Quant Finance

Price Movement Strategy, Technical Indicators and momentum prediction using machine learning, Backtesting, Portfolio Optimization (Markowitz Portfolio theory)

Yash Adhiya - Presentation
HackRush' 23 IIT Gandhinagar

Collecting the Historical Price movement data

Library used: **yfinance**

Steps:

Importing the library yfinance
Download the file form NSE which contains all the listed stocks
Extract the Equity Name from that file
Pass the names in the yfinance to the function call and we are done

```
for name in equity_details.SYMBOL:
    try:
        data = yf.download(f'{name}.NS',start="2018-08-01", end="2022-08-30")
        data.to_csv(f'/content/drive/MyDrive/My Drive/StokesData_Hackthon/{name}.csv')
    except Exception as e:
        print(f'{name} ===> {e}')
```

Guess what ?

By this method I have downloaded 1800+ listed company's data for the further Analysis.

For the Portfolio Analysis, The stocks for the analysis are as follows :

```
['RELIANCE.NS', 'TCS.NS', 'INFY.NS', 'SBIN.NS', 'HDFCBANK.NS',
'HDFC.NS', 'TITAN.NS', 'HEROMOTOCO.NS', 'TATAMOTORS.NS',
'BPCL.NS']
```

My entire Analysis would contain these stocks as my portfolio, but yes for backtesting and robustness check I have uses more than 20 stocks, surely.

Momentum Prediction Using Machine Learning

1. **Exponential Moving Average**
   - EMA: Exponential Moving Average is a type of moving average that gives more weight to recent prices, and less weight to older prices. It is calculated by taking the average price of a security over a specified time period, with more weight given to the most recent data. The EMA is often used to identify trends in the market.

```
#calculation of exponential moving average
def EMA(df, n):
    EMA = pd.Series(df['Close'].ewm(span=n, min_periods=n).mean(), name='EMA_' + str(n))
    return EMA
```

2. **ROC: Rate of Change is a momentum oscillator**
   - Rate of Change is a momentum oscillator that measures the percentage change in price between the current price and the price n periods ago. It is calculated by taking the difference between the current price and the price n periods ago, and dividing that by the price n periods ago. The ROC is often used to identify overbought and oversold conditions.

```
#calculation of rate of change
def ROC(df, n):
    M = df.diff(n - 1)
    N = df.shift(n - 1)
    ROC = pd.Series(((M / N) * 100), name = 'ROC_' + str(n))
    return ROC
```

3. MOM: SImple Momentum

- Momentum is a simple indicator that measures the difference between the current price and the price n periods ago. It is calculated by subtracting the closing price n periods ago from the current closing price. The momentum indicator is often used to identify trends in the market.

```
#Calculation of price momentum
def MOM(df, n):
    MOM = pd.Series(df.diff(n), name='Momentum_' + str(n))
    return MOM
```

4. RSI: Relative Strength Index

is a momentum oscillator that measures the magnitude of recent price changes to evaluate overbought or oversold conditions. It is calculated by taking the average gain and loss over a specified time period and using that information to determine the strength of the price action. The RSI ranges from 0 to 100, with readings above 70 indicating overbought conditions and readings below 30 indicating oversold conditions.

```
#calculation of relative strength index
def RSI(series, period):
    delta = series.diff().dropna()
    u = delta * 0
    d = u.copy()
    u[delta > 0] = delta[delta > 0]
    d[delta < 0] = -delta[delta < 0]
    u[u.index[period-1]] = np.mean(u[:period]) #first value is sum of avg gains
    u = u.drop(u.index[:(period-1)])
    d[d.index[period-1]] = np.mean(d[:period]) #first value is sum of avg losses
    d = d.drop(d.index[:(period-1)])
    rs = u.ewm(com=period-1, adjust=False).mean()
    rd = d.ewm(com=period-1, adjust=False).mean()
    return 100 - 100 / (1 + rs / rd)
```

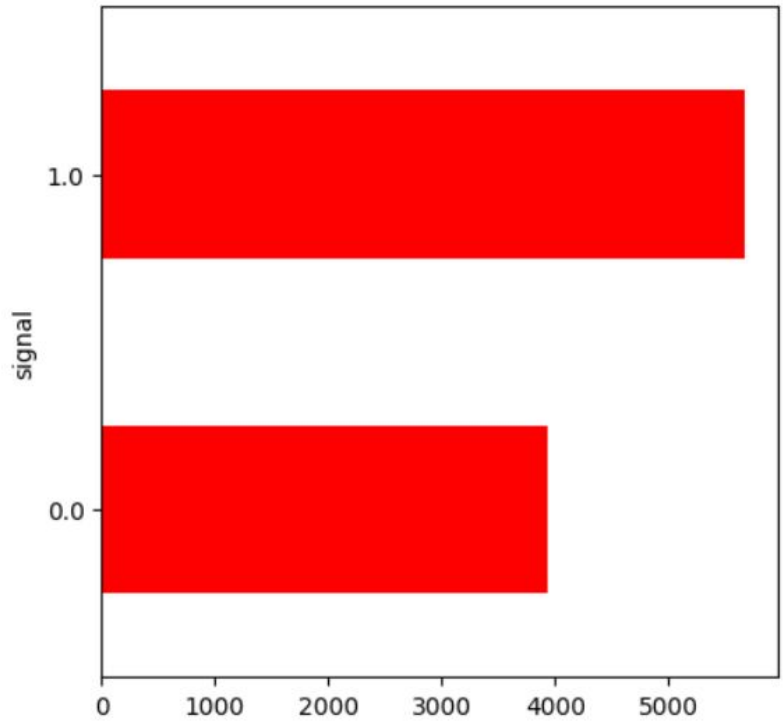
5**. STOK:** Stochastic Oscillator is a momentum indicator that compares the closing price of a security to its price range over a specified time period. It is calculated by taking the difference between the current closing price and the lowest low over a specified time period, and dividing that by the difference between the highest high and the lowest low over the same period. The Stochastic Oscillator ranges from 0 to 100, with readings above 80 indicating overbought conditions and readings below 20 indicating oversold conditions.

6. **STOD:** Stochastic Oscillator %D is a three-period moving average of the Stochastic Oscillator %K. It is used to smooth out the Stochastic Oscillator %K line, making it easier to read and interpret. The Stochastic Oscillator %D is often used in conjunction with the Stochastic Oscillator %K to confirm price movements and identify potential buy and sell signals.

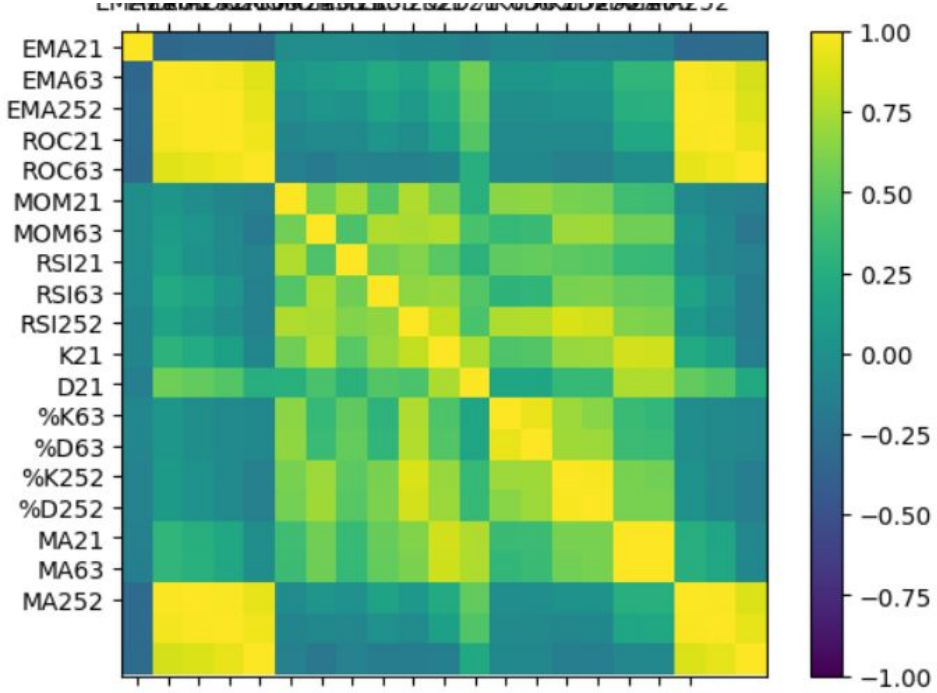
How am I creating the Signals ??

```
# Initialize the `signals` DataFrame with the `signal` column
datas['signal'] = 0.0

# Create short simple moving average over the short window
datas['short_mavg'] = datas['Close'].rolling(window=30, min_periods=1, center=False).mean()

# Create long simple moving average over the long window
datas['long_mavg'] = datas['Close'].rolling(window=120, min_periods=1, center=False).mean()

# Create signals
datas['signal'] = np.where(datas['short_mavg'] > datas['long_mavg'], 1.0, 0.0)
```

WOW ! I have prepared my target set for ML training



What Can we observe from this??



The Momentums are Correlated !!!

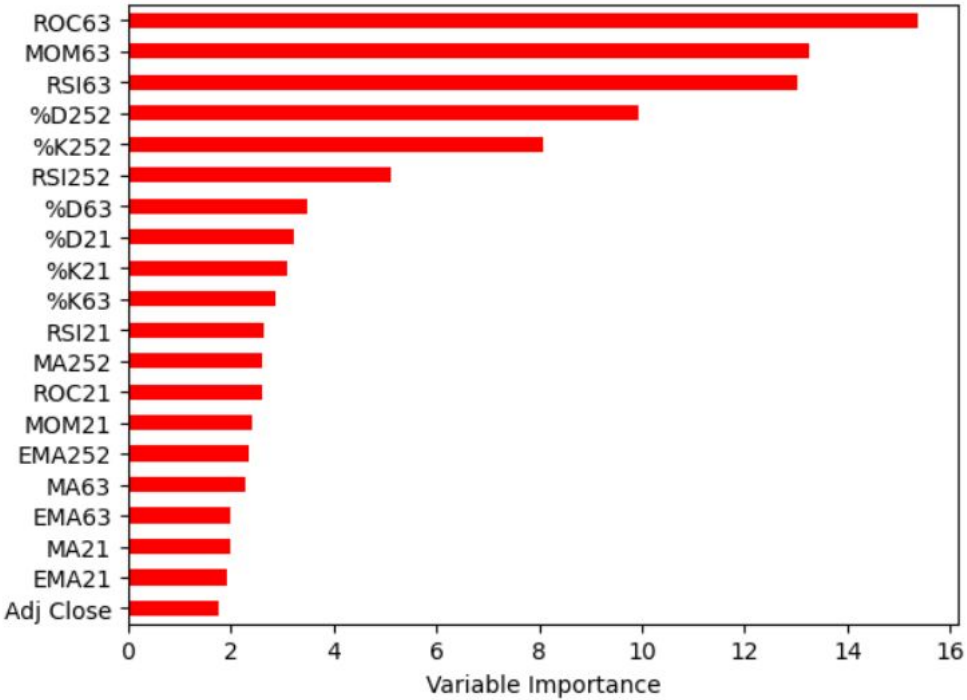
Which means what??

I can define a feature Importance of Random Forest Classifier and Rank Them?

Should I do that??

Nice, this project is driving me crazy

Result of Feature Importance VIsualization !



ANY GUESS WHAT NEXT ?

Yes! **ROC63** Momentum Oscillators has the strongest promise towards our signal!

SO WHAT a Trader can do?

Just Simply set ROC63 in his/her Trading Environment and GET the MONEY

YOU DON'T BELIEVE, WAIT !

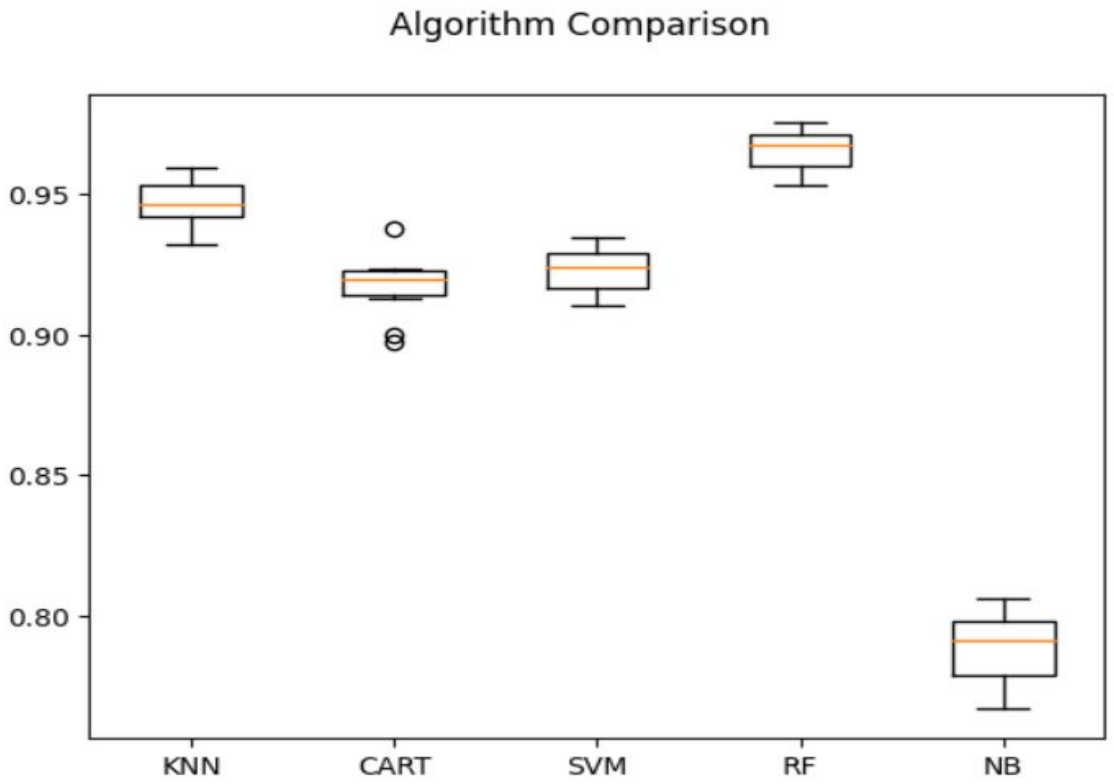
It’s Finally a time to let MACHINE LEARNING answer all our questions !

Modeling

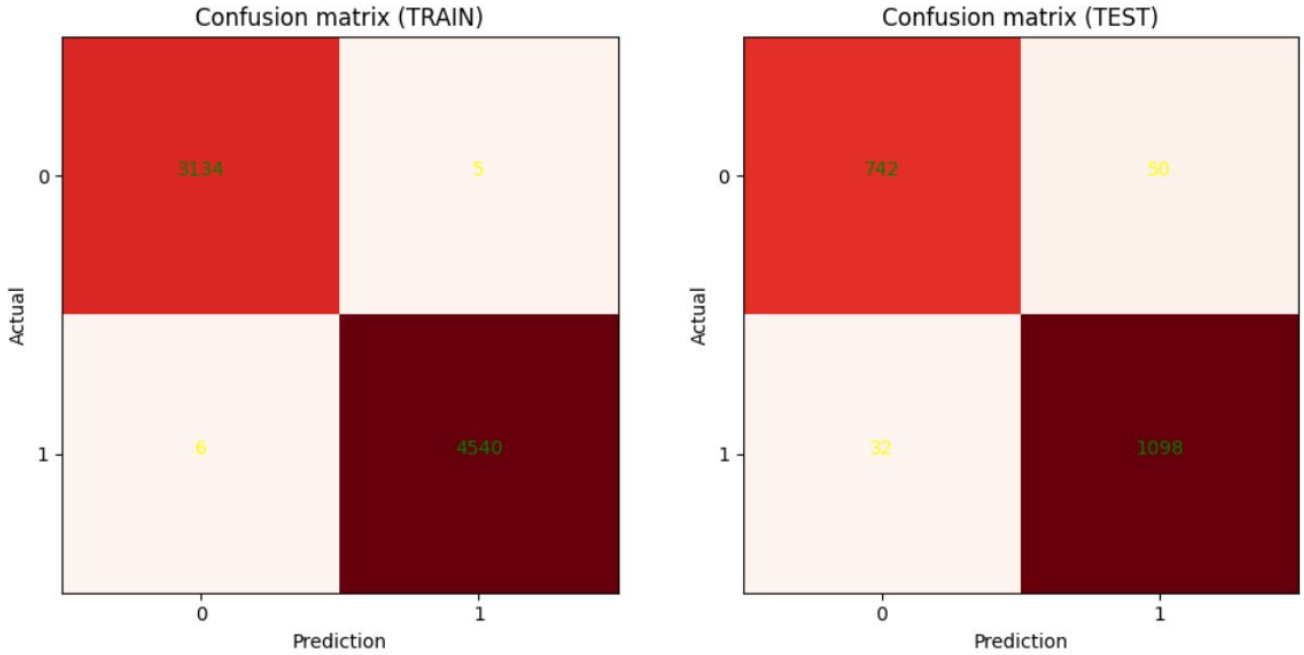
In order to know which algorithm technic is the best for our strategy, we evaluate 5 non linear different methods:

1. K-Nearest Neighbors (KNN)
2. Classification and Regression Trees (CART)
3. Support Vector Machines (SVM)
4. Random Forest (RF)
5. Gaussian Naïve Bayes (NB)

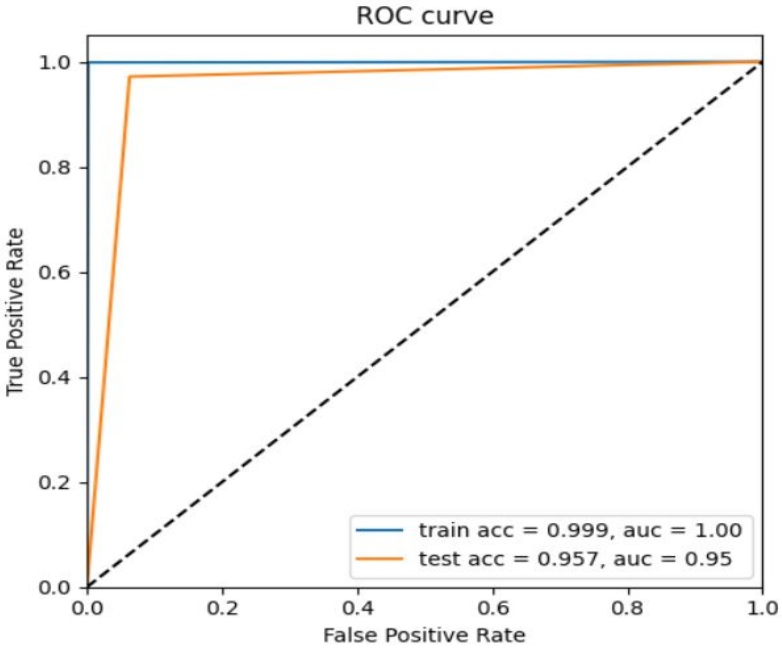
SEE, What Have I found ??



OKAY, Then Apply RF !



ALSO the AUC/ ROC curve, Please look it



What Inferences can be made?

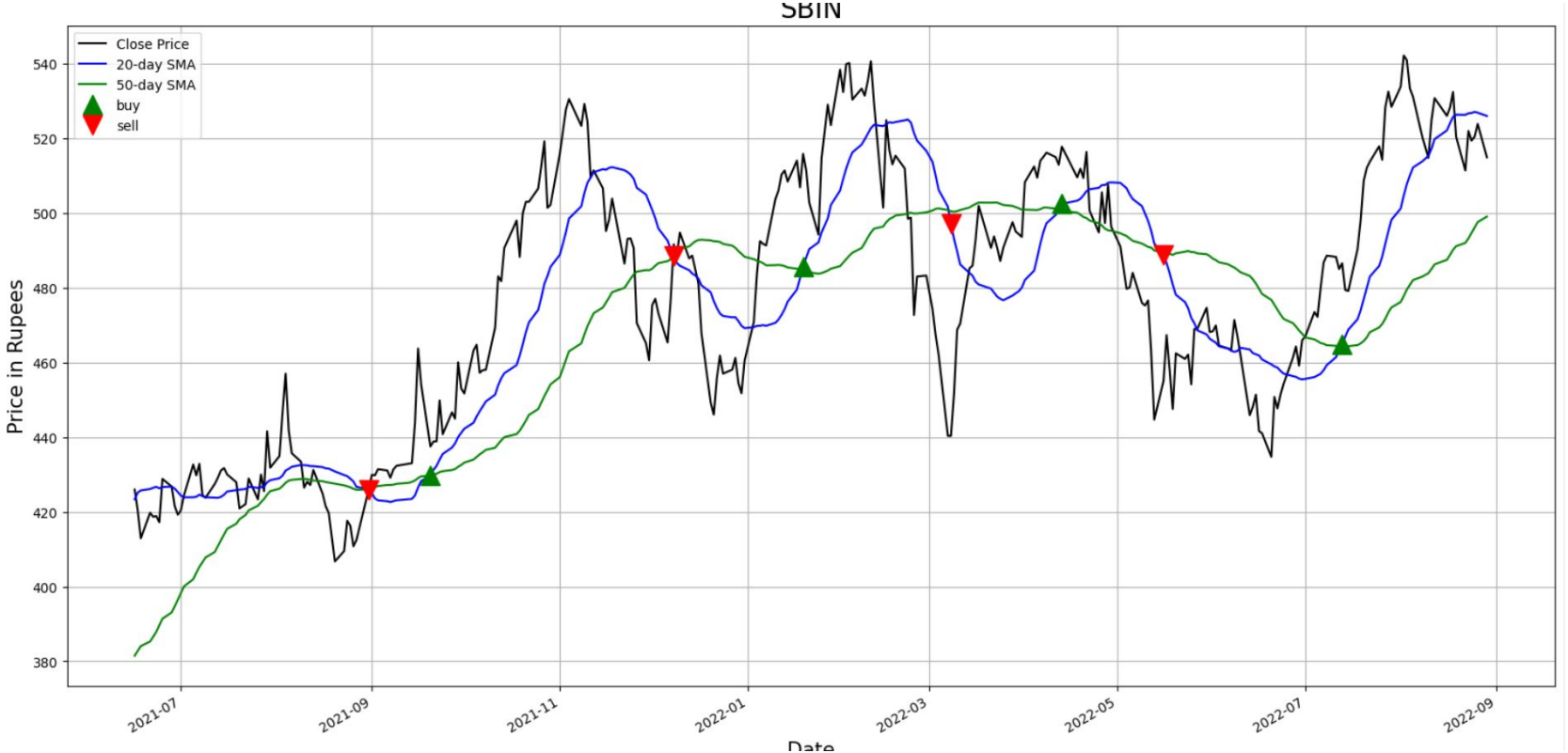
It means whichever Feature we are training after feature Importance, it gives almost 99% accuracy and this Momentum Oscillators are Trustworthy Believe !!

Machine Learning can’t lie!

Final Result

Based on the return from our strategy, we do not deviate that much from the actual market return. Indeed, the achieved momentum trading strategy made us well predict the stock prices direction to invest/disinvest in order to make profits. However, as our accuracy is not 100% therefore, we made relatively few losses compared to the actual returns.

MY own Strategy : Generating Signals(Golden CrossOver) and Backtesting



MY backtesting Results for 10 stocks

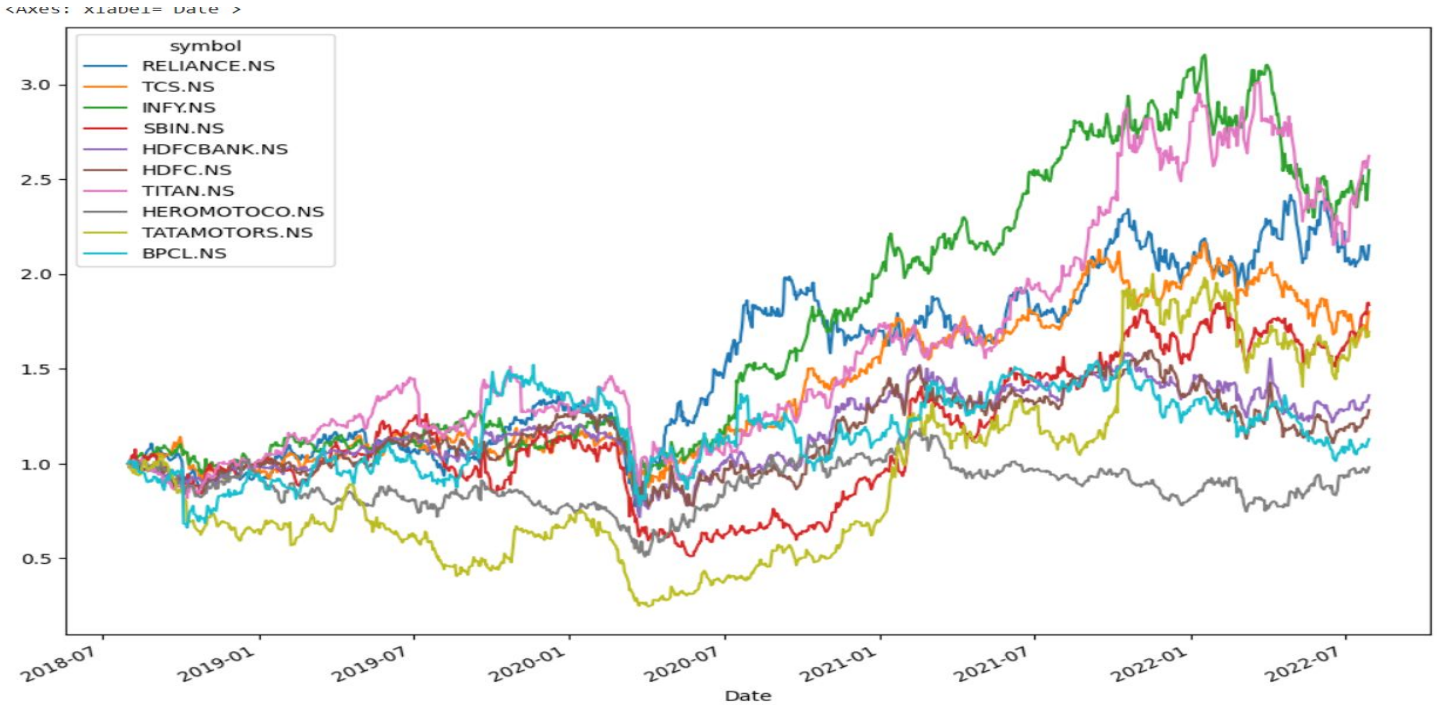
Return is 125% (Crazyyy) after portfolio optimization and selecting proper sharpe ratio

Start	2018-07-31 18:30:00+00:00
End	2022-07-28 18:30:00+00:00
Period	986 days 00:00:00
Start Value	100.0
End Value	225.648793
Total Return [%]	125.648793
Benchmark Return [%]	74.115425
Max Gross Exposure [%]	100.0
Total Fees Paid	1.165293
Max Drawdown [%]	35.810668
Max Drawdown Duration	154 days 00:00:00
Total Trades	245
Total Closed Trades	235
Total Open Trades	10
Open Trade PnL	78.293174
Win Rate [%]	74.893617
Best Trade [%]	152.151362
Worst Trade [%]	-43.044685
Avg Winning Trade [%]	36.835164
Avg Losing Trade [%]	-10.909201
Avg Winning Trade Duration	540 days 18:16:21.818181816
Avg Losing Trade Duration	280 days 05:41:41.694915256
Profit Factor	17.29915
Expectancy	0.201513
Sharpe Ratio	1.045095
Calmar Ratio	0.645627
Omega Ratio	1.206715
Sortino Ratio	1.456933

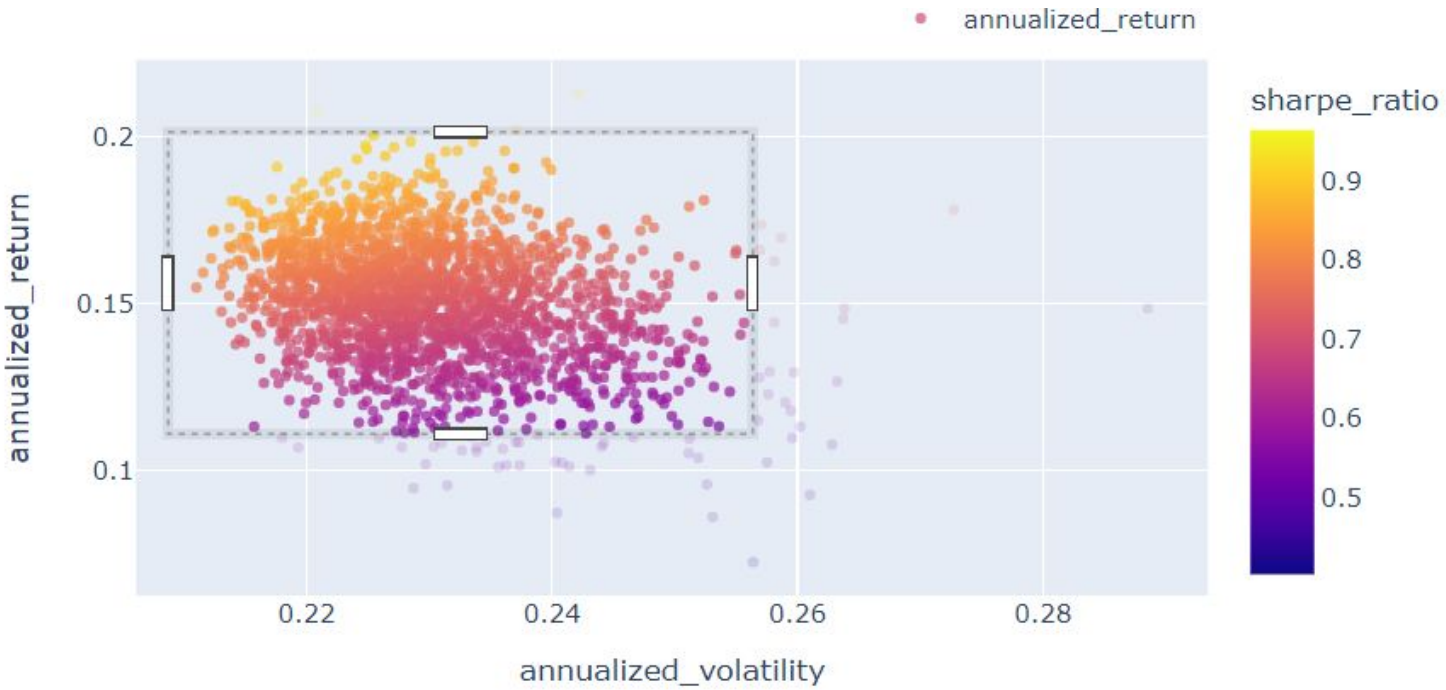
Name: 179, dtype: object

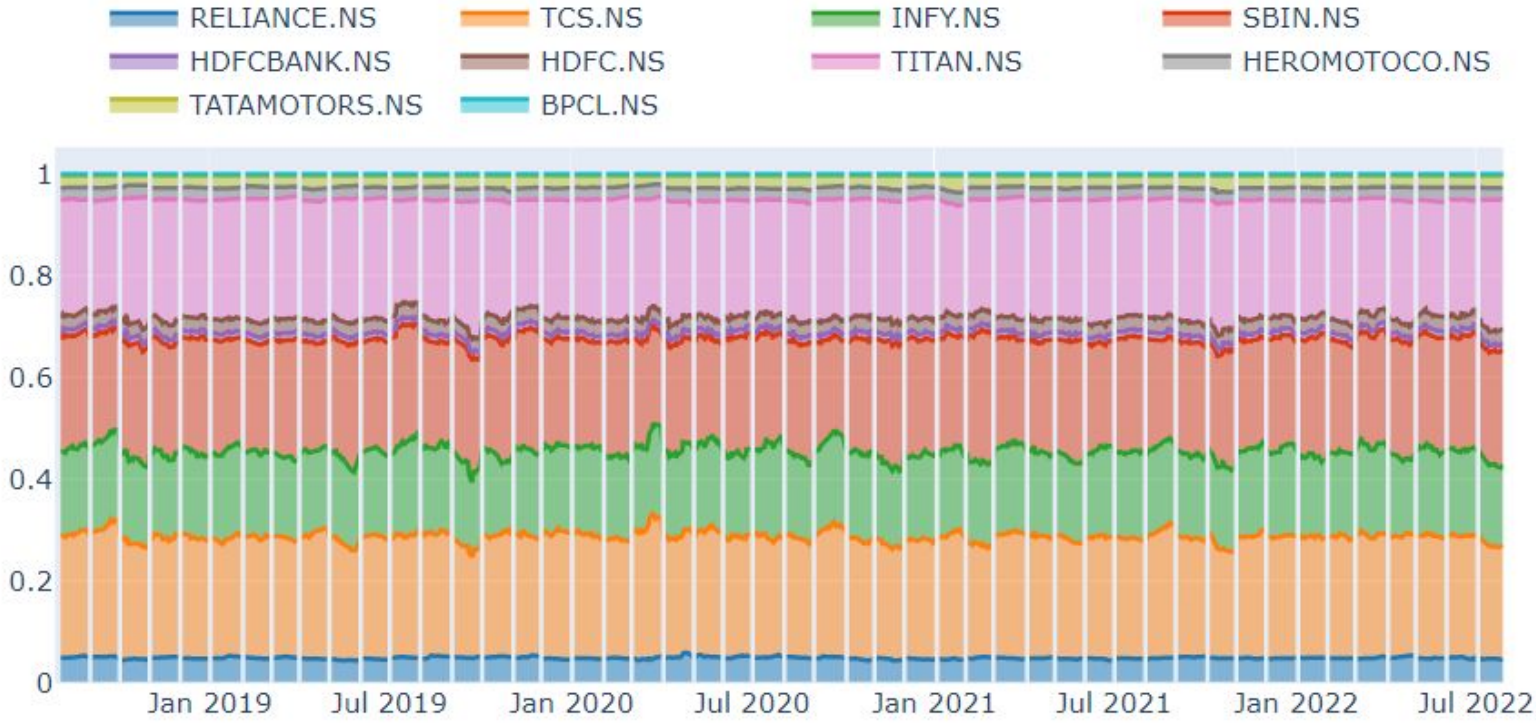
Portfolio Optimization - AGAIN THOSE 10 stocks

This is my portfolio! EQUITIES movement



RISK - Return Graph for all the weights (Markowitz's Theory)




RELIANCE.NS
TCS.NS
INFY.NS
SBIN.NS
HDFCBANK.NS
HDFC.NS
TITAN.NS
HEROMOTOCO.NS
TATAMOTORS.NS
BPCL.NS
1
0.8
0.6
0.4
0.2
0
Jan 2019
Jul 2019
Jan 2020
Jul 2020
Jan 2021
Jul 2021
Jan 2022
Jul 2022

Thank you !

I am looking forward to winning this, and also wants to further continue it to give it a research shape.