Algorithmic Trading and Portfolio Optimization

Ayanika Bera

Institute of Engineering and Management, Kolkata (2021-2025)

Project Team - 15

Project Guide: Subhadip Boral

Report submitted to: IDEAS – Institute of Data Engineering, Analytics and Science Foundation, ISI Kolkata

1. Abstract

This study explores a systematic approach to algorithmic trading through the collection, analysis, and optimization of a selected stock portfolio. The process began by gathering data on NIFTY200 stocks, followed by a thorough fundamental analysis to evaluate key financial metrics. These metrics were used to cluster the stocks using K-Means clustering and Agglomerative Clustering into groups relating them together with respect to their profitability, volatility, and risk and stability.

Subsequently, stocks from these clusters were utilized to perform portfolio optimization using Monte Carlo Simulation, aiming to maximize returns while managing risk. The optimization process was performed across three sets of stocks- stocks selected by clustering, stocks rejected by clustering, and taking all the NIFTY200 stocks together. A portfolio of Rs. 1 crore is taken. We run the Monte Carlo simulation across the 3 sets of stocks on our portfolio value to see how much profit is generated from an investment date of 1st April, 2020 to selling them on 30th April, 2024. From the stocks obtained from this analysis, we test their returns between May 2024 to June 2024.

The outcome of this strategy was analyzed to determine the profit percentage, demonstrating the potential of using algorithmic techniques for informed and systematic trading decisions. The findings and methodologies used are discussed ahead in greater details.

2. Introduction

A stock market index is a key real-time indicator that is used by investors to track and monitor the performance of stocks and other financial assets. One such index used for stocks listed at the National Stock Exchange (NSE) is Nifty 200. The Nifty 200 Index is designed to reflect the behaviour and performance of large and mid-market capitalization companies. Nifty 200 includes all companies forming part of Nifty 100 and Nifty Full Midcap 100 index. Here, NIFTY200 stocks are analysed and grouped algorithmically, based on machine learning algorithms and mathematical simulations applied on it.

What is fundamental analysis?

Fundamental analysis focuses on the economic forces of supply and demand that causes prices to move higher, lower, or stay the same. It is a method of evaluating a stock by attempting to measure its intrinsic value. Analysts who follow this method study everything from the overall economy and industry conditions to the financial condition and management of companies. If this intrinsic value is under the current market price, then the market is overpriced and should be sold. If market price is below the intrinsic value, then then market is undervalued and should be bought.
Some of the common fundamental analysis indicators used in this project include-
Beta, revenue growth, earning growth, return on equity, return on assets, price-to-equity ratio, EBITDA, operating cashflow, debt-to-equity ratio, current ratio, and price-to-book ratio.

What is technical analysis?
Technical analysis is the study of market action, primarily through the use of charts and graphs, for the purpose of forecasting future price trends. It is based on the idea that if a trader can identify previous market patterns, they can form a fairly accurate prediction of future price trajectories. Some of the most commonly used technical indicators include-
Garman-Klass volatility, moving average convergence divergence (MACD), Relative Strength Index (RSI), Average True Range (ATR), and Bollinger bands.
Technical analysis has not been used in this project.

Fundamental Analysis VS Technical Analysis

A fundamental analyst studies the cause of market movement, while the technical analyst studies the effect. Fundamental analysis is better for long term investment, while technical analysis is better for short term investment, as it readily captures bullish and bearish markets. Fundamental analysis will dismiss these sudden trends. This project makes the use fundamental analysis as technical analysis indicators did not create well defined clusters.

What is algorithmic trading?

Algorithmic approaches have revolutionized the financial markets by automating trading decisions using complex algorithms and high-speed data analysis. This approach minimizes human intervention, reduces errors, and allows for the execution of trades at optimal speeds and prices. With the increasing availability of financial data and advancements in computational technology, algorithmic trading has become a crucial tool for traders and investors aiming to maximize returns and manage risk effectively.

Technology Involved

The project leverages several key technologies and methodologies, including:

- **Python:** YFinance used for gathering data; Pandas and NumPy for file reading, data manipulation and analysis; matplotlib and seaborn for plotting and visualization; Scikit-Learn (sklearn) for performing K-Means clustering and scaling the date; and SciPy for performing Agglomerative Clustering.
- **Portfolio Optimization:** Monte Carlo simulation is a powerful technique used in portfolio optimization to assess the potential outcomes of different investment strategies or different allocations under varying conditions. Implementing Monte Carlo simulation in Python involves combining statistical analysis, simulation, and optimization techniques to gain insights into portfolio performance under different allocations.

Data

Stocks belonging to Nifty 200 index is found out. Their corresponding prices are calculated using the yfinance package for necessary computation.

Procedure Used

The project followed a structured methodology:

1. **Data Collection:** Gathered data on Nifty200 stocks using Python's yfinance package..
2. **Fundamental Analysis:** The intrinsic values of a stock are found out using yfinance package.
3. **Clustering:** Applied K-means clustering and Agglomerative Clustering to categorize the stocks into distinct clusters based on their fundamental metrics.
4. **Portfolio Optimization:** Used Monte Carlo Simulation to select stocks with the most weightage that provided the highest profit percentages.
5. **Backtesting:** Used the optimized portfolio obtained for 1st Arpil 2020 – 30th April 2024 to check how profitable they are for May 2024-June 2024.

3. Project Objective

The project aims to find-

1. Find how accurate fundamental analysis is in investing
2. Whether K-Means Clustering and Agglomerative Clustering are good at segregating stocks based on fundamental analysis indicators
3. See if Monte Carlo Simulation works in portfolio optimization
4. Find out performances of stocks by this method
5. To check if the stocks chosen by K-Means and Agglomerative clustering coupled with Monte Carlo simulation produce the maximum profit percentage.

4. Methodology

Importing Packages

```
import pandas as pd    #read files and manipulate dataframes
import numpy as np   #used in creation of n-dimensional arrays and for computaion
import yfinance as yf  #Yahoo Finance package to read relevant data related to stock
import seaborn as sns  #for plotting and visualization
from sklearn.preprocessing import StandardScaler  #to standardize the data before clustering
from sklearn.cluster import KMeans  # to perform K-Means
import matplotlib.pyplot as plt  #for plotting and visualization
from sklearn.metrics import silhouette_score  #Silhouette Score used to find optimum 'k' for K-means
import scipy.cluster.hierarchy as sch  #used for agglomerative clustering and clreating dendrograms
```

Necessary packages along with their functions

Data Collection

NIFTY 200 stocks are used in this project. The index information is collected from this site - https://www.niftyindices.com/indices/equity/broad-based-indices/nifty-microcap-250-dummy/nifty-200.

1	Company Name	Industry	Symbol	Series	ISIN Code
2	ABB India Ltd.	Capital Goods	ABB	EQ	INE117A01022
3	ACC Ltd.	Construction Materials	ACC	EQ	INE012A01025
4	APL Apollo Tubes Ltd.	Capital Goods	APLAPOLLO	EQ	INE702C01027
5	AU Small Finance Bank Ltd.	Financial Services	AUBANK	EQ	INE949L01017
6	Adani Energy Solutions Ltd.	Power	ADANIENSOL	EQ	INE931S01010
7	Adani Enterprises Ltd.	Metals & Mining	ADANIENT	EQ	INE423A01024
8	Adani Green Energy Ltd.	Power	ADANIGREEN	EQ	INE364U01010
9	Adani Ports and Special Economic Zone Ltd.	Services	ADANIPORTS	EQ	INE742F01042
10	Adani Power Ltd.	Power	ADANIPOWER	EQ	INE814H01011
11	Adani Total Gas Ltd.	Oil Gas & Consumable Fuels	ATGL	EQ	INE399L01023
12	Aditya Birla Capital Ltd.	Financial Services	ABCAPITAL	EQ	INE674K01013
13	Aditya Birla Fashion and Retail Ltd.	Consumer Services	ABFRL	EQ	INE647O01011
14	Alkem Laboratories Ltd.	Healthcare	ALKEM	EQ	INE540L01014
15	Ambuja Cements Ltd.	Construction Materials	AMBUJACEM	EQ	INE079A01024
16	Apollo Hospitals Enterprise Ltd.	Healthcare	APOLLOHOSP	EQ	INE437A01024
17	Apollo Tyres Ltd.	Automobile and Auto Components	APOLLOTYRE	EQ	INE438A01022
18	Ashok Leyland Ltd.	Capital Goods	ASHOKLEY	EQ	INE208A01029

Excel sheet containing data

Next, we read this .csv file and extract the “Symbol”. We will use this symbol to extract the relevant data from yfinance package.

```
In [6]:  symbol_list = df_ticker['Symbol'].unique().tolist()
         symbol_list

Out[6]: ['ABB',
         'ACC',
         'APLAPOLLO',
         'AUBANK',
         'ADANIENSOL',
         'ADANIENT',
         'ADANIGREEN',
         'ADANIPORTS',
         'ADANIPOWER',
         'ATGL',
         'ABCAPITAL',
         'ABFRL',
         'ALKEM',
         'AMBUJACEM',
         'APOLLOHOSP',
         'APOLLOTYRE',
         'ASHOKLEY',
         'ASIANPAINT',
         'ASTRAL',
         'AUROPHARMA',
         'DMART',
         'AXISBANK',
         'BSE',
```

This is how ‘Symbols’ are stores and collected.

Next, fundamental analysis metrics are fetched from yfinance for every stock. The following is a sample code to find out ‘beta’ value. We will be replacing ‘beta’ with other fundamental indicators and store it in our dataframe. Here, ‘.NS’ is added after each ticker as yfinance stores NSE stocks with a “.NS” suffix. Some stocks might not contain the indicators we are about to find, so it is necessary to see if they exist, otherwise add None.

In [7]:
```python
beta_values = []

for i in symbol_list:
    ticker = yf.Ticker(i + ".NS")
    if 'beta' in ticker.info:
        beta_value = ticker.info['beta']
        beta_values.append(beta_value)
    else:
        beta_values.append(None)

df_ticker['beta_value'] = beta_values
df_ticker
```

Out[7]:

	Company Name	Industry	Symbol	Series	ISIN Code	beta_value
0	ABB India Ltd.	Capital Goods	ABB	EQ	INE117A01022	0.721
1	ACC Ltd.	Construction Materials	ACC	EQ	INE012A01025	0.401
2	APL Apollo Tubes Ltd.	Capital Goods	APLAPOLLO	EQ	INE702C01027	0.794
3	AU Small Finance Bank Ltd.	Financial Services	AUBANK	EQ	INE949L01017	0.841
4	Adani Energy Solutions Ltd.	Power	ADANIENSOL	EQ	INE931S01010	1.170
...	...	...	...	...	...	...
196	Wipro Ltd.	Information Technology	WIPRO	EQ	INE075A01022	0.745

Similarly, we find relevant values for ‘revenueGrowth’, ‘earningsGrowth’, ‘returnOnEquity’, ‘returnOnAssets’, ‘trailingPE’, ‘forwardPE’, ‘ebitda’, ‘debtToEquity’, ‘currentRatio’, ‘priceToBook’, and ‘operatingCashflow’. ticker.info from yfinance package contains information on all the properties mentioned here.

In [55]:
```python
df_ticker
```

revenue_growth	earning_growth	return_on_equity	trailing_priceeq	forward_priceeq	EBITDA	returnOnAssets	debtToEquity	currentRatio	priceToBook	operatingCashflow
0.278	0.875	NaN	138.646350	84.042280	1.785817e+10	NaN	0.824	NaN	29.263670	NaN
0.129	3.010	0.15334	21.634583	19.626990	2.904540e+10	0.06174	2.173	1.595	3.089009	2.995110e+10
0.076	-0.156	0.22161	58.400910	NaN	1.192170e+10	0.09743	31.744	1.339	11.868713	1.111560e+10
0.169	-0.129	0.13041	28.150414	30.353607	NaN	0.01537	NaN	NaN	3.430124	-1.520906e+11
0.402	-0.022	0.08997	98.413320	NaN	6.102000e+10	0.04884	270.520	1.011	8.866378	6.037620e+10
...	...	...	...	...	...	...	...	...	...	...
-0.042	-0.034	0.14497	26.891504	21.821194	1.620290e+11	0.07441	21.917	2.577	3.898649	1.762160e+11

Fundamental indicators added to the dataframe

Fundamental Indicators

This section contains the fundamental indicators used in the analysis along with a brief description.

1. **Beta Value**: Measures a stock's volatility relative to the market. A beta greater than 1 indicates higher volatility, while a beta less than 1 indicates lower volatility. It assesses risk and market sensitivity. High-beta stocks might offer higher returns but come with greater risk.
2. **Revenue Growth**: Indicates how fast a company's revenue is increasing over time. Strong revenue growth can be a sign of a growing company and potential future profitability.

3. **Earnings Growth**: Shows the increase in a company's earnings over time. Consistent earnings growth is often a sign of a healthy and potentially profitable company.
4. **Return on Equity (ROE)**: Measures the profitability relative to shareholders' equity. High ROE indicates efficient use of equity to generate profits.
5. **Return on Assets (ROA)**: Measures how efficiently a company uses its assets to generate profit. Higher ROA indicates more efficient asset use.
6. **Trailing Price-to-Earnings (P/E) Ratio**: Current share price divided by the earnings per share (EPS) over the last 12 months. Compare with industry averages to determine if the stock is overvalued or undervalued.
7. **Forward Price-to-Earnings (P/E) Ratio**: Current share price divided by the estimated future earnings per share. Helps gauge future growth expectations and valuation.
8. **EBITDA**: Earnings before interest, taxes, depreciation, and amortization. Provides a clearer view of a company's operational profitability by excluding non-operational expenses.
9. **Debt-to-Equity Ratio**: Measures a company's financial leverage. High ratio indicates more debt, which could be risky in a downturn.
10. **Current Ratio**: Current assets divided by current liabilities. Indicates the company's ability to pay short-term obligations.
11. **Price-to-Book (P/B) Ratio**: Current share price divided by book value per share. A high P/B ratio suggests a stock could be overvalued, while a lower P/B ratio could mean the stock is undervalued. Traditionally, any value under 1.0 is considered desirable for value investors, indicating an undervalued stock may have been identified.
12. **Operating Cash Flow**: Cash generated from operations. Reflects the company's ability to generate cash from core operations.

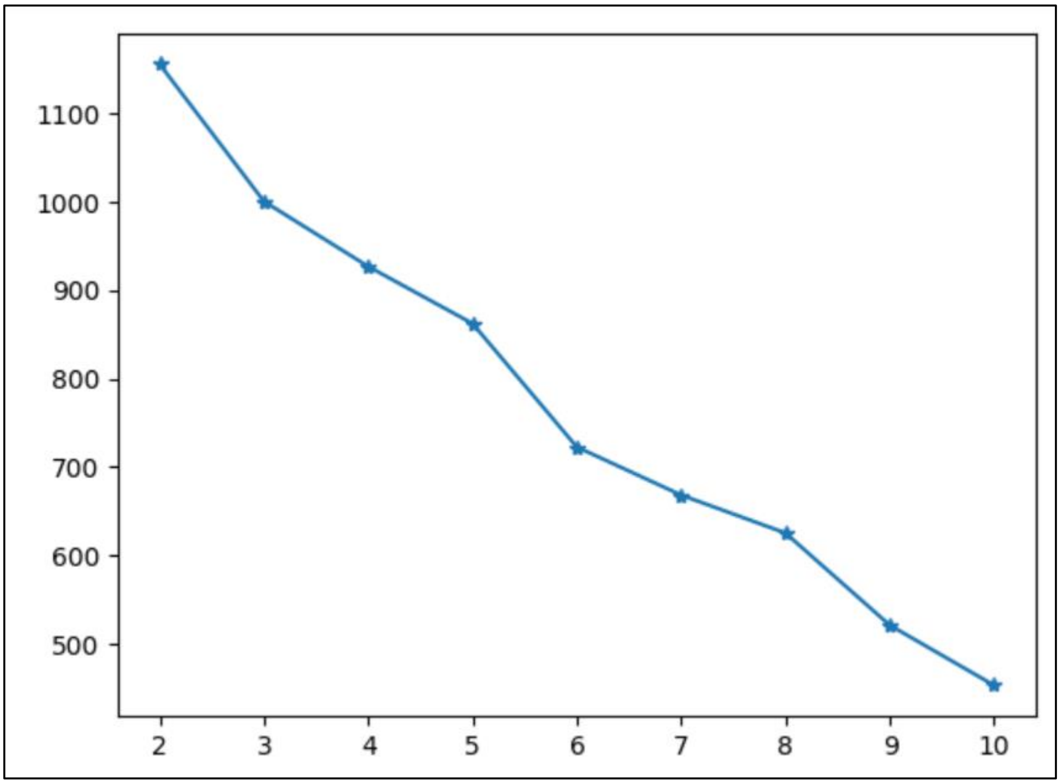
Data Preprocessing

Our final dataframe with all the indicators contain NaN (null) values. Further clustering cannot be performed on a dataset with null values. We will remove the rows (these are stock specific) which contain null values. After removing null values, we are left with 104/200 stocks.

Now, we create a new dataframe with just numerical values, indexed at the respective symbol. We remove the company name, industry, series, and ISIN to just keep the indicators. Then, we standardize the values.

K-Means Clustering

K-means clustering divides a set of n observations into k clusters, with each observation assigned to the cluster whose mean (cluster center or centroid) is closest, thereby acting as a representative of that cluster. K-means clustering is a part of unsupervised machine learning learning algorithm. This clustering helps us understand our data in a unique way– by grouping things together into clusters. We choose our k value = 6.



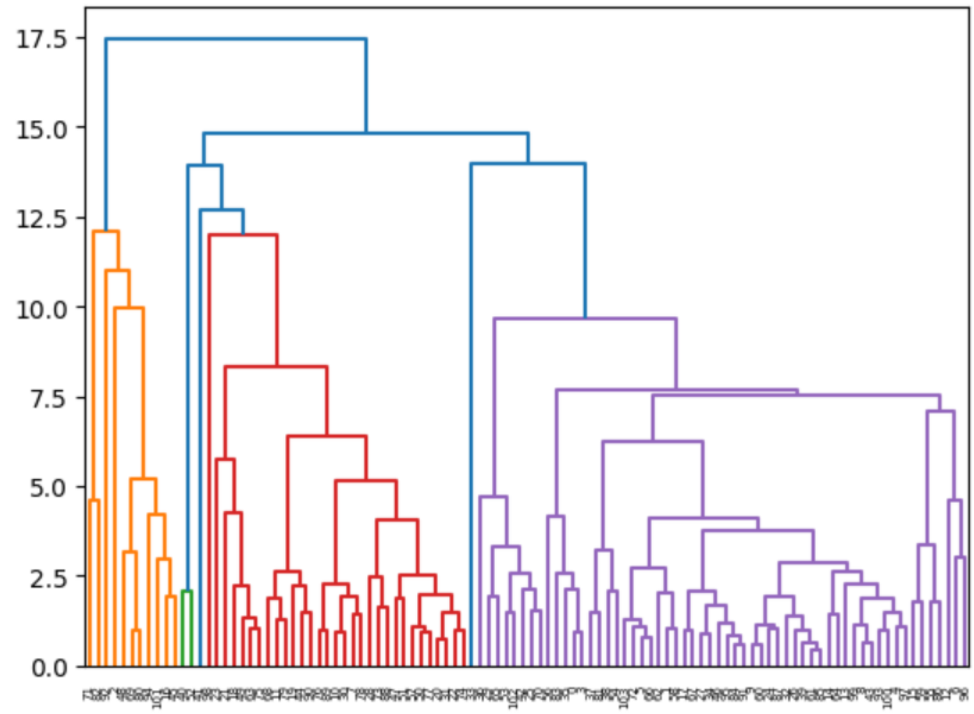
K=6 chosen from the Elbow Curve.

We then plot the stocks against different indicators cluster-wise, to find their profitability, risk, stability and growth.

Agglomerative Clustering

The agglomerative clustering is the most common type of hierarchical clustering used to group objects in clusters based on their similarity. It's also an unsupervised machine learning algorithm. The algorithm starts by treating each object as a singleton cluster. Next, pairs of clusters are successively merged until all clusters have been merged into one big cluster containing all objects. The result is a tree-based representation of the objects, named dendrogram.

```
In [266]:  dendrogram = sch.dendrogram(sch.linkage(X_scaled, method='ward'))
```



We are going to categorize the clusters as high growth, stable, and undervalued.

```
# Interpret clusters
for cluster in cluster_analysis.index:
    print(f"Cluster {cluster}:")
    metrics = cluster_analysis.loc[cluster]
    high_growth = metrics['revenue_growth']['mean'] > data['revenue_growth'].mean()
    stable = metrics['beta_value']['mean'] < 1 and metrics['debtToEquity']['mean'] < data['debtToEquity'].mean()
    undervalued = metrics['trailing_priceeq']['mean'] < data['trailing_priceeq'].mean()
    and metrics['priceToBook']['mean'] < data['priceToBook'].mean()
```

If the mean of revenue growth of a particular cluster is greater than the mean of the revenue growth of all stocks combined, then that cluster is classified as high growth.
If the mean of beta of that stock is less than 1 and the mean of debt-to-equity ratio of a particular cluster is greater than the mean of debt-to-equity of all stocks combined, then that cluster is classified as stable.
If the mean of trailing price-to-equity ratio of a particular cluster is greater than the mean of the same across all stocks combined, and the mean of price-to-book ratio of the cluster is greater than that of all stocks combined, then that cluster is classified as undervalued.

Selecting Stocks

The stocks selected as high growth and stable by agglomerative clustering is stored in one dataframe. This gives 93 stocks. Similarly, stocks performing appropriately as analysed by graphs after KMeans clustering is also stored in a dataframe (more details about it in the next section). This gives 88 stocks. Then, the stocks which are common to both the clustering method are selected. This gives us 77 stocks. These 77 stocks will be analyzed for portfolio optimization.

Portfolio Optimization

For portfolio optimization, we will run Monte Carlo simulation on the clusters of stocks- stocks accepted by both clustering methods, stocks rejected by the clustering methods, and stocks from the entire NIFTY 200 index.

5000 simulations of random allocations are generated to find the optimum allocation for each group of stock. The portfolio performance will analyzed based on the Sharpe ratio. The Sharpe ratio gives the return delivered per unit of risk taken. The code for the simulation is shown below.

```
In [19]:  sim = 5000
          all_weights = np.zeros((sim, len(symbol_list)))
          ret_arr = np.zeros(sim)
          vol_arr = np.zeros(sim)
          sharpe_arr = np.zeros(sim)

          for i in range(sim):
              weights = np.array(np.random.random(77))
              weights = weights/np.sum(weights)
              all_weights[i,:] = weights
              ret_arr[i] = np.sum(returns.mean()*250*weights)
              vol_arr[i] = np.sqrt(np.dot(weights.T,np.dot(returns.cov()*250,weights)))
              sharpe_arr[i] = (ret_arr[i] - 0.075)/vol_arr[i]

In [20]:  sharpe_arr.max()

Out[20]:  1.9188339785915278

In [21]:  sharpe_arr.argmax()

Out[21]:  4550

In [32]:  weights = all_weights[4550,:]
```

The weights are assigned to each stock based on the index at which the highest Sharpe Ratio was calculated. These stocks will have a greater weightage in our investment.

As we have 77 stocks initially (this is for the cluster selected by both types of clustering), we will select the best performing stocks each time and run a separate loop. In this case, in first case, all stocks have Sharpe Ratio > 0.01 are chosen (50 stocks). After running the simulation again, the stocks with Sharpe's ration > 0.02, which resulted in 21 stocks being chosen. For the third iteration, we run these 21 stocks and select the top 10 stocks. Finally, we run the Monte Carlo simulation to these 10 stocks to get their weight and according perform calculations on them accordingly.

A similar approach is applied to the rejects of clustering. Because they are just 27 in number, the top 10 stocks are selected from the 1st simulation. A final round of simulation is applied on these 10 stocks to rank them by their weight. A similar approach is applied to the stocks which represent the entire NIFTY 200.

An important observation- as the number of stocks reduced after each iteration for each cluster, the Sharpe ratio increased, and so did profit percentage. This proves that this optimization is headed towards the correct direction.

Portfolio Calculation

Now that we have the top 10 performing stocks from each segment, we determine their price as of 1st April 2020. We set our investment amount as Rs. 1,00,00,000 (1 crore). We figure out how many stocks we can buy from this amount- where more priority is given to stocks with a higher weight. We find their price as of 30th April, 2024, the day we are selling these stocks. The profit percentage is noted. Now, we buy these sets of stocks on 2nd May (1st working day of May) and sell them on 3rd June (1st working day of June) and record the profit percentage we get. Our end goal is to determine which set of stock earns the highest profit.

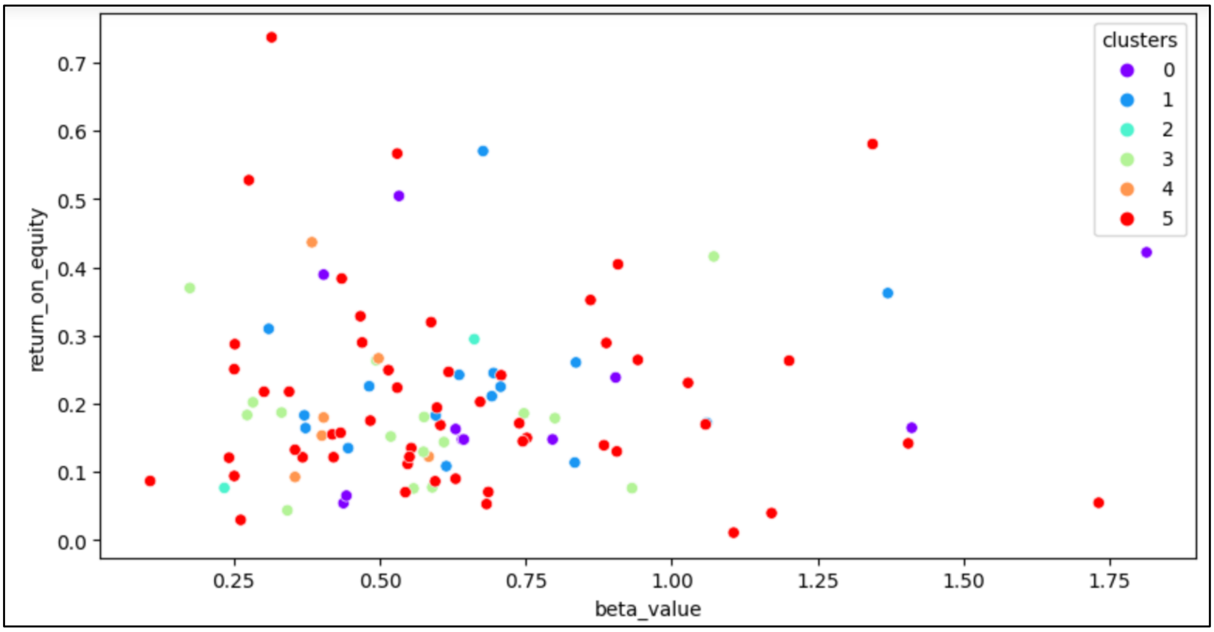
```
In [39]:
portfolio_value = 10000000
combined_df['Investment'] = combined_df['Weight'] * portfolio_value
combined_df['Shares'] = (combined_df['Investment'] / combined_df['Price_May2024']).astype(int)

# Calculate the value of each stock on 3rd June 2024
combined_df['Value_2024'] = combined_df['Shares'] * combined_df['Price_2024']

# Calculate total investment used and remaining cash
total_investment_used = (combined_df['Shares'] * combined_df['Price_May2024']).sum()
remaining_cash = portfolio_value - total_investment_used

# Calculate total portfolio value in June 2024 and profit/loss
total_value_2024 = combined_df['Value_2024'].sum()
profit_loss = total_value_2024 - total_investment_used
profit_percent = profit_loss / total_investment_used * 100
```

A piece of code which performs the same

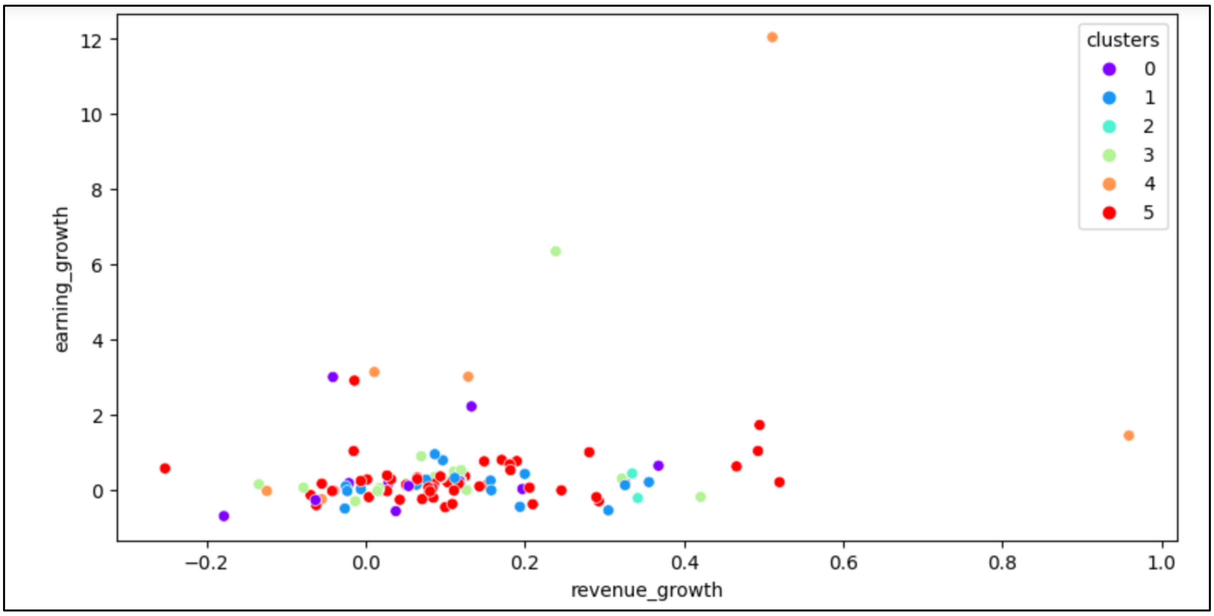
We confirm that the stocks selected by clustering produce the highest profit percentage.

5. Data Analysis and Results

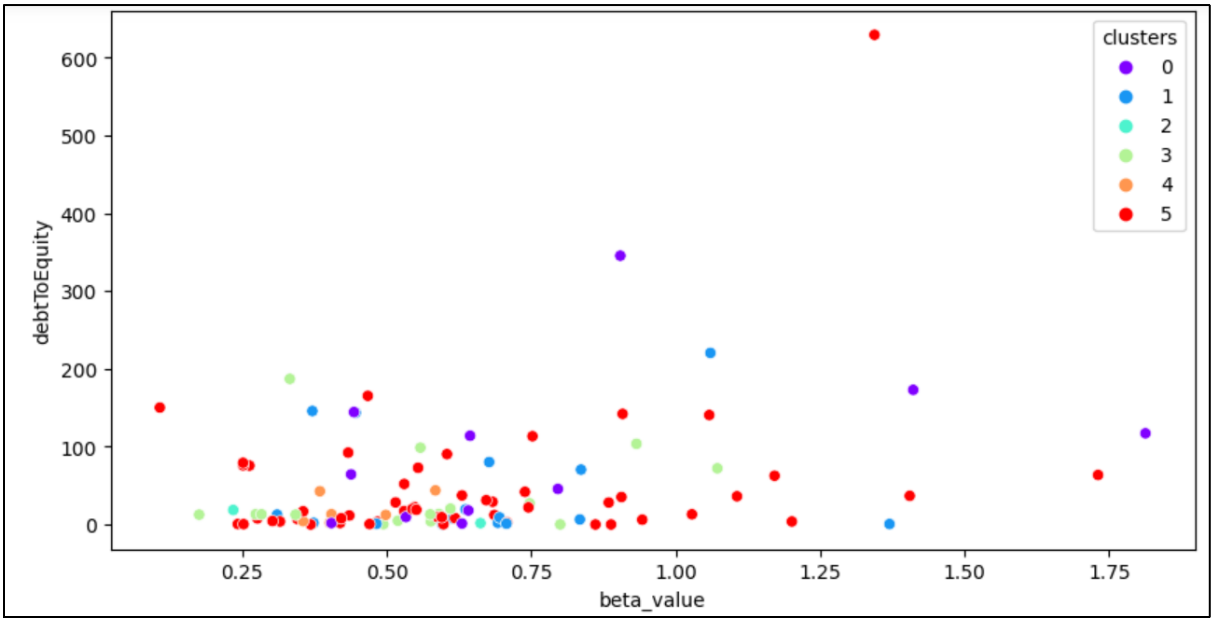
Understanding results from K-Means Clustering



Return on Equity and beta are plotted. All stocks belonging to clusters 0, 2 and 1 have beta value < 1, which signifies less volatility. Only one stock from cluster 1 has beta > 1 but it also has a higher return on equity. A bulk of the stock belongs to cluster 5, and most of them are less volatile, and most of them have commendable return on equity. Cluster 0 has most of its stocks with beta < 1 along with good ROE.



4 out of 6 stocks belonging to cluster 4 have higher earning growth than most of the stocks. Consistent earnings growth is often a sign of a healthy and potentially profitable company.



Most of the cluster 5 stocks have low beta and low debt-to-equity ratio. This is perfect for conservative investors seeking low risk. Some of the cluster 5 stocks also have high beta and higher debt-to-equity, which is suitable for aggressive investors looking for higher profit margin, but is willing to risk.

Some of the stocks under cluster 5 include- TTIN, PRESTIGE, OIL, NHPC, SUPREMEIND, and HAL, and all 6 of them make it to our list of optimized portfolio. The remaining 4 stocks come from clusters 1 and 4, each contributing 2 stocks. From all the other graphs plotted (not displayed here because they don't show much of a significant result + graphs changed after this because I re-ran the code), cluster 3 is rejected because they are very stagnant.

Understanding results from Agglomerative Clustering

```
Cluster 0:
  - Stable Stocks

Cluster 1:
  - Undervalued Stocks

Cluster 2:
  - High Growth Stocks

Cluster 3:
  - High Growth Stocks
```

From the 4 stocks obtained after agglomerative clustering, we choose to reject cluster 1 as they are undervalued.

He combine clusters 0, 2 and 3 and find intersection of it with stocks selected by K-Means.

Findings from Monte Carlo Simulation

1. **Stocks selected by K-Means and Agglomerative clustering-**

	Symbol	Weight	Price_April2020	Price_2024	Investment	Shares	Value_2024
0	CGPOWER	0.225950	5.257566	549.049988	2.259499e+06	429761	2.359603e+08
1	TRENT	0.200196	462.846893	4643.514160	2.001962e+06	4325	2.008320e+07
2	HAL	0.198356	243.561844	3947.199951	1.983557e+06	8143	3.214205e+07
3	OIL	0.096982	41.931381	420.299988	9.698213e+05	23128	9.720698e+06
4	SUPREMEIND	0.077691	801.269653	4794.834961	7.769087e+05	969	4.646195e+06
5	SJVN	0.069660	16.840117	137.149994	6.966033e+05	41365	5.673209e+06
6	BAJAJ-AUTO	0.040755	1839.106323	9037.202148	4.075540e+05	221	1.997222e+06
7	NHPC	0.038893	15.995337	98.000000	3.889348e+05	24315	2.382870e+06
8	PRESTIGE	0.038628	179.633514	1400.849976	3.862790e+05	2150	3.011827e+06
9	TIINDIA	0.012888	257.003326	3791.000000	1.288808e+05	501	1.899291e+06

These are the 10, best performing stocks, along with their prices, investment, and number of shares purchased. Over a span of 4 years, they produced a profit % of over 3000%. This produced a Sharpe's ratio > 3, which is a very good value (also higher than all the previous iterations)

```
Summary:
Total Investment Used: Rs.9997784.47
Remaining Cash: Rs.2215.53
Total Portfolio Value in June 2024: Rs.317516832.56
Total Profit: Rs.307519048.09
Total Profit %: 3075.87
```

Interestingly, CGPOWER, HAL, SUPREMEIND belong to Capital Goods industry and SVJN and NHPC belong to Power industry.

Now, these stocks are validated- we calculate how profitable they are for a month in 2024. This is how they perform now.

Out[95]:

	Symbol	Weight	Price_May2024	Price_2024	Investment	Shares	Value_2024
0	CGPOWER	0.225950	549.049988	686.250000	2.259499e+06	4115	2.823919e+06
1	TRENT	0.200196	4643.514160	4662.200195	2.001962e+06	431	2.009408e+06
2	HAL	0.198356	3947.199951	5273.649902	1.983557e+06	502	2.647372e+06
3	OIL	0.096982	420.299988	446.233337	9.698213e+05	2307	1.029460e+06
4	SUPREMEIND	0.077691	4794.834961	5547.382812	7.769087e+05	162	8.986760e+05
5	SJVN	0.069660	137.149994	143.149994	6.966033e+05	5079	7.270588e+05
6	BAJAJ-AUTO	0.040755	9037.202148	9260.190430	4.075540e+05	45	4.167086e+05
7	NHPC	0.038893	98.000000	113.150002	3.889348e+05	3968	4.489792e+05
8	PRESTIGE	0.038628	1400.849976	1735.199951	3.862790e+05	275	4.771800e+05
9	TIINDIA	0.012888	3791.000000	3835.300049	1.288808e+05	33	1.265649e+05

```
Summary:
Total Investment Used: Rs.9991044.67
Remaining Cash: Rs.8955.33
Total Portfolio Value in June 2024: Rs.11605327.09
Total Profit: Rs.1614282.42
Total Profit %: 16.16
```

16% return for a span of a month is quite a good performance. We will now compare this result with the other sets.

2. Stocks rejected by Clustering

Out[29]:

	Symbol	Weight	Price_April2020	Price_2024	Investment	Shares	Value_2024
0	IOC	0.209647	37.269249	166.441727	2.096470e+06	56251	9.362514e+06
1	NTPC	0.205159	63.053024	369.049988	2.051590e+06	32537	1.200778e+07
2	BDL	0.199952	86.610512	983.625000	1.999519e+06	23086	2.270797e+07
3	ONGC	0.153618	52.517639	282.799988	1.536180e+06	29250	8.271900e+06
4	BHEL	0.110070	20.486383	292.700012	1.100702e+06	53728	1.572619e+07
5	RELIANCE	0.037544	973.840149	2933.100098	3.754407e+05	385	1.129244e+06
6	JINDALSTEL	0.033662	69.553703	941.849976	3.366218e+05	4839	4.557612e+06
7	VOLTAS	0.021458	459.112823	1476.561401	2.145842e+05	467	6.895542e+05
8	POWERGRID	0.011262	68.530640	313.600006	1.126248e+05	1643	5.152448e+05

```
Summary:
Total Investment Used: Rs.9822812.76
Remaining Cash: Rs.177187.24
Total Portfolio Value in June 2024: Rs.74968000.22
Total Profit: Rs.65145187.46
Total Profit %: 663.20
```

In 4 years, the rejected stocks produced a profit % of around 660%, which is almost 4.5 times lesser than that produced by stocks selected by clustering. This proves that our clustering produced good result.

```
Summary:
Total Investment Used: Rs.9997365.67
Remaining Cash: Rs.2634.33
Total Portfolio Value in June 2024: Rs.11532968.14
Total Profit: Rs.1535602.47
Total Profit %: 15.36
```

The profit % produced for May ’24 – June ’24 is 15.36%, which is lesser than what was generated for clustered stocks.

6. Conclusion

It can be concluded that the stocks chosen by K-Means and Agglomerative Clustering, further optimized Monte Carlo Simulation, provides us with 10 stocks which have performed exceptionally well in the past few years. Stock markets ae extremely unpredictable, no one knows in which direction the market will move. The past 4 years have thrown us a lot of uncertainties – COVID-19 pandemic, 2 wars – Ukraine-Russia and Israel-Palestine, and the General Assembly election this year. In spite of all these, every stock has shown some sort of increment over the past 4 years. Stock markets, have, and will always continue to be the best form of investment, if done thoughtfully and intelligently. It is also very easy to lose money in stock market, specially in day trading and short-term trading, if one is not careful enough. But over a period of time, investment in stocks yield a good profit percentage.

7. APPENDICES

All the code files are uploaded here- https://github.com/ayanika02/ISI-Project

References-

1. https://www.geeksforgeeks.org/scraping-wikipedia-table-with-pandas-using-read_html/
2. https://www.tutorialspoint.com/how-to-compare-two-lists-in-python
3. https://www.freecodecamp.org/news/dataframe-to-csv-how-to-save-pandas-dataframes-by-exporting/
4. https://www.investopedia.com/terms/p/price-to-bookratio.asp#:~:text=The%20P%2FB%20ratio%20has,stock%20may%20have%20been%20identified.
5. https://www.datanovia.com/en/lessons/agglomerative-hierarchical-clustering/
6. https://www.niftyindices.com/indices/equity/broad-based-indices/nifty-microcap-250-dummy/nifty-200
7. https://www.investopedia.com/ask/answers/difference-between-fundamental-and-technical-analysis/
8. https://cdn.preterhuman.net/texts/unsorted2/Stock%20books%20029/John%20J%20Murphy%20-%20Technical%20Analysis%20Of%20The%20Financial%20Markets.pdf
9. https://www.bajajbroking.in/blog/nifty-200-meaning-and-significance
10. https://www.linkedin.com/pulse/analyzing-3000-stocks-using-clustering-based-ratios-hamid/
11. https://www.investopedia.com/articles/active-trading/101014/basics-algorithmic-trading-concepts-and-examples.asp
12. https://www.investopedia.com/articles/investing/041013/differences-between-forward-pe-and-trailing-pe.asp#:~:text=Trailing%20P%2FE%20is%20a,reliable%20indicator%20of%20future%20performance.
13. https://builtin.com/data-science/portfolio-optimization-python
14. https://mishraayush447.medium.com/portfolio-optimization-using-python-b8d2b64e520e
15. https://www.youtube.com/watch?v=BfS2H1y6tzQ&ab_channel=Socratica
16. https://archive.is/20201024071659/https://towardsdatascience.com/clustering-analysis-on-stock-selection-2c2fd079b295
17. https://www.winvesta.in/blog/fundamental-analysis-a-complete-guide#:~:text=Fundamental%20analysis%20is%20a%20method,stock%20does%20not%20change%20overnight.
18. https://www.youtube.com/watch?v=H6du_pfuznE&ab_channel=KrishNaik