Portfolio Optimization and Back Testing Challenge

Introduction

In this report, a systematic approach to construct an optimal portfolio using a curated list of 10 stocks is presented. The strategy is driven by **momentum and quality** parameters followed by rigorous backtesting. Historical data of the selected stocks is collected over a time frame of 2 years from 1st March 2021 to 1st March 2023 and then rebalanced and backtested for over a period of another year till 1st March 2024.

To select assets for this momentum + quality-based portfolio, we have to begin by defining a **tracking universe**. A tracking universe is a predefined set or group of stocks from which our portfolio will be created. According to this strategy, it is pointless to track all the stocks in the market as it will lead to a highly inefficient decision-making process, so a tracking universe is important as it acts as our viewpoint of the stock market and all our decisions are limited to the stocks in this set.

A tracking universe should be based on criteria which we understand well. It can be any of the already established indices, or it can be based on market capitalisation, or stocks from a particular sector like tech, pharma, etc.

For a momentum-based portfolio, it is important to select a tracking universe which shows positive growth over a long period of time as theoretically, the stocks which are stable and are growing, will continue to do so in the future. It is also important to select a set which has assets from multiple sectors to encourage diversification and reduce portfolio risk. Considering these factors, an already established index can act as an excellent tracking universe for our portfolio

After comparing multiple top U.S Indices, I have selected **NASDAQ-100 (^NDX)** as our tracking universe as it showed a return rate of **100.29%** over 5 years (1st March 2018 to 1st March 2023) and it contains assets from diverse sectors such as industrial, technological, retail, biotech, healthcare, telecommunication, media and service companies.

Theory

Momentum is the rate of change of stock returns or the index. If the rate of change of returns is high, then the momentum is considered high; if the rate of change of returns is low, the momentum is considered low. In a momentum portfolio, a tracking universe is set up in which assets are ranked on the basis of momentum. This momentum of each stock is calculated by summing up its daily returns for a period of time (2 years in this report) and then the top 10 such stocks from our tracking universe are chosen in which our capitals are invested in. It is a type of growth portfolio where capital is invested in already well performing equities and works on the principle that the stocks which already show upward trend will continue to do so.

After investing in the top 10 stocks, a **rebalancing** process is done. Rebalancing involves periodically buying or selling the assets in the portfolio. Rebalancing is done after every set period of time such as annually, quarterly, monthly (or even daily for intraday momentum portfolio). The time period is chosen such as to minimize transaction costs. In the rebalancing process, the momentum of the stocks in the tracking universe are calculated again for the same time period including the new data available since the last rebalancing process, and then ranked and bought once again.

Momentum investing theory states that assets with strong past returns tend to continue performing well due to investor behaviour, institutional trends, and market inefficiencies. It relies on ranking securities based on past performance, selecting top performers, and rebalancing periodically while managing risks through diversification and exit strategies. Momentum strategies can be highly profitable but also come with risks. Pure momentum strategies can sometimes buy stocks with unsustainable growth (speculative rallies, assets which are overvalued) which might falter out later. To minimize such risks, we can carry out quality screening such as high ROE (Return on Equity), low D-E ratios (Debt to Equity), etc. Such a Momentum-Quality Portfolio removes weaker businesses which are prone to crash.

Momentum-Quality Investing will tend to perform better than Pure Momentum strategies almost always as the applied quality filters will help in selecting stocks which show genuine growth, making trends much more sustainable. This will also help to reduce turnover costs as genuine quality stocks are much likely to sustain trends.

Execution

Using the yfinance library, closing prices of all the currently listed NASDAQ-100 stocks were exported from Yahoo Finance. The 2-year return of each stock was calculated using the formula:

$$2\ year\ return\ (\%) = \frac{closing\ price\ (1st\ March\ 2023) - closing\ price\ (1st\ March\ 2021)}{closing\ price\ (1st\ March\ 2021)} \times 100$$

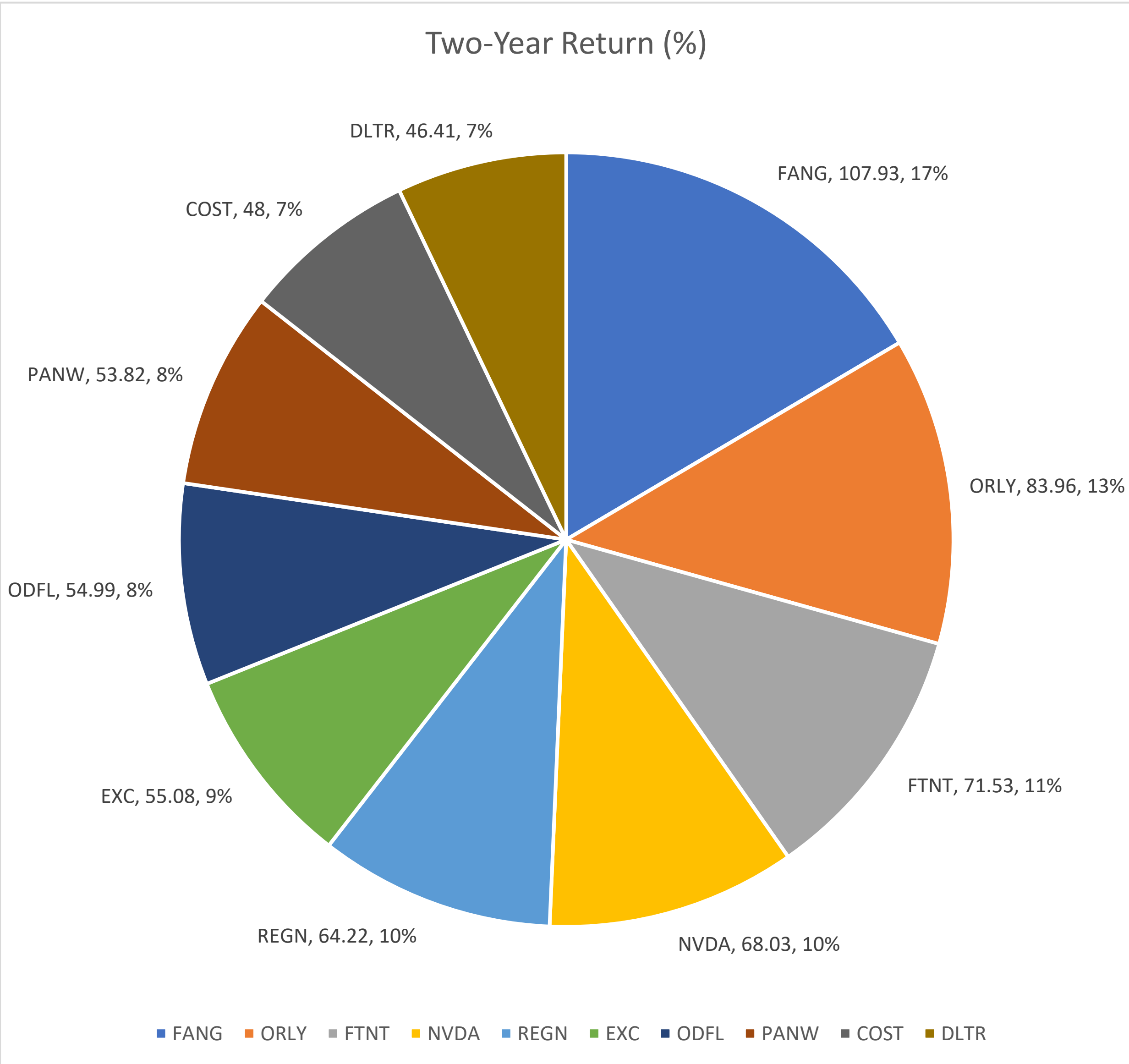
Note : Adjusted closing price is downloaded instead of closing price. Adjusted closing price adjusts for splits and dividends and/or capital gain distributions. This gives us a clean data and prevents any abnormalities while implementing our strategy.

```
import yfinance as yf
import pandas as pd
import numpy as np

# enables future-safe behaviour for downcasting
pd.set_option('future.no_silent_downcasting', True)

# list of NASDAQ100 stocks
tickers = ["AAPL", "MSFT", "NVDA", "GOOGL", "AMZN", "META", "TSLA", "AVGO", "PEP", "COST",
"ADBE", "NFLX", "AMD", "INTC", "QCOM", "TXN", "CSCO", "PYPL", "AMGN", "SBUX", "HON",
"ISRG", "GILD", "BKNG", "ADI", "REGN", "MDLZ", "MU", "VRTX", "ASML", "LRCX", "PANW",
"ABNB", "SNPS", "MRNA", "PDD", "CDNS", "ORLY", "MCHP", "DXCM", "ADP", "KLAC", "MAR",
"FTNT", "MNST", "ATVI", "KDP", "CTSH", "IDXX", "EXC", "ROST", "AEP", "ODFL", "WBD", "CTAS",
"BIIB", "CSX", "PCAR", "XEL", "PAYX", "VRSK", "EBAY", "CHTR", "DLTR", "MRVL", "TEAM",
"WDAY", "KHC", "ANSS", "EA", "FANG", "FAST", "ILMN", "LULU", "SIRI", "GFS", "DDOG", "ZS",
```

```
"OKTA", "CPRT", "DOCU", "NXPI", "MPWR", "BIDU", "TTD", "CDW", "CEG", "CRWD", "VRSN",
"ALGN", "SWKS", "SPLK", "MTCH", "ZM", "FOXA", "NTES", "SGEN", "PTON", "LCID"]

data = []

for ticker in tickers:
    try:
        # downloads stock data
        stock_data = yf.download(ticker, start="2021-03-01", end="2023-03-01",
auto_adjust=False)

        adj_close_2022 = stock_data.loc[: "2021-03-01"]["Adj Close"].iloc[-1]
        adj_close_2024 = stock_data.loc[: "2023-03-01"]["Adj Close"].iloc[-1]

        # calculates the two-year return
        two_year_return = round(((adj_close_2024 - adj_close_2022) / adj_close_2022) * 100,
2)

       # appends cleaned data
        data.append([ticker, adj_close_2022, adj_close_2024, two_year_return])

    except Exception as e:
        print(f"Error fetching data for {ticker}: {e}")

# creates the DataFrame
columns = ["Ticker", "Adj Close 2021-03-01", "Adj Close 2023-03-01", "Two-Year Return (%)"]
results_df = pd.DataFrame(data, columns=columns)

# sets ticker as index
results_df.set_index("Ticker", inplace=True)

# saves to Excel
excel_filename = "C:/Users/Adwaiy/Desktop/test_01.xlsx"
results_df.to_excel(excel_filename)

print(f"Data saved to {excel_filename}")
```

The data was exported to an excel sheet, the data was cleaned to format the redundant metadata and strings and then sorted in descending order to give back 10 stocks with the highest 2-year return:

Ticker	Adj Close 2021-03-01	Adj Close 2023-03-01	Two-Year Return (%)
FANG	60.90	126.63	107.93
ORLY	451.23	830.10	83.96
FTNT	34.65	59.44	71.53
NVDA	13.81	23.20	68.03
REGN	462.46	759.44	64.22
EXC	24.12	37.40	55.08
ODFL	108.36	167.94	54.99
PANW	61.23	94.18	53.82
COST	315.82	467.42	48.00
DLTR	99.23	145.28	46.41



Now to build a Momentum + Quality Portfolio, we must apply quality filters to the above found high momentum stocks. The three key quality metrics we'll use are:

1. **Return on Equity (ROE)** → Measures shareholder profitability

Return on equity is a measure of a company's financial performance. It is calculated by dividing net income by shareholders' equity. Because shareholders' equity is equal to a company's assets minus its debt, ROE is a way of showing a company's return on net assets. Whether an ROE is deemed good or bad will depend on what is normal among a stock's peers. For e.g., a normal ROE in the utility sector could be 10% or less. A technology or retail firm with smaller balance sheet accounts relative to net income may have normal ROE levels of 18% or more. Relatively high or low ROE ratios will vary significantly from one industry group or sector to another. However, a common shortcut for investors is to consider anything less than 10% a poor return on equity and anything near the long-term average of the S&P 500 acceptable; We will consider stocks with ROE only greater than 15%.

$$ROE\ (\%) = \frac{Net\ Income}{Shareholder's\ Equity} \times 100$$

2. **Return on Assets (ROA)** → Measures operational efficiency

Return on assets ratio is commonly expressed as a percentage using a company's net income and total assets. A higher ROA means a company is more efficient and productive at managing its balance sheet to generate profits. A lower ROA indicates there's room for improvement. Asset-heavy industries (e.g., manufacturing) tend to have lower ROA, making industry comparisons important. ROAs should always be compared among firms in the same sector, however, a ROA of over 5% is generally considered good.

$$ROA\ (\%) = \frac{Net\ Income}{Total\ Assets} \times 100$$

3. **Debt-to-Equity Ratio (D/E)** → Measures financial risk

The debt-to-equity (D/E) ratio is used to evaluate a company's financial leverage and is calculated by dividing a company's total liabilities by its shareholder equity. Higher D/E (>1.5) suggests heavy reliance on borrowed funds, which can be risky in economic downturns. Industry-specific benchmarks are essential.

$$D/E = \frac{Total\ Debt}{Stakeholders'\ Equity}$$

Including all these quality factors in our calculations, we add a few lines more for the required columns in our yfinance code above:

```
# fetch fundamental data
        stock = yf.Ticker(ticker)
        info = stock.info

        roe = info.get("returnOnEquity", np.nan) * 100 if info.get("returnOnEquity") else
np.nan
        roa = info.get("returnOnAssets", np.nan) * 100 if info.get("returnOnAssets") else
np.nan
        debt_to_equity = info.get("debtToEquity", np.nan)

# append cleaned data
        data.append([ticker, adj_close_2021, adj_close_2023, two_year_return, roe, roa,
debt_to_equity])
```

After factoring in the quality factors, cleaning up the metadata (redundant strings in the excel), and sorting the stocks in descending order of their 2-Year Return, we finally get:

Ticker	Adj Close 2021-03-01	Adj Close 2023-03-01	Two-Year Return (%)	ROE (%)	ROA (%)	D/E Ratio(%)
FANG	60.90	126.63	107.94	12.92	6.336	32.861
NVDA	13.81	23.20	68.03	119.177	57.417	12.946
REGN	462.46	759.44	64.22	15.951	7.314	10.012
ODFL	108.36	167.94	54.99	27.9	17.539	3.965
COST	315.82	467.42	48.00	32.894	8.735	31.431
SNPS	254.17	363.76	43.12	15.951	6.647	7.147
CPRT	27.46	35.23	28.32	19.185	12.133	1.434
MPWR	377.30	475.98	26.15	68.776	11.141	0.502
PAYX	83.11	103.99	25.13	44.254	11.643	20.981
MNST	43.78	50.88	16.23	21.274	14.775	7.295

So, the top 10 stocks we invest in initially are:

Diamondback Energy Inc., NVIDIA Corp., Regeneron Pharmaceuticals Inc., Old Dominion Freight Line Inc., Costco Wholesale Corp., Synopsys Inc., Copart Inc., Monolithic Power Systems Inc., Paychex Inc., Monster Beverage Corp.

Two allocate 10,000 dollars to 10 stocks, we can divide our total capital between them and allocate it equally, 1000 dollars to each. This is a good way of assigning values to the stock as this promotes diversification and reduces risk as we don't bias one stock over the other. This is known as an **equally weighted portfolio**.

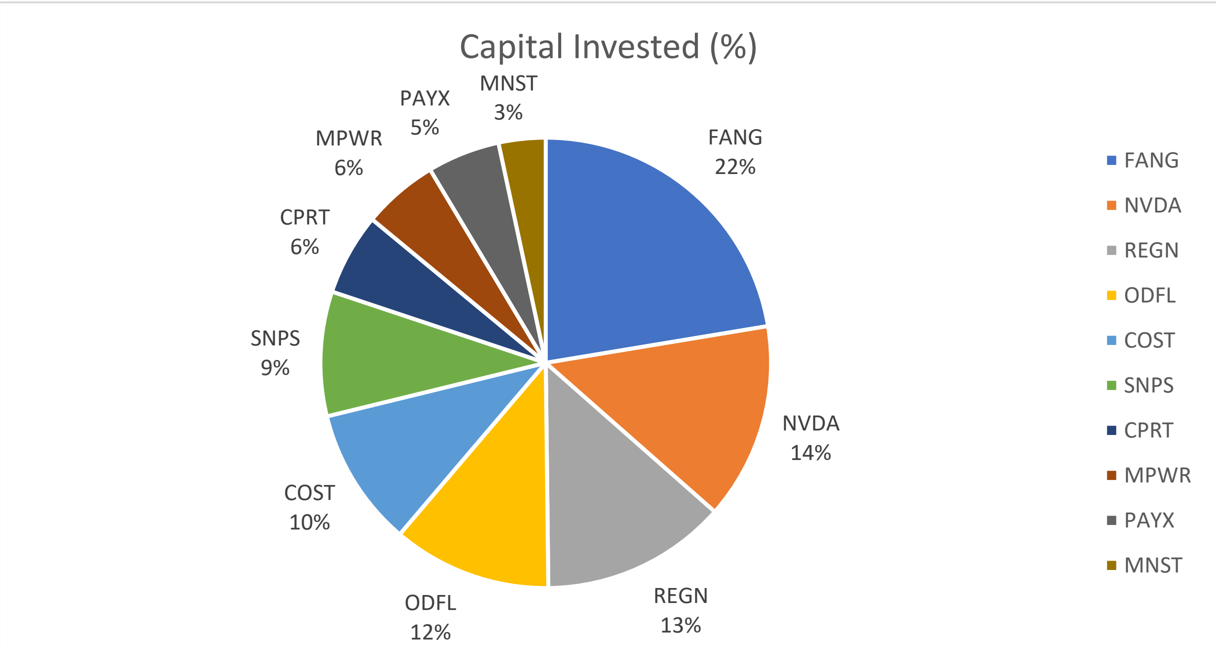
Or we could assign weight to each stock proportional to their 2-years return as this will ensure more profit being generated as we assign more value to stocks with higher return. This is also known as **skewed portfolio**. We assign weight using the formula:

$$Weight\ (stock\ x) = \frac{2\ year\ return\ (stock\ x)}{\sum_{i=1}^{10} 2\ year\ return\ (stock\ i)}$$

$$Allocation\ (stock\ x) = Total\ Capital\ \times Weight\ (stock\ x)$$

Using this method, we prioritize the Stocks with higher return and allocate more capital to them:

Ticker	Two-Year Return (%)	Weight	Capital Invested ($)
FANG	107.94	0.223881	2238.81
NVDA	68.03	0.141102	1411.02
REGN	64.22	0.133203	1332.03
ODFL	54.99	0.114053	1140.53
COST	48.00	0.099563	995.63
SNPS	43.12	0.089432	894.32
CPRT	28.32	0.058739	587.39
MPWR	26.15	0.054243	542.43
PAYX	25.13	0.052119	521.19
MNST	16.23	0.033665	336.65



Backtesting and Rebalancing

To evaluate our portfolio model, we will **backtest** the data quarterly (every 3 months), check the overall returns, rebalance the portfolio and then again reinvest all the capital and repeat the process 3 more times, so that we backtest the model over an entire year.

Backtesting is the general method to see how well a strategy or model behaves. It assesses the potential of a trading strategy by discovering how it would play out using historical data. If backtesting works. An ideal backtest chooses sample data from a relevant time period of a duration that reflects a variety of market conditions. In this way, one can better judge whether the results of the backtest represent a fluke or sound trading.

To backtest our weighted portfolio of 10 stocks over a historical period (3 months at a time), we will evaluate its performance using financial metrics like:

1. **Overall return**

Measures the total percentage gain or loss of an investment over a period. It shows how much the portfolio grew or shrank from start to end.

$$Overall\ Return = \frac{Capital_{final}}{Capital_{initial}} - 1$$

2. **Sharpe ratio**

Describes how much excess return you receive for each additional unit of risk you assume. A higher ratio implies a higher investment return compared to the amount of risk of the investment. Sharpe ratio of >1 is considered good.

$$Sharpe\ Ratio = \frac{R_p\ (Portfolio\ Return) - R_f\ (Risk\ Free\ Rate)}{\sigma_p\ (Volatility)}$$

3. **Maximum drawdown**

Represents the maximum observed loss from a peak to a trough of a portfolio, before a new peak is attained. Maximum drawdown is an indicator of downside risk over a specified time period.

$$Max\ DD = \frac{Trough\ Value - Peak\ Value}{Peak\ Value}$$

The code for backtesting is:

```
import yfinance as yf
import pandas as pd
import numpy as np

# tickers and respective weights are defined
tickers = "FANG", "NVDA", "REGN", "ODFL", "COST", "SNPS", "CPRT", "MPWR", "PAYX", "MNST"]
```

```python
weights = np.array([0.223881, 0.141102, 0.133203, 0.114053, 0.099563, 0.089432, 0.058739,
0.054243, 0.052119, 0.033665])  # Given weights


initial_investment = 10000
start_date = "2023-03-01"
end_date = "2023-06-01"

# downloads stock prices
data = yf.download(tickers, start=start_date, end=end_date, auto_adjust=False)["Adj Close"]

# calculates daily returns
returns = np.log(data / data.shift(1)).dropna()

# computes portfolio daily returns
portfolio_returns = returns.dot(weights)

# computes cumulative returns
cumulative_returns = np.exp(portfolio_returns.cumsum())

# computes overall return (final value - 1)
overall_return = cumulative_returns.iloc[-1] - 1

# annualized return
trading_days = len(portfolio_returns)
annual_return = cumulative_returns.iloc[-1] ** (252 / trading_days) - 1

# annualized volatility
annual_volatility = portfolio_returns.std() * np.sqrt(252)

# computes sharpe ratio (risk-free rate assumed 4% annualized)
risk_free_rate = 0.04
sharpe_ratio = (annual_return - risk_free_rate) / annual_volatility

# computes drawdown
cumulative_max = cumulative_returns.cummax()
drawdown = (cumulative_returns / cumulative_max) - 1
max_drawdown = drawdown.min()  # lowest drawdown value

# Display Results
print("Overall Return: {:.2%}".format(overall_return))
print("Sharpe Ratio: {:.2f}".format(sharpe_ratio))
print("Max Drawdown: {:.2%}".format(max_drawdown))
```

We get the following Results:

- Overall Return: 10.22%
- Sharpe Ratio: 2.31
- Max Drawdown: -4.11%

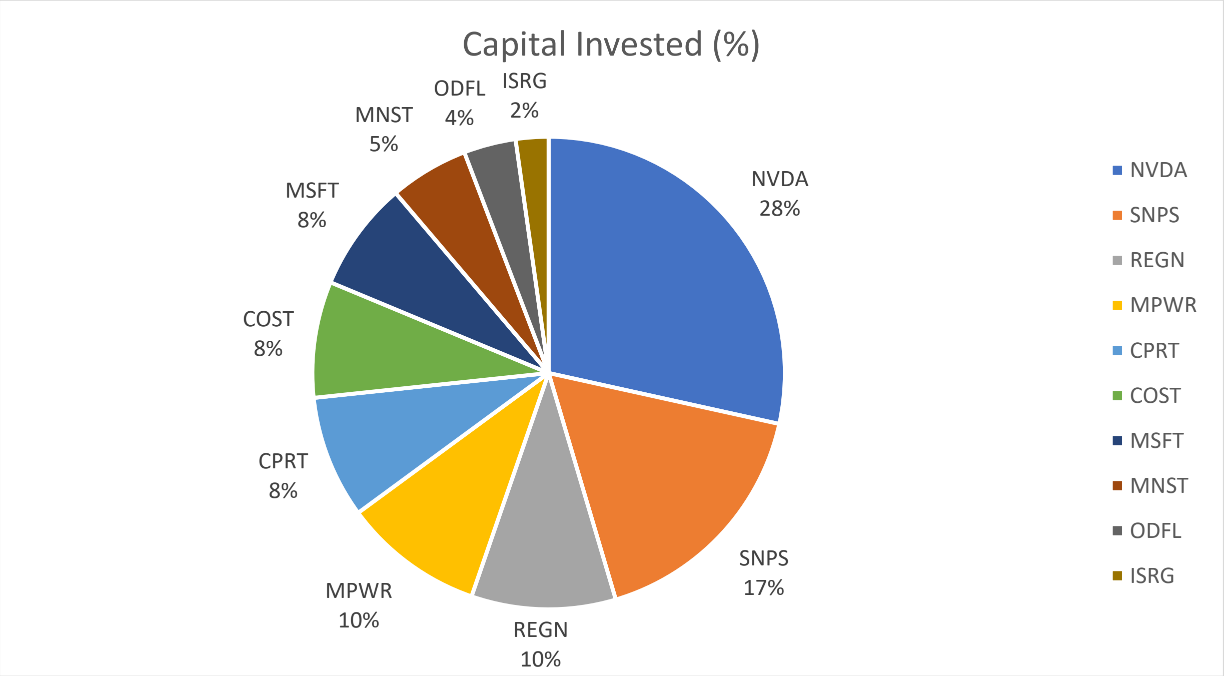
Final Capital can be calculated using :

$$Final\ capital = \ Initial\ Capital\ \times \left(1 - \frac{Overall\ Return(\%)}{100}\right)$$

Therefore, capital after 3 months will be **$11022**

After rebalancing the portfolio for the quarter 1st June 2023 to 1st September 2023 and assigning weights once again we get:

Ticker	Two-Year Return (%)	Weight	Capital Invested ($)
NVDA	133.00	0.284569	3136.52
SNPS	79.25	0.169558	1868.87
REGN	46.17	0.098783	1088.79
MPWR	45.03	0.096339	1061.85
CPRT	39.25	0.083988	925.71
COST	37.16	0.0795	876.24
MSFT	35.11	0.075132	828.10
MNST	25.27	0.054069	595.95
ODFL	16.67	0.035674	393.20
ISRG	10.46	0.022389	246.77



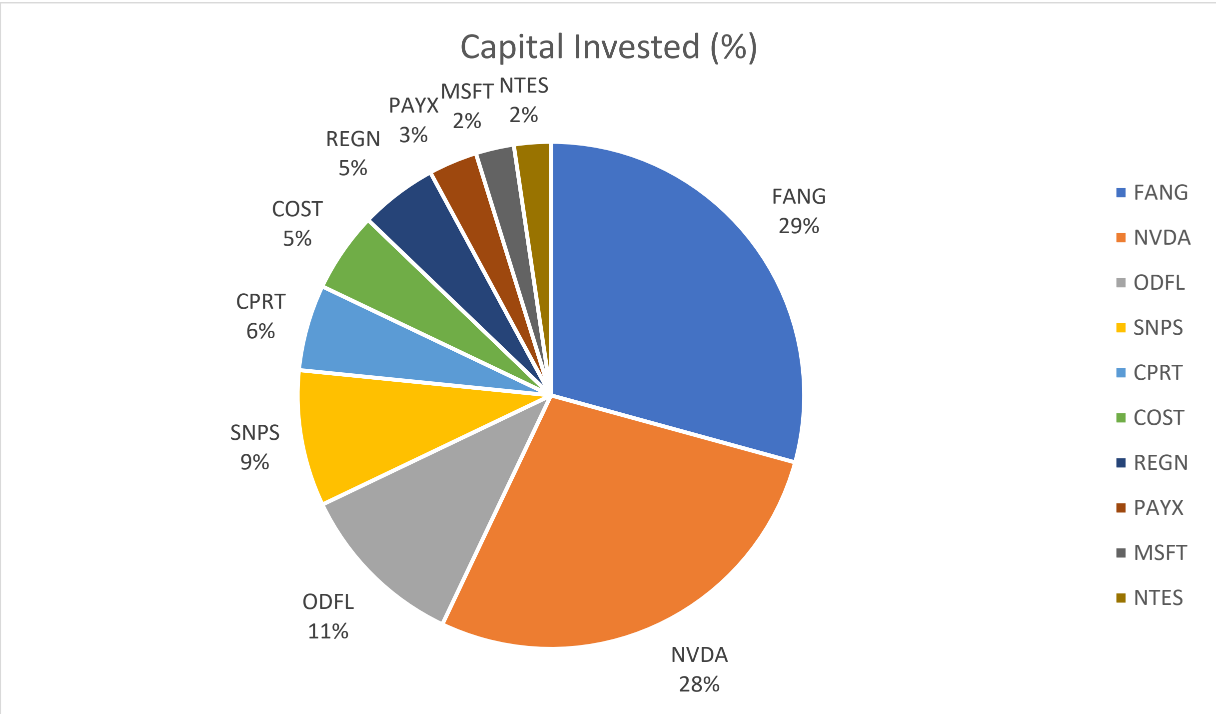
After backtesting this data for three months till 1st September 2024, we get the results:

- Overall Return: 6.28%
- Sharpe Ratio: 1.55
- Max Drawdown: -6.63%

Therefore, capital after 6 months will be **$11714.18**

After rebalancing the portfolio for the quarter 1st September 2023 to 1st December 2023 and assigning weights once again we get:

Ticker	Two-Year Return (%)	Weight	Capital Invested ($)
FANG	126.60	0.29	3428.13
NVDA	120.23	0.28	3255.45
ODFL	46.97	0.11	1271.97
SNPS	37.54	0.09	1016.63
CPRT	23.72	0.05	642.29
COST	22.01	0.05	596.11
REGN	21.37	0.05	578.69
PAYX	13.44	0.03	364.01
MSFT	10.56	0.02	285.90
NTES	10.16	0.02	275.11



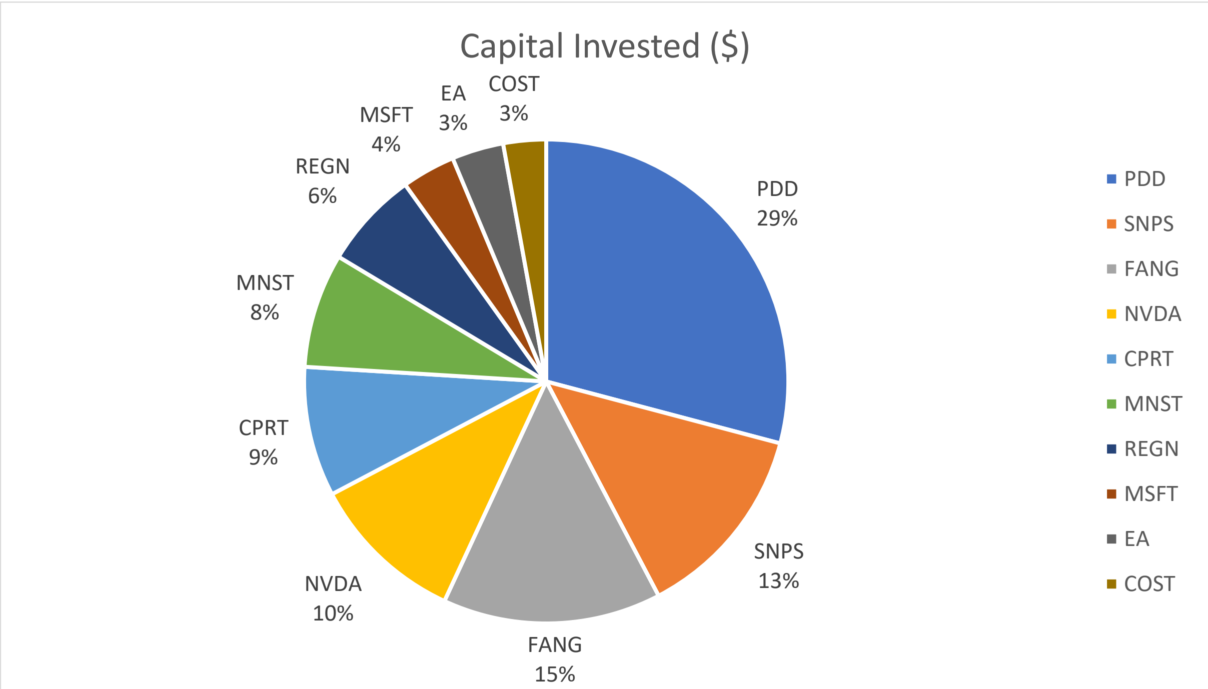
After backtesting this data for three months till 1st September 2023, we get the results:

- Overall Return: 7.43%
- Sharpe Ratio: 2.06
- Max Drawdown: -6.09%

Therefore, capital after 9 months is **$12584.54**

After rebalancing the portfolio for the quarter 1st December 2023 to 1st March 2024 and assigning weights once again we get:

Ticker	**Two-Year Return (%)**	**Weight**	**Capital Invested ($)**
PDD	137.39	0.291182	3664.39
SNPS	62.28	0.131998	1661.13
FANG	68.84	0.1459	1836.09
NVDA	48.98	0.103801	1306.29
CPRT	40.95	0.086789	1092.20
MNST	36.07	0.076453	962.13
REGN	30.64	0.064942	817.27
MSFT	16.90	0.035811	450.67
EA	16.31	0.034569	435.04
COST	13.47	0.028555	359.35



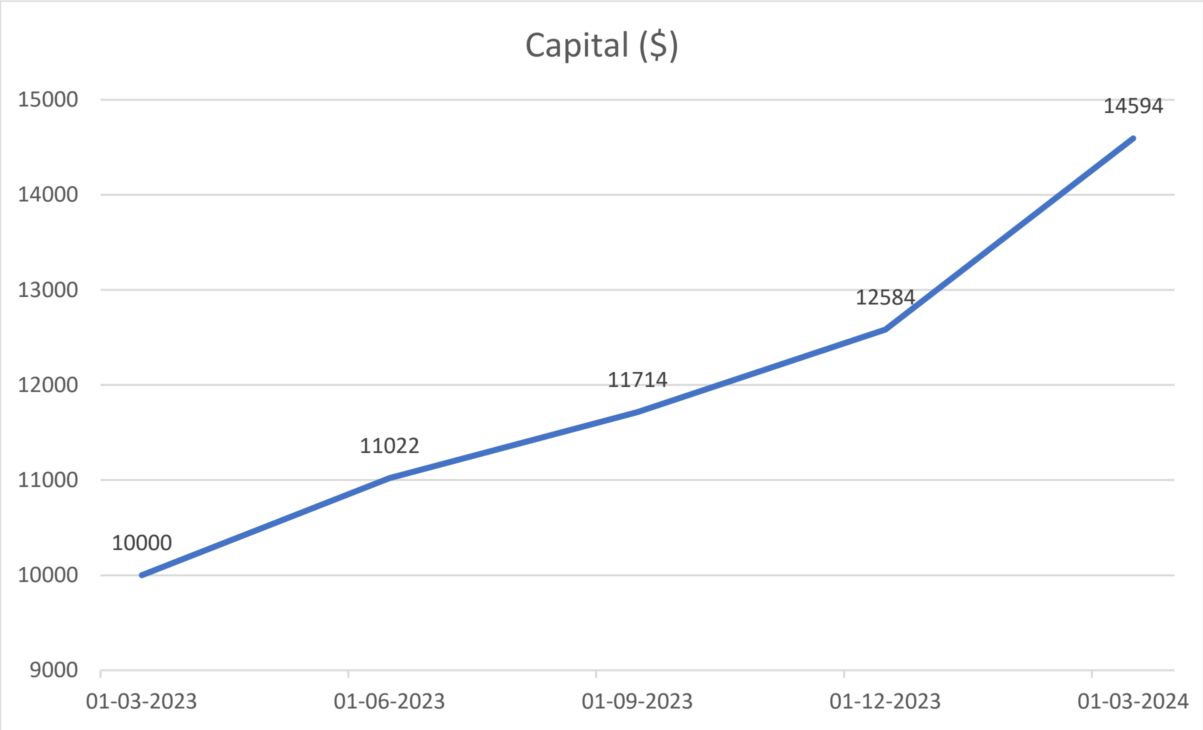
After backtesting this data for three months till 1st March 2024, we get the results:

- Overall Return: 15.97%
- Sharpe Ratio: - 7.21
- Max Drawdown: -1.84%

Therefore, capital after 1 Year is **$14594.29**

Conclusion

Capital Growth over one year is given by:



The total overall returns over the entire year were **45.94%**

Incurring transaction costs of investing and rebalancing (including commissions, slippage regulatory fees) at cost per trade = 1% , we get:

$$Total\ Transaction\ Cost = \sum Capital\ Invested\ \times Cost\ per\ trade$$

$$Total\ Transaction\ Cost = (10{,}000 + 11022 + 11714 + 12584)\ \times 1\%$$

Therefore, Total Transaction cost is **$453.20**

Therefore, Return after transaction cost = $14594.29 - $453.20 = **$14141.09 (41.41%)**

Disclaimer: The 41.41% return reflects strong performance under favourable market conditions and effective momentum-quality screening during the backtest, though actual results may vary with changing market dynamics.

Limitations & Takeaways

Momentum Investing tends to be riskier than other strategies even when incorporated with improvements such as quality factor as it is a growth strategy and heavily relies on the continuation of trends. If the momentum reverses, it is likely to result in substantial losses. Implementing the strategy for the year 2024 – 2025 when the market was not doing very well and all the stocks were crashing, resulted in a not-so-great outcome.

This strategy is more sensitive to market volatility. Price swings can disrupt trends and unpredictable results become inevitable. Sudden market corrections or shifts can rapidly result in growth decline

The frequent buying and selling that comes with momentum investing can lead to increased transaction costs. These costs eat into profits, particularly in markets with high trading fees or bid-ask spreads.

We should maintain a careful balance while choosing the frequency of rebalancing process so as to minimize net income reduction due to transaction costs.

It is important to stay in tune with the market and understand whether the rise in the prices of stock (the upward trend) is genuine or just useless noise without any underlying value. We can add more quality checks to consider while purchasing these stocks.

In general, a price-based momentum strategy, if implemented in the proper market cycle can give us great returns as demonstrated in this report, in fact, better more often than not, better than the market returns.

- Adwaiy Sakolkar

24IM10014