, scaling, and model training, this approach enables data-driven predictions for weight allocation in portfolios. However, future improvements could include:

* **Extended Historical Data**: Integrating additional, more extensive historical weight data to further refine the predictive capability.
* **Parameter Optimization**: Testing various hyperparameters (e.g., LSTM units, batch size) for optimizing model performance.
* **Alternative Models**: Considering advanced time-series models, such as GRUs or Transformer-based models, for further experimentation.