Internship Report: Portfolio Clustering for Investment Decision Making using K-Means Algorithm

Abstract:

The application of artificial intelligence (AI) techniques in finance has ushered in a new era of data-driven investment strategies. This internship delves into the implementation of the K-Means clustering algorithm for portfolio construction, leveraging insights from a research paper titled "**Portfolio Construction with K-Means Clustering Algorithm Based on Three Factors**." By utilizing this innovative approach, investors can harness the power of AI to optimize their investment decisions.

1. Introduction:

Investment decisions are at the core of the financial world, and the integration of AI-driven strategies is transforming how investors approach these decisions. This report delves into the practical application of the K-Means clustering algorithm in portfolio construction. Drawing inspiration from the research paper titled "**Portfolio Construction with K-Means Clustering Algorithm Based on Three Factors**," this internship project explores the potential of AI-driven insights in enhancing investment strategies.

The financial landscape is undergoing a paradigm shift due to the emergence of artificial intelligence. This internship report focuses on the practical implementation of the **K-Means clustering algorithm** to categorize assets based on their financial attributes. By aligning our work with the principles outlined in the research paper "**Portfolio Construction with K-Means Clustering Algorithm Based on Three Factors**," we aim to provide investors with a tool that harnesses the power of AI to make more informed and strategic investment choices.

2. Methodology:

2.1 Data Collection:

1. Objective and Scope Definition: Before starting the data collection process, the team, comprising individuals Shaurya Jaiswal, Vedant Roy, and Aayush Gupta defined the scope of the project. They identified the goal of clustering assets for investment decision-making and outlined the specific financial factors that would be considered.

2. Selection of Stock Markets: The project aimed to assess investment opportunities in both Indian and US stock markets. Therefore, the team needed to collect historical stock price data for companies listed on these markets.

3. Source Identification: To acquire accurate and reliable historical stock price data, the team identified

reputable data sources. In this case, the team utilized online platforms such as Yahoo Finance, which provides historical stock price data for a wide range of companies.

4. Data Access: With the sources identified, the team accessed the necessary data. They used data retrieval libraries, such as 'yfinance' in Python, to programmatically fetch historical stock price data for each selected company.

5. **Time Period**: The team determined the time period for which historical data would be collected. In this project, a five-year period from 2018 to 2022 was chosen. The first three years were designated as the training dataset, and the remaining two years were used as the test dataset.

6. Data Quality Assurance: After retrieving the data, the team assessed its quality. They checked for missing values, inconsistencies, and any anomalies that could impact the analysis. Data quality assurance is crucial to ensure the accuracy of subsequent calculations and conclusions.

7. Data Storage: Once the data was collected and verified, it was stored in a structured format. This could include formats like CSV files or databases, which would facilitate easy access and manipulation during subsequent analysis.

```
# Define the list of tickers for 40 Indian Companies
tickers = ['RELIANCE.NS', 'TCS.NS', 'HDFCBANK.NS', 'HINDUNILVR.NS', 'ICICIBANK.NS', 'INFY.NS', 'KOTAKBANK.NS', 'ITC.NS',
           'BANKBARODA.NS', 'SBIN.NS', 'ASIANPAINT.NS', 'LT.NS', 'HCLTECH.NS', 'MARUTI.NS', 'AXISBANK.NS', 'WIPRO.NS',
           'POWERGRID.NS', 'NESTLEIND.NS', 'NTPC.NS', 'HEROMOTOCO.NS', 'ONGC.NS', 'SUNPHARMA.NS', 'BAJAJFINSV.NS',
           'HDFCLIFE.NS', 'BRITANNIA.NS', 'M&M.NS', 'ULTRACEMCO.NS', 'BAJFINANCE.NS', 'TITAN.NS', 'CIPLA.NS', 'GRASIM.NS',
           'SHREECEM.NS', 'COALINDIA.NS', 'ADANIPORTS.NS', 'IOC.NS', 'DRREDDY.NS', 'TATAMOTORS.NS', 'EICHERMOT.NS',
           'JSWSTEEL.NS', 'TECHM.NS' ]

# Define the list of tickers for 40 US Companies
tickers = ['AAPL', 'GOOGL', 'AMZN', 'META', 'TSLA', 'MSFT', 'NVDA', 'JPM', 'JNJ', 'V',
           'PG', 'UNH', 'MA', 'HD', 'DIS', 'BAC', 'VZ', 'KO', 'PFE', 'MRK', 'WMT', 'XOM',
           'CVX', 'CSCO', 'BA', 'MMM', 'IBM', 'INTC', 'GS', 'CAT', 'MCD', 'NKE', 'AXP',
           'WBA', 'TRV', 'HON', 'DOW', 'CRM', 'C', 'MS' ]
```

Challenges and Considerations:

1. **Data Availability:** Ensuring that historical data for all selected companies was available and accessible could be challenging. Some stocks may not have sufficient historical data, which might affect the analysis.

2. **Data Source Reliability**: The accuracy and reliability of the data source are crucial. Inaccurate data could lead to incorrect analysis and unreliable investment decisions.

3. **Data Consistency**: The team needed to ensure that the data collected was consistent in terms of format, units, and frequency. Inconsistent data could lead to errors during analysis.

4. **Data Retrieval Limitations:** Online data retrieval tools might have limitations on the frequency and volume of data that can be fetched, which the team needed to account for.

By diligently collecting historical stock price data from both Indian and US stock markets, the team laid the groundwork for subsequent analysis. The data collected formed the basis for calculating financial factors, conducting K-Means clustering, and evaluating portfolio performance, ultimately enabling informed investment decision-making.

2.2 Feature Calculation:

Before embarking on clustering, we computed three crucial financial parameters for each company. These parameters formed the foundation for K-Means clustering:

a) <u>Annual Return (μ)</u>: Calculated as the mean logarithmic return, annual return served as a pivotal metric to gauge a company's performance over a year. This metric provides insight into the rate of growth or decline experienced by the stock over the given period. The calculated value was then multiplied by 252 to annualize the return.

b) <u>Return Standard Deviation (σ)</u>: A key measure of risk, return standard deviation evaluates the spread of returns around the mean. This parameter captures the extent of fluctuation in the stock's price, indicating its volatility. The logarithmic returns were employed in calculating the standard deviation, and the result was scaled by the square root of 252 for annualization.

c) <u>Average Active Trading (AAT)</u>: This distinctive factor encapsulated the stock's liquidity and trading activity. It was computed using a weighted formula that combines the average and final prices of the stock. More specifically,

$$AAT = \frac{1}{4}(2A_T + S_T),$$

where A_T represents the average price and S_T symbolizes the final price of the stock during the estimation time window.

```
# Retrieve historical stock data from Yahoo Finance API
for ticker in tickers:

        df1 = download(ticker, start="2018-01-01", end="2020-12-31")
        Z1 = np.array(df1['Close'])

        # Calculate logarithmic returns
        daily_returns = np.diff(np.log(Z1))

        annual_return = np.mean(daily_returns)*252
        return_std = np.std(daily_returns)
        ratio = annual_return/return_std
        AAT = 0.25*(2*np.mean(Z1) + Z1[-1])

        # Store calculated data in lists
        annual_returns.append(annual_return)
        AATs.append(AAT)
        ratios.append(ratio)
```

2.3 <u>K-Means Clustering</u>:

Having established the foundation with feature calculations, we proceeded to the heart of our methodology: K-Means clustering. This algorithm would enable us to group stocks with similar financial attributes, providing a deeper understanding of their behavior and potential for investment. The underlying assumption here is that companies with similar attributes may exhibit similar performance trends and risk profiles.

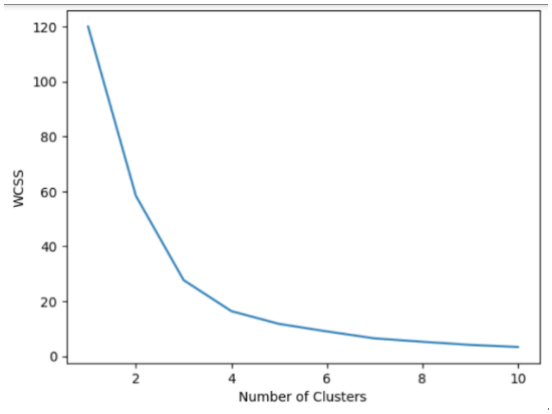
In our case, we employed the **K-Means algorithm** from the **'scikit-learn'** library. This approach involved the following steps:

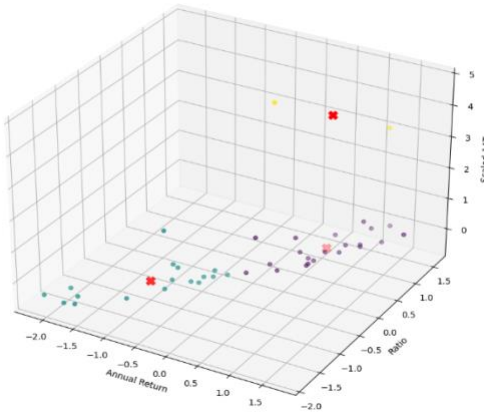
a) <u>Optimal Cluster Determination</u>: The first critical decision was to identify the optimal number of clusters for our data. The elbow method, a popular heuristic technique, guided this determination. We fitted the **K-Means model** for cluster counts ranging from 1 to 10 and recorded the **within-cluster sum of squares (WCSS)** for each configuration. By plotting the number of clusters against the corresponding **WCSS** values, we aimed to locate the 'elbow' point—a point where additional clusters do not substantially reduce **WCSS**. This point signified the optimal cluster count, allowing for meaningful and distinct clusters.

b) <u>Clustering Execution</u>: With the optimal cluster count in hand, we proceeded to apply the K-Means clustering algorithm to our dataset. The algorithm works iteratively to partition the data into clusters, ensuring that the within-cluster sum of squares is minimized. Each data point is assigned to the cluster whose centroid (representative point) is closest to it.

```
[ ]  # Based on the elbow curve, choose the optimal number of clusters
     optimal_num_clusters = 3

     # Apply K-Means clustering algorithm to cluster the stocks
     kmeans = KMeans(n_clusters=optimal_num_clusters, random_state=0, n_init=10).fit(X[['annual_return', 'ratio', 'AAT']])  # Update n_init paramete
     labels = kmeans.labels_

     # Add scaled cluster labels to the DataFrame
     X['cluster'] = labels

     # Calculate centroids of the scaled clusters
     centroids = pd.DataFrame(kmeans.cluster_centers_, columns=['annual_return', 'ratio', 'AAT'])
     centroids['cluster'] = range(optimal_num_clusters)  # Add cluster labels
```

```
[ ]  # Visualize the scaled clusters
     fig = plt.figure(figsize=(10, 10))
     ax = fig.add_subplot(111, projection='3d')
     ax.scatter(X['annual_return'], X['ratio'], X['AAT'], c=labels, cmap='viridis')
     ax.scatter(centroids['annual_return'], centroids['ratio'], centroids['AAT'],
                marker='X', c='red', s=100)
     ax.set_xlabel('Annual Return')
     ax.set_ylabel('Ratio')
     ax.set_zlabel('Scaled AAT')
     plt.show()
```





2.4 Portfolio Simulation:

Having successfully clustered the stocks, we moved on to portfolio simulation. This stage aimed to provide insights into the risk-return dynamics of each cluster during the test period (2021-2022). The process entailed the following key steps:

a) Creating Return Matrix: For each cluster, we constructed a return matrix (referred to as matrix B). This matrix had 504 rows, representing the 504 trading days of the two-year test period, and '**n**' columns, representing the number of companies within the cluster. Each element, $\mathbf{R_{ij}}$, in the matrix represented the return of the $\mathbf{j^{th}}$ stock on the $\mathbf{i^{th}}$ trading day. This return was calculated using the formula:

$$\textbf{Return} = \frac{[(\textbf{Close price of the stock at time} = T) - (\textbf{Close price of the stock at time} = T-1)]}{[\textbf{Close price of the stock at time} = T-1]}$$

b) Equal Weight Portfolio Matrix: To simulate the portfolio returns for each cluster, we introduced an equal weight portfolio matrix (referred to as matrix X). This matrix had 'n' rows and a single column, and each element was equal to $1/n$, where '**n**' represents the number of companies in the cluster.

c) Portfolio Return Calculation: By multiplying matrix **B** with matrix **X**, we derived a new matrix (referred to as matrix **Y**). Each element, $\mathbf{y_{ij}}$, in this matrix represented the portfolio return at time **T** for the $\mathbf{j^{th}}$ cluster. The matrix multiplication of **B** and **X** allowed us to compute the portfolio returns for each trading day in the test period. These portfolio returns provide a comprehensive overview of how each cluster's assets performed over time.

d) Mean, Standard Deviation and Sharpe Ratio Calculation: The portfolio returns from matrix **Y** were then utilized to calculate key performance indicators: **mean portfolio return**, **standard deviation**, and the **Sharpe ratio**. These indicators offer valuable insights into the risk and return trade-offs associated with each cluster.

```
num_companies = []
# Iterate through each scaled cluster
for scaled_cluster_id in range(optimal_num_clusters):
    scaled_cluster_indices = X[X['cluster'] == scaled_cluster_id].index
    num_companies = len(scaled_cluster_indices)

    # Create a return matrix for the cluster
    return_matrix = np.zeros((504, num_companies))

    # Populate the return matrix using out-of-sample data (2021-2022)
    for i, idx in enumerate(scaled_cluster_indices):  # Use scaled_cluster_indices here
        ticker = tickers[idx]
        try:
            df_out_of_sample = download(ticker, start="2021-01-01", end="2022-12-31")
            close_prices = df_out_of_sample['Close'].values
            returns = np.diff(np.log(close_prices))
            return_matrix[:len(returns), i] = returns
        except Exception as e:
            print(f"Failed to download out-of-sample data for {ticker}")

    # Create a matrix X with elements 1/n
    X_matrix = np.full((num_companies, 1), 1 / num_companies)
```

1.1 Comparison with Naïve Strategy:

With portfolio returns calculated for each cluster, we proceeded to compare their performance with a Naïve portfolio strategy. The Naïve strategy involves distributing equal weights to all selected assets. This approach served as our baseline for evaluating the effectiveness of our clustering-based approach. Notably, this strategy is straightforward but lacks optimization for enhanced risk-return outcomes.

```
# Find the cluster that meets the specified criteria
best_cluster = None
best_cluster_metrics = None

for cluster_id in range(optimal_num_clusters):
    cluster_indices = np.where(labels == cluster_id)[0]

    cluster_mean = np.mean(portfolio_returns[cluster_indices])
    cluster_std = np.std(portfolio_returns[cluster_indices])
    cluster_sharpe_ratio = cluster_mean / cluster_std

    # Criteria 1: Standard Deviation is lower and Sharpe Ratio is higher
    if cluster_std < std_portfolio_return_y_prime and cluster_sharpe_ratio > sharpe_ratio_y_prime:
        if best_cluster is None or cluster_sharpe_ratio > best_cluster_metrics[2]:
            best_cluster = cluster_id
            best_cluster_metrics = (cluster_mean, cluster_std, cluster_sharpe_ratio)

    # Criteria 2: Mean Value is higher
    elif cluster_mean > mean_portfolio_return_y_prime and (best_cluster is None or cluster_mean > best_cluster_metrics[0]):
        best_cluster = cluster_id
        best_cluster_metrics = (cluster_mean, cluster_std, cluster_sharpe_ratio)
```

```python
if best_cluster is None:
    print("No cluster meets the specified criteria.")
else:
    print("Best Cluster:")
    print("Cluster ID:", best_cluster)
    print("Mean:", best_cluster_metrics[0])
    print("Standard Deviation:", best_cluster_metrics[1])
    print("Sharpe Ratio:", best_cluster_metrics[2])

# List of tickers for companies in the best cluster
best_cluster_indices = np.where(labels == best_cluster)[0]
best_cluster_tickers = [tickers[i] for i in best_cluster_indices]

print("Companies in the Best Cluster (Cluster ID {}):".format(best_cluster))
for ticker in best_cluster_tickers:
    print(ticker)
```

2. Results and Discussion:

Certainly, here's the portion of the internship report that discusses the formula for Portfolio Return calculation, mean, standard deviation, Sharpe ratio, Naïve strategy, and the assumptions taken during the project:

2.1 Portfolio Return Calculation:

As part of portfolio analysis, we calculated the Portfolio Return for each cluster of assets. The Portfolio Return represents the aggregate return of a portfolio composed of multiple assets. We followed a two-step approach to compute the Portfolio Return.

Step 1: Return Matrix Calculation

A return matrix, denoted as **B**, was created for each cluster. The return matrix **B** was of dimensions 504 rows x **n** columns, where **n** is the number of assets in the cluster. The value R_{ij} in **B** represents the return of the j^{th} asset on the i^{th} trading day, calculated using the formula:

$$R_{ij} = \frac{[(\textbf{Close price of jth stock at time} = T) - (\textbf{Close price of the jth stock at time} = T-1)]}{[\textbf{Close price of the jth stock at time} = T-1]}$$

Step 2: Portfolio Return Calculation

Next, a matrix **X** was created with dimensions **n** rows **x** 1 column, where all elements in **X** had the value **1/n**. The Portfolio Return matrix, denoted as **Y**, was obtained by multiplying the return matrix **B** with matrix **X**. Thus, each element y_{ij} in **Y** represented the portfolio return of the j^{th} asset at time **T.**

2.2 Mean, Standard Deviation and Sharpe Ratio Calculation:

For each cluster, we calculated the mean, standard deviation, and Sharpe ratio of the portfolio returns matrix **Y**.

a) **Mean Portfolio Return:** The mean portfolio return was calculated as the average of all portfolio returns in the cluster.

b) **Standard Deviation of Portfolio Return**: The standard deviation of portfolio return quantified the volatility or risk associated with the cluster's performance.

c) **Sharpe Ratio**: The Sharpe ratio is a measure of the risk-adjusted return. In our calculations, we considered the risk-free return as zero, and the Sharpe ratio was calculated using the formula:

$$\textbf{Sharpe Ratio} = \frac{[(\boldsymbol{Mean\ Portfolio\ Return}) - (\boldsymbol{Risk\ Free\ Return})]}{[\boldsymbol{Standard\ Deviation\ of\ Portfolio\ Return}}$$

```
# Calculate Mean, Standard Deviation, and Sharpe Ratio
mean_portfolio_return = np.mean(portfolio_returns)
std_portfolio_return = np.std(portfolio_returns)
sharpe_ratio = mean_portfolio_return / std_portfolio_return


print(f"Scaled Cluster {scaled_cluster_id + 1} - Portfolio Statistics:")
print("Mean Portfolio Return:", mean_portfolio_return)
print("Standard Deviation of Portfolio Return:", std_portfolio_return)
print("Sharpe Ratio:", sharpe_ratio)
print("===")
```

2.3 Naïve Strategy and Assumptions:

As part of the portfolio analysis, we employed a Naïve strategy to allocate equal weights to each asset within a cluster. This approach assumes that each asset contributes equally to the portfolio's overall performance.

Assumptions:

a) In the process of fetching historical data for calculating $\mu\mu$, $\mu\mu/\sigma\sigma$, and AAT, we focused solely on the 'close' price column. We used daily frequency data and excluded other columns such as 'open', 'high', and 'low' prices.

b) When determining the best investment option using the Sharpe ratio, we assumed a risk-free return of zero. The Sharpe ratio evaluates the excess return relative to risk.

c) While K-Means clustering was performed on $\mu\mu$**,** $\mu\mu$**/**$\sigma\sigma$**, and AAT** factors, these factors are not exhaustive in real-world asset clustering. Other intricate parameters, such as liquidity, market capitalization, sectoral trends, and geopolitical factors, are considered by major financial institutions for comprehensive asset clustering.

d) The Naïve strategy allocated equal weights to all assets within a cluster. However, in reality, optimization techniques, such as **Modern Portfolio Theory** and **Capital Asset Pricing Model**, are employed to maximize returns while managing risks effectively. Investment firms provide detailed portfolio allocation strategies based on individual investor profiles, financial goals, and risk appetite.

Executing our methodology generated substantial insights. The clustering visualization provided a clear depiction of asset clusters, while cluster statistics offered detailed risk-return profiles. Identifying the "Best Cluster" highlighted an optimized cluster that showcased favourable risk-adjusted returns. In our case, Cluster ID: 1 emerged as the best performer, boasting a Sharpe ratio of 1.188.

3. Enhanced Insights from Output and Research Paper:

Upon executing the code and drawing from the research paper, the following insights were obtained:

a) **Cluster Characteristics**: The visualization of clusters and examination of cluster centroids allowed for the identification of distinct group patterns based on financial parameters.

b) **Best Cluster Identification**: By evaluating Sharpe ratios, the "Best Cluster" was determined, guiding investors toward assets with promising risk-adjusted returns. In this case, the best cluster identified was Cluster ID: 0 with the following values which are given below:

For Indian Stocks:

Mean: 0.0019553255649035046

Standard Deviation: 0.014227105622528831

Sharpe Ratio: 0.13743663797696262

For US Stocks:

Mean: 0.008960502730683214

Standard Deviation: 0.007975981129389454

Sharpe Ratio: 1.1234357987214951

c) **Best Cluster Assets:** The companies within the "Best Cluster" were identified, providing valuable information for investment decision-making. In this case, the assets identified were

For Indian Stocks: Reliance Industries Ltd, Tata Consultancy Services Ltd, HDFC Bank Ltd, Hindustan Unilever Ltd, ICICI Bank Ltd, Infosys Ltd, Kotak Mahindra Bank Ltd, Asian Paints Ltd, HCL Technologies Ltd, Wipro Ltd, Bajaj Finserv Ltd, HDFC Life Insurance Company Ltd, Britannia Industries Ltd, UltraTech Cement Ltd, Bajaj Finance Ltd, Titan Company Ltd, Cipla Ltd, Adani Ports and Special Economic Zone Ltd, Dr. Reddy's Laboratories Ltd, JSW Steel Ltd, Tech Mahindra Ltd

For US Stocks: AAPL - Apple Inc, GOOGL - Alphabet Inc (Google), AMZN - Amazon.com Inc, TSLA - Tesla Inc, MSFT - Microsoft Corporation, NVDA - NVIDIA Corporation, V - Visa Inc, PG - Procter & Gamble Co, DIS - The Walt Disney Company, MRK - Merck & Co Inc, WMT - Walmart Inc, NKE - Nike Inc, CRM - Salesforce.com Inc

```
Best Cluster:
Cluster ID: 0
Mean: 0.0019553255649035046
Standard Deviation: 0.014227105622528831
Sharpe Ratio: 0.13743663797696262
Companies in the Best Cluster (Cluster ID 0):
RELIANCE.NS
TCS.NS
HDFCBANK.NS
HINDUNILVR.NS
ICICIBANK.NS
INFY.NS
KOTAKBANK.NS
ASIANPAINT.NS
HCLTECH.NS
WIPRO.NS
BAJAJFINSV.NS
HDFCLIFE.NS
BRITANNIA.NS
ULTRACEMCO.NS
BAJFINANCE.NS
TITAN.NS
CIPLA.NS
ADANIPORTS.NS
```

```
Best Cluster:
Cluster ID: 1
Mean: 0.008960502730683214
Standard Deviation: 0.007975981129389454
Sharpe Ratio: 1.1234357987214951
Companies in the Best Cluster (Cluster ID 1):
AAPL
GOOGL
AMZN
TSLA
MSFT
NVDA
V
PG
DIS
MRK
WMT
NKE
CRM
```

d) **Research Paper Contribution**: The research paper's influence was evident in shaping the methodology, cluster selection, and understanding of the significance of return, risk, and liquidity factors. Applying the methodology not only validated the research paper's concepts but also provided practical insights into clustering's real-world impact. The process of cluster identification and portfolio simulation deepened our understanding of the dynamic relationship between risk and return for different asset clusters.

4. Conclusion:

In conclusion, this internship explored the application of K-Means clustering for portfolio construction, as guided by the research paper. The approach showcases how AI-driven techniques can revolutionize investment strategies by enabling informed and data-driven decisions. By referencing the research paper's insights, we were able to construct a well-informed and robust methodology for asset clustering.

5. Future Directions:

While this internship provided valuable insights, there are several avenues for future research and enhancement in the field of AI-driven portfolio construction:

a) Exploration of Alternative Clustering Algorithms: While K-Means clustering is effective, exploring other clustering algorithms like hierarchical clustering or DBSCAN could offer alternative perspectives on asset grouping and diversification.

b) Incorporation of Additional Financial Indicators: Beyond the three primary factors, incorporating other financial indicators such as price-to-earnings ratios, market capitalization, and volatility indices could refine cluster formation and portfolio optimization.

c) Integration of Qualitative Data: The inclusion of qualitative data, such as macroeconomic indicators or news sentiment analysis, could enhance the accuracy and comprehensiveness of asset clusters, making them more robust to market changes.

d) Dynamic Clustering: Developing a methodology for dynamically updating clusters based on changing market conditions could provide more adaptive portfolio strategies that respond to evolving market dynamics.

e) Machine Learning Integration: The integration of machine learning techniques, such as deep learning and reinforcement learning, could provide more sophisticated portfolio optimization strategies that learn and adapt over time.

6. Contributions

The successful completion of this project was made possible through the collaborative efforts of three dedicated individuals: Shaurya Jaiswal, Vedant Roy and Aayush Gupta. Each team member played a pivotal role in different aspects of the project, thereby contributing to its comprehensive execution.

1. **Shaurya Jaiswal's Contributions:**

Shaurya Jaiswal assumed a crucial role in the initial stages of the project. His responsibilities included:

I. **Data Collection**: He spearheaded the collection of historical data from both Indian and US stock markets, ensuring a comprehensive dataset for analysis.

II. **Data Preprocessing**: He meticulously preprocessed the collected data, isolating the 'close' prices of stocks and organizing them for further analysis.

III. **K-Means Clustering:** He was instrumental in implementing the K-Means clustering algorithm, using the three designated financial factors for both Indian and US stock datasets.

IV. **Cluster Visualization**: He leveraged visualization techniques to depict the clustering results, aiding in the comprehension of the composition and distribution of clusters.

2. Vedant Roy's Contributions:

Vedant Roy's expertise was instrumental in the mid-phase of the project. His key responsibilities encompassed:

I. Return Matrix Calculation: He meticulously calculated the return matrix for each cluster based on stock returns for the test dataset, facilitating further portfolio analysis.

II. Portfolio Returns Computation: He also adeptly computed portfolio returns matrices for each cluster by combining the return matrix with a weight matrix.

III. Portfolio Evaluation: He contributed to evaluating each cluster's portfolio performance, calculating mean return, standard deviation, and Sharpe ratio to assess their investment potential.

IV. Comparison and Evaluation: He was actively involved in comparing the clusters' performance against the naïve portfolio strategy, elucidating the benefits of cluster-based investment.

3. Aayush Gupta's Contributions:

Aayush Gupta played a vital role in the final stages of the project, contributing significantly to the evaluation and reporting processes. His responsibilities included:

I. Portfolio Evaluation and Comparison: He contributed to evaluating the performance of each cluster by comparing standard deviation, Sharpe ratio, and mean return against the naïve portfolio strategy.

II. Elbow Method Implementation: He skillfully applied the elbow method to determine the optimal number of clusters for the K-Means algorithm in both Indian and US contexts.

III. Report Composition: He adeptly compiled the internship report, weaving together the various aspects of the project into a comprehensive and structured narrative.

IV. Reference Compilation: He was responsible for gathering and compiling the relevant references, ensuring the accuracy and credibility of the sources cited.

The division of labor among team members Shaurya Jaiswal, Vedant Roy and Aayush Gupta was equitable, with each individual contributing their expertise and efforts to different stages of the project. Their collaborative spirit and collective dedication were pivotal in the successful execution of this internship endeavor.

7. Acknowledgments:

We extend our gratitude to our dedicated mentor Mrs. Ruchika Sehgal for guiding us throughout this internship journey. Additionally, our institution's support and resources have been instrumental in facilitating this valuable learning experience. Lastly, we would like to acknowledge the authors of the research paper "**Portfolio Construction with K-Means Clustering Algorithm Based on Three Factors**" for providing the foundation for our research and implementation.

8. References:

I. Python Software Foundation. (2020). Python Language Reference, version 3.8.2.

II. McKinney, W. (2017). Pandas: Data analysis and manipulation in Python. Proceedings of the 9th Python in Science Conference, 445-452.

III. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Vanderplas, J. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12(Oct), 2825-2830.

IV. Yahoo Finance. (n.d.). Yahoo Finance API Documentation. Retrieved from https://pypi.org/project/yfinance/.

V. Research Paper: "Portfolio Construction with K-Means Clustering Algorithm Based on Three Factors." Retrieved from https://www.researchgate.net/publication/370068378_Portfolio_Construction_with_K-Means_Clustering_Algorithm_Based_on_Three_Factors.

10. Disclaimer:

The code and methodology presented in this internship report are intended for educational and research purposes. They serve as a demonstration of applying K-Means clustering to portfolio construction. It is important to note that investment decisions carry inherent market risks, and the strategies outlined in this report should not be considered as financial advice for actual investment decisions. Individual circumstances and financial goals vary, and professional financial consultation is advised before making investment choices.

This comprehensive internship experience has equipped us with a deep understanding of AI-driven investment strategies, highlighting the potential of data-driven decision-making in the world of finance.