**Marcellus Internship Report**

*- Amitava Roy*

*Dated : 10 July, 2023*

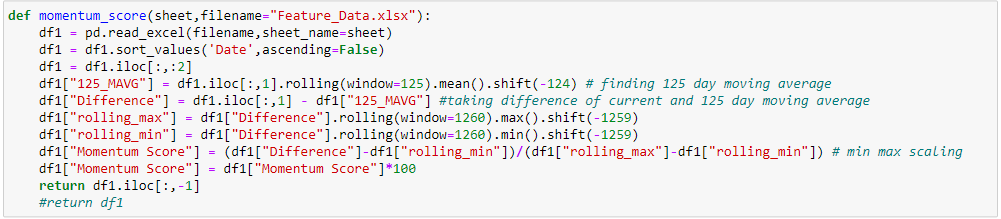
**Aim of the Project**:

The project aimed to develop a set of trading strategies on the S&P500 Index that can generate Buy, Hold and Sell Signals daily that can maximize CAGR and minimize drawdown. 18 years of daily data was used for the back testing of the project.

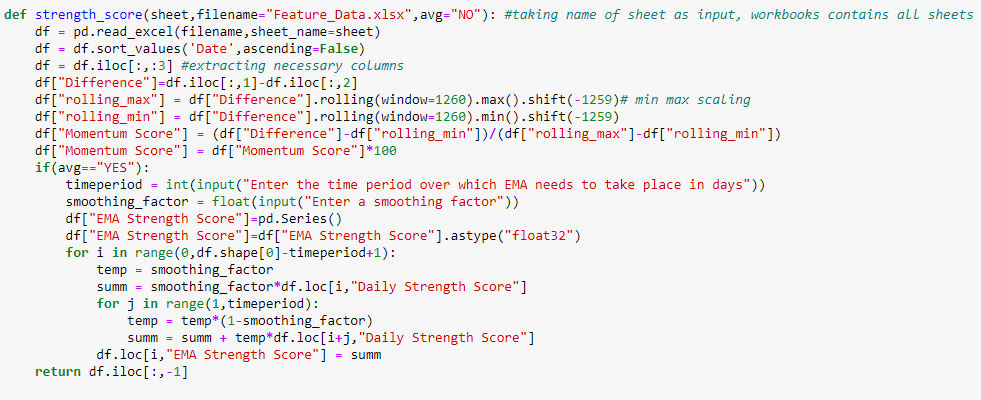
**Features Used**:

The project used 7 daily features which have been used by CNN for constructing their Fear and Greed Index namely:

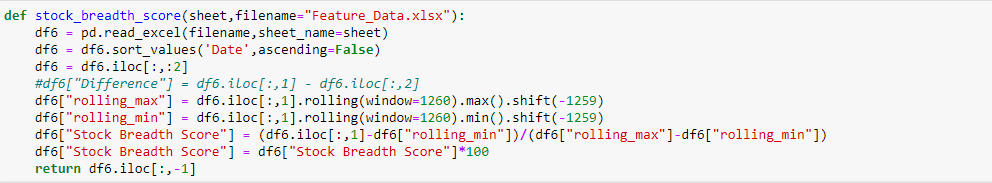
* *Market Momentum*: Difference between SPXT and its 125-Day moving average



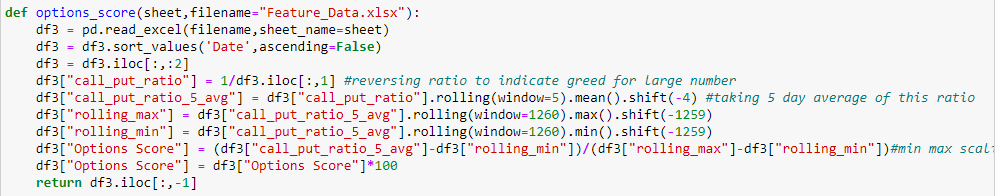
* *Stock Price Strength*: Difference between 52-week highs and 52-week lows



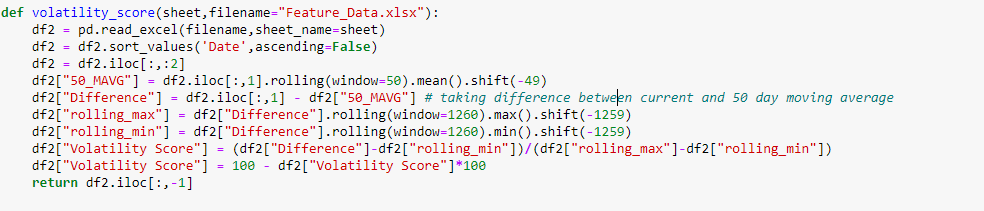
* *Stock Price Breadth*: McClellan Volume Summation Index



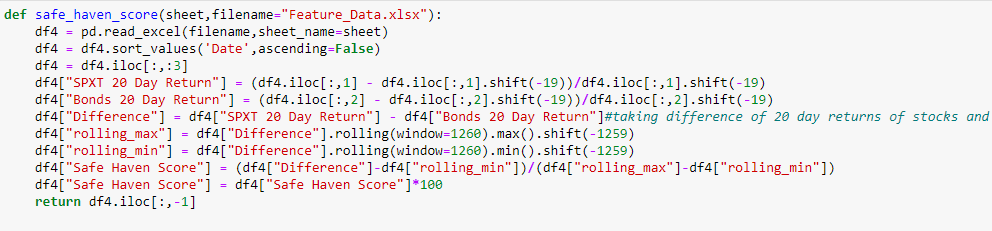
* *Put and Call Options*: 5-Day average of (Number of Call Options/Number of Put Options)



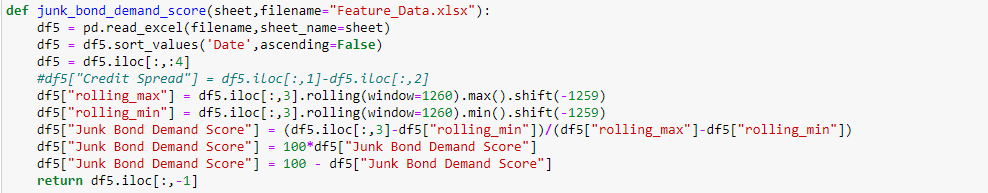
* *Market Volatility*: Difference between VIX and its 50-Day Moving average



* *Safe Haven Demand*: Difference in 20-Day Stock and Bond Returns

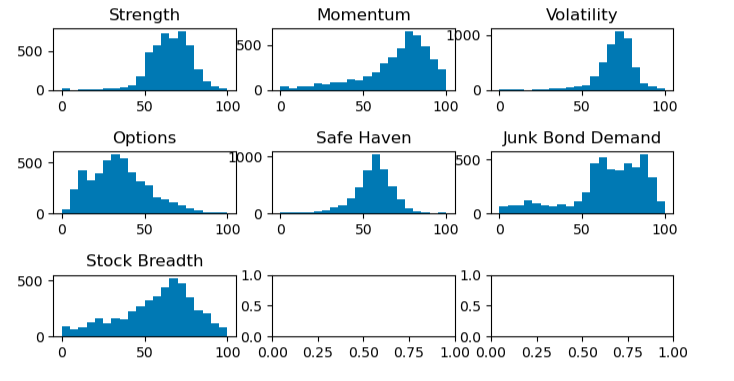


* *Junk Bond Demand*: Yield spread - junk bonds v investment grade



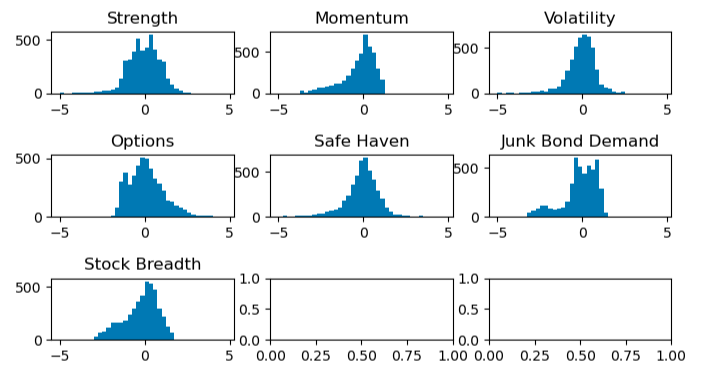
Note: *All these features have been normalized to a value between 0 and 100 using Min-Max Scaling. 5-year rolling minimum and maximum values were used for scaling the features. The data was extracted from Bloomberg Terminal (last 18 years data).Fear and Greed Index was calculated by taking an average of all these 7 features.*

**Distributions of all the 7 features:**

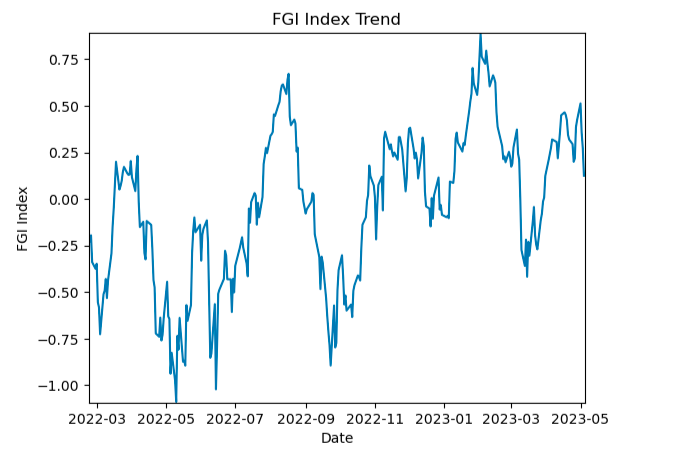


Note: *Since, all the features have different means and medians, the z-score of each feature was calculated for a better understanding*.

**Distributions of Z-Scores:**

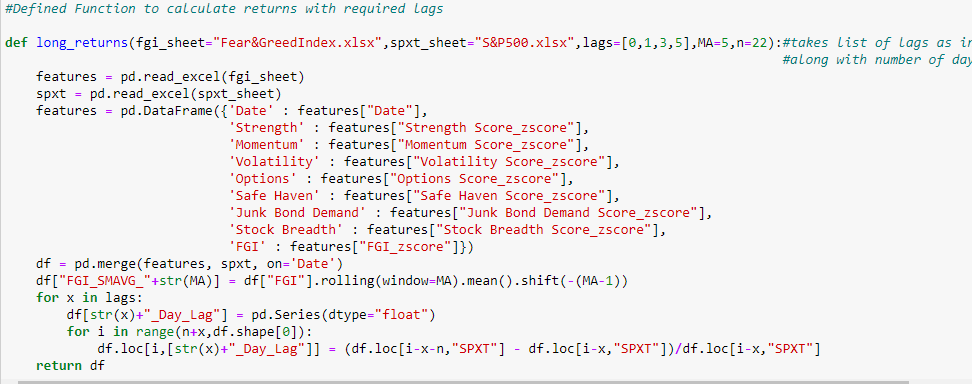


**Plotting the Trend of FGI\_zscore for the last 300 days:**



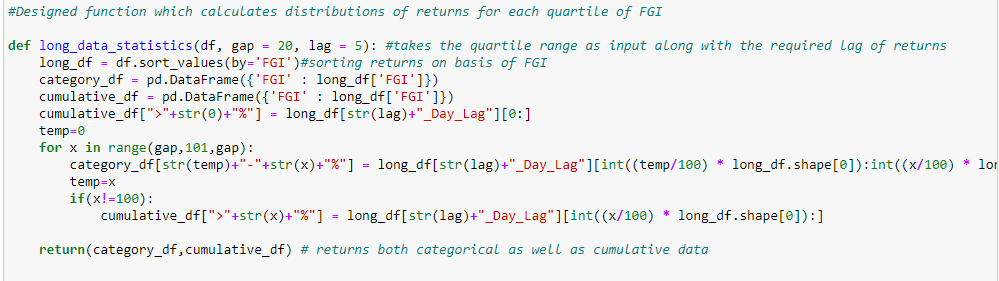
**Calculating Returns:**

Designed a function that can calculate returns for any “n” number of days with a given list of lags.

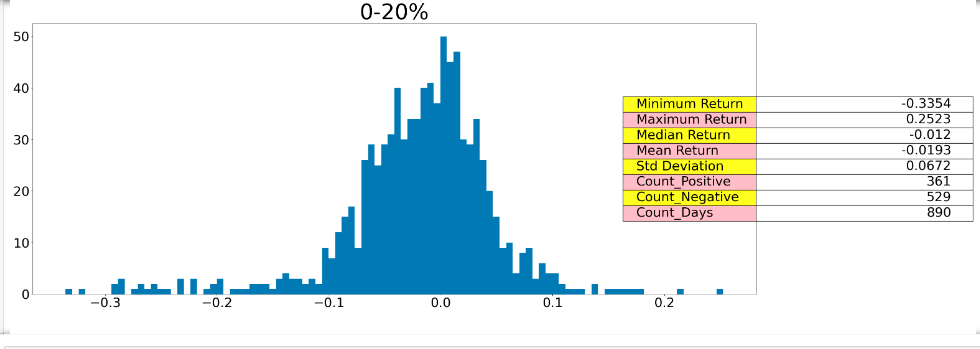


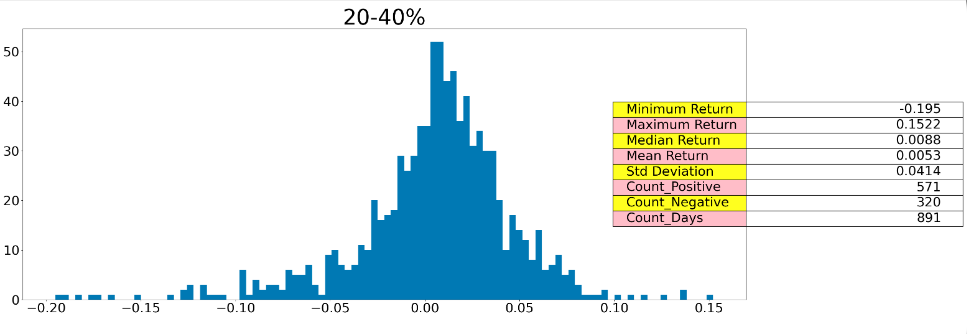
**FGI Based Percentile Wise Analysis:**

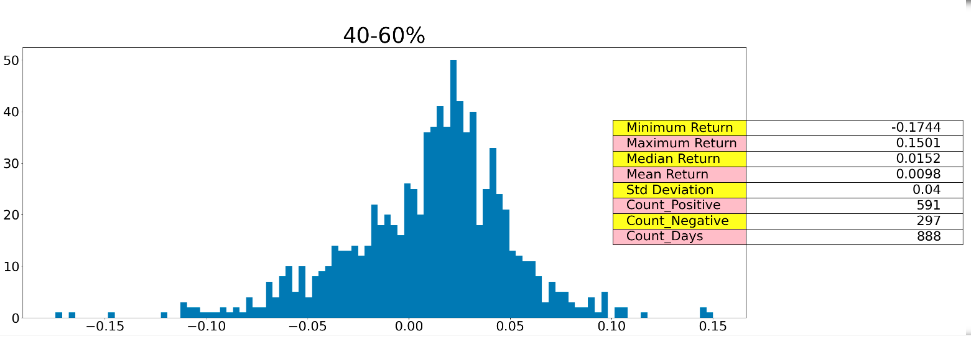
Designed a function to sort returns based on FGI scores categorically for each quartile. The function takes the required percentile range and the lag of the returns as input, and finds data both categorically and cumulatively.

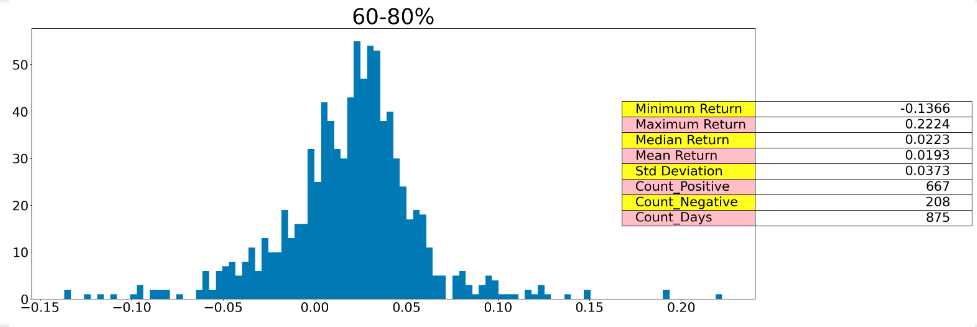


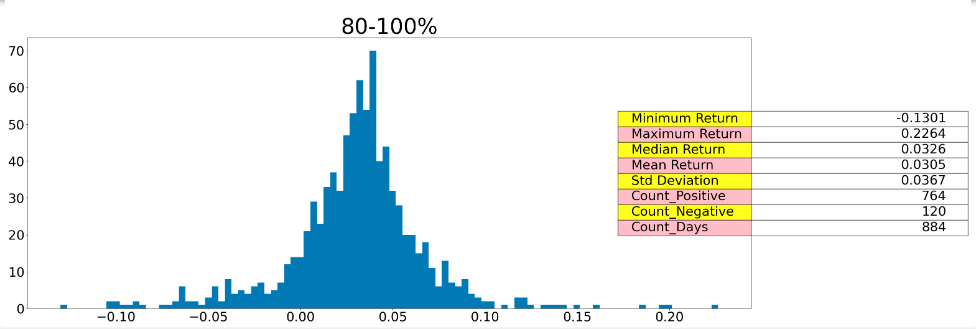
Categorical Distributions:











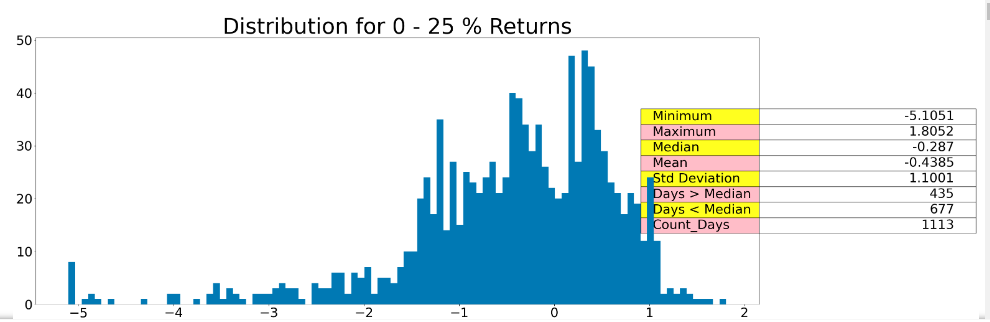
Note: *Created similar plots for cumulative data as well*

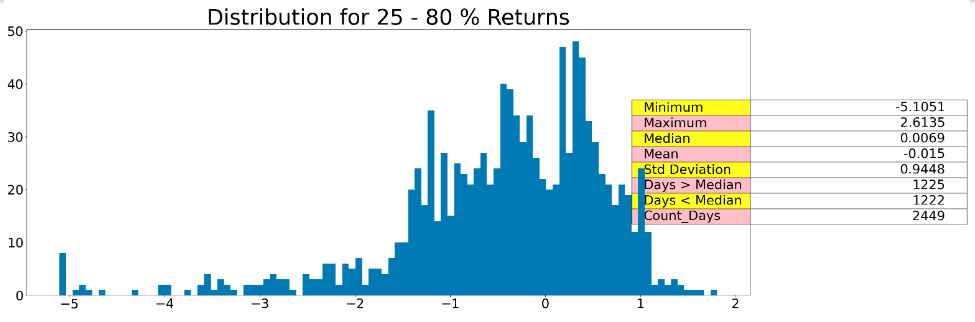
**Returns Based Percentile Wise Analysis of Every Feature:**

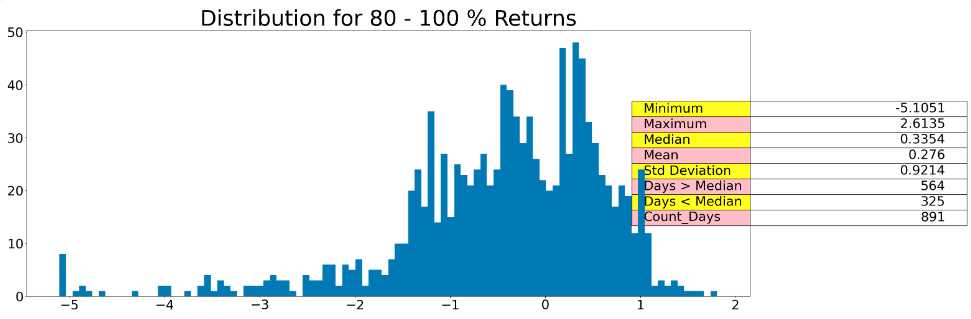
Conduced a similar analysis like before for each feature based on 3 cohort of returns i.e. low returns, medium returns, and high returns.

Designed a function that can sort the scores of each feature based on returns and plot them on a histogram for each cohort of returns and display some statistics about that data like minimum, maximum, median, mean, std deviation, days where the value of feature was more than its overall median, and lesser than its overall median.

*Example of the Analysis (Strength Score)*:

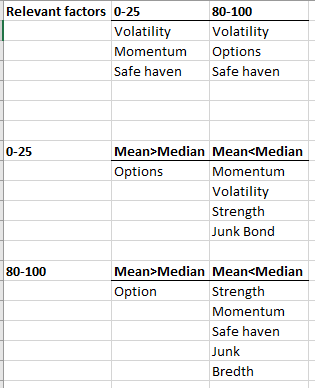






Note: *Similarly, this was done for all the other features as well*

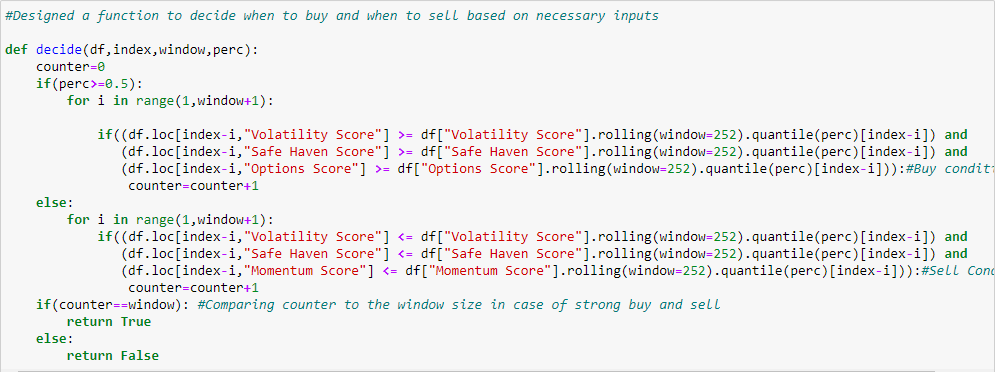
**Insights Drawn from the Analysis:**



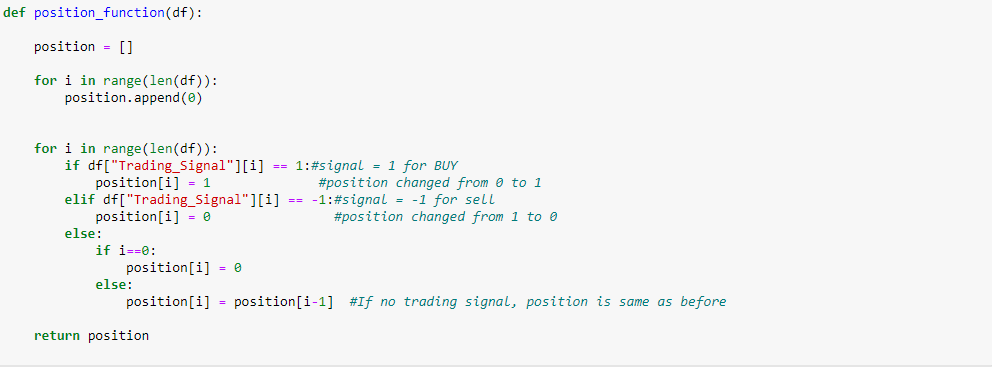
Note: *Therefore, Volatility and Safe Haven turned out to be leading indicators for high returns as well as low returns. Momentum was found to be a good indicator for low returns and Options Score seemed to be a good indicator for high returns. Based on these insights, trading strategies were built.*

**Designing functions for building trading strategies:**

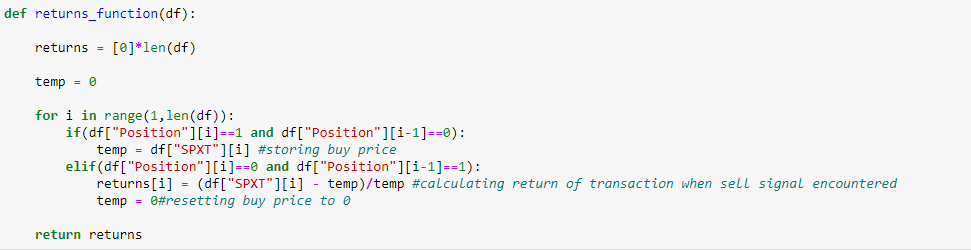
Designed the “*decide*” function which takes the decision to buy or sell. It takes as input the current index at which the decision needs to be taken and the window size i.e. the number of days prior to the current index whose signal needs to be looked at.



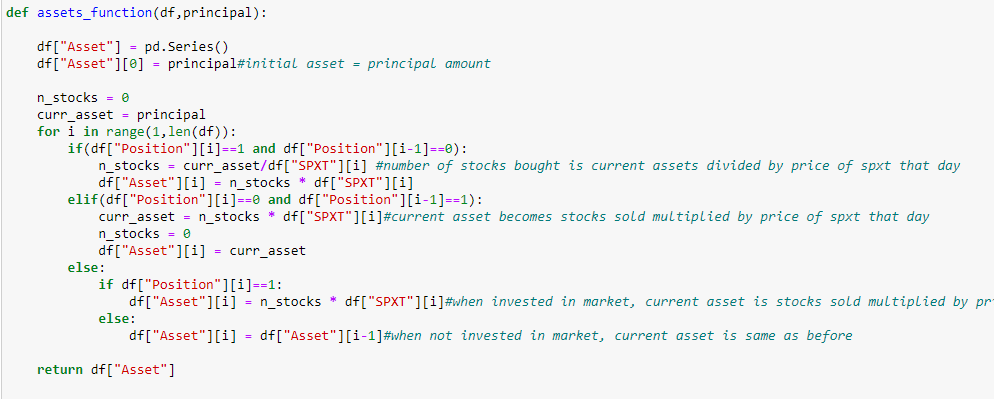
Designed the “*position\_function*” which returns the current position at every index i.e. 1 whenever we are invested in the market and 0 otherwise.



Designed the “*returns\_function*” which calculates the return associated with every transaction and shows it next to every to every selling price i.e. every time we are selling and squaring off our position.



Designed the “*assets\_function*” which calculates the current assets under management for every index. It continuously changes whenever we are invested in the market and remains constant when we are not.



**Building Trading Strategies:**

Based on above insights, 4 different strategies were built:

*Strategy 1*: Simple Buy and Sell signal, no fixed holding period

*Strategy 2*: Strong Buy and Sell signal (2 consecutive days), no fixed holding period

*Strategy 3*: Strong Buy (2 consecutive days) and Simple Sell signal, no fixed holding period

*Strategy 4*: Strong Buy (2 consecutive days) and Simple Sell signal, fixed holding period of 60 days

**Results:**

*Strategy 1*: CAGR = 14.56% , Drawdown = -42.55% , No. of Transactions = 31

*Strategy 2*: CAGR = 15.25% , Drawdown = -31.90% , No. of Transactions = 27

*Strategy 3*: CAGR = 15.25% , Drawdown = -31.90% , No. of Transactions = 27

*Strategy 4*: CAGR = 9.27% , Drawdown = 29.56% , No. of Transactions = 33

Following this, designed a function that can display a menu to choose from these 4 strategies, which will then execute the chosen strategy and display the CAGR, Drawdown, No. of transactions and current signal, and return 18 years of backtesting data as a dataframe.

Apart from this, also designed appending functions that can append daily data for each feature extracted from Bloomberg into the existing database.