Written By : Anshuta Awasthi
GT user name : aawasthi32
Date : 23 -03-2019

Manual Strategy

Part 1:

Following indicators have been used :

1. **Bollinger Bands ® :**
2. **Price / SMA(Simple Moving Average)**
3. **MACD(Moving Average Convergence Divergence)**

1. **Bollinger Bands® :** Bollinger bands is a technique of using simple moving average with upper and lower bands . In the figure below, orange curve in the upper subplot is depicting 20 day simple moving average and green and red curves are upper and lower bands respectively, which are outcome of simply adding and subtracting the 2 times standard deviation with the SMA. Standard deviation is the measurement of volatility.It shows how the stock price varies from it's true value. The bandwidth is the "envelope" between lower and upper bands.When the market becomes more volatile , bandwidth widens and for the period when market is less volatile, bandwidth narrows down. Bollinger Bands are effective trading indicators for providing market signals in terms of bandwidth(volatility measure) and breakouts (excursions above and below the bands).

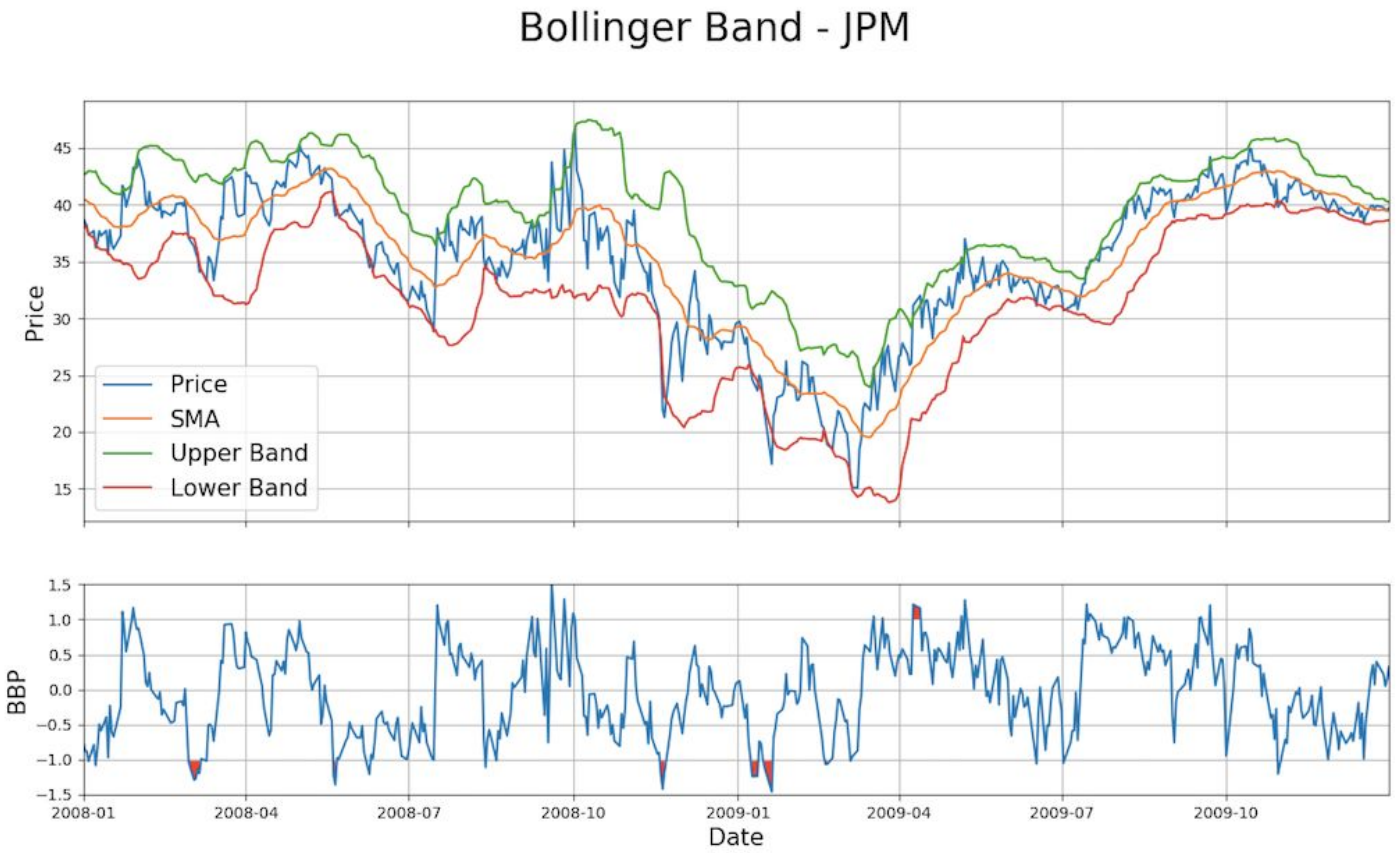
   Pseudo code for computation of Bollinger Band Percentage :
   1. The first step in calculating BB is to compute the simple moving average of the stock, typically using a 20-day lookback period. A 20-day lookback would average out the Adj. close price for the first 20 days . This is done by using pandas rolling mean with min_period.
   2. Next step is to measure standard deviation of "Adj close price" of the stock ,with the 20 day lookback window.Done by using pandas rolling std with min_period.
   3. Calculate the upper band and lower band values as follows :
      Upper Band = SMA + 2 times rolling standard deviation
      Lower Band = SMA - 2 times rolling standard deviation.
   4. Calculate BB percentage as below :
      BBP = (Price - SMA)/(2 * Rolling Std)

   BB for trading interpretation: Lecture video shows the method for using bollinger bands that might be effective.

   1. We look for instances when the price is outside of either upper and lower band and crosses to the inside the "envelope". When the price curve crosses the upper band from outside to inside that will

be “SELL” signal , because we have large excursion from the SMA and we see that it is going back from outside to inside towards average that means price will further go down.

2. Conversely, when the price curve crosses the lower band from outside to inside towards SMA ,that will be “BUY” signal indicating that price will go up towards average.
3. We calculate the percentage as shown in the formula in step 4 of pseudo code above .That will give the values of the excursions outside the bands i.e values >1 and values <-1.
4. **The second subplot in the below figure shows BB percentage values . Values above 1 and below -1 are the regions highlighted with red . These are the regions when the excursion crosses the band from outside to inside ,indicating SELL / BUY signals** .



2. Price over SMA :

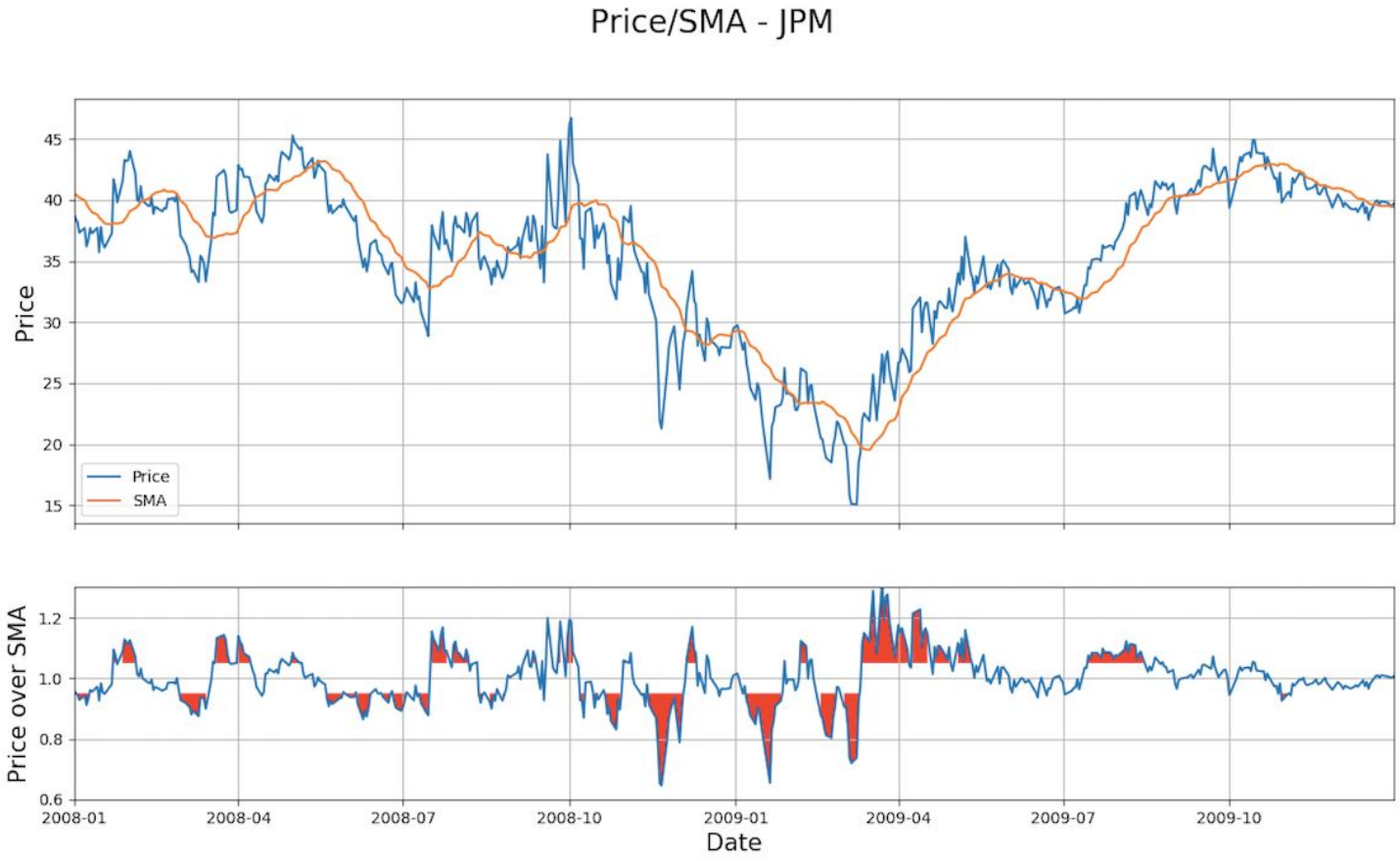
Simple Moving Average is very effective trading analysis tool. SMA is often used to identify trend direction. Moving average crossovers are commonly used by traders to generate potential buy and sell signals.Here we will use Price SMA Crossover.
Price crossovers depicts shifts in momentum and can be used as a basic entry or exit strategy. SMA over a 20 day lookback period can show the true value of stock .When we see large excursions(downwards / upwards) from that price, we should expect that price is eventually going to come back towards the average(an arbitrage opportunity). In first subplot of figure below , we see large price excursions on both sides of SMA.

Pseudo Code for Price/SMA calculation:

1. The first step in calculating BB is to compute the simple moving average of the stock, typically using a 20-day lookback period. A 20-day lookback would average out the Adj. close price for the first 20 days . This is done by using pandas rolling mean with min_period.
2. Calculate as follows :

   Price over SMA = Price/SMA

Price/SMA for trading interpretation: In the 2nd subplot we have the curve showing SMA crossover i.e. ratio of Price and SMA . In this plot , we have highlighted the areas(in Red) which are below 0.95 or above 1.05. The ratio of 1 denotes when the price is exactly same as SMA. The higher values of ratio depicts that price has gone up and will eventually drop back to average , so Price/SMA > 1.05 denotes the SELL signal . Conversely, values below 0.95 denotes that price has gone below average and will eventually rise up to the SMA level , hence shows “BUY” signal.



3. **MACD (Moving Average Convergence Divergence) :** MACD is one of the simplest and very effective momentum indicator. MACD gets derived from long exponential moving average(26 days window) and short exponential moving average(12 days window) , hence tells about the trends and momentum both.MACD is all about the convergence and divergence of the two moving averages. Convergence occurs when the moving averages move towards each other. Divergence occurs when the moving averages move away from each other. The shorter moving average (12-day) is faster and responsible for most MACD

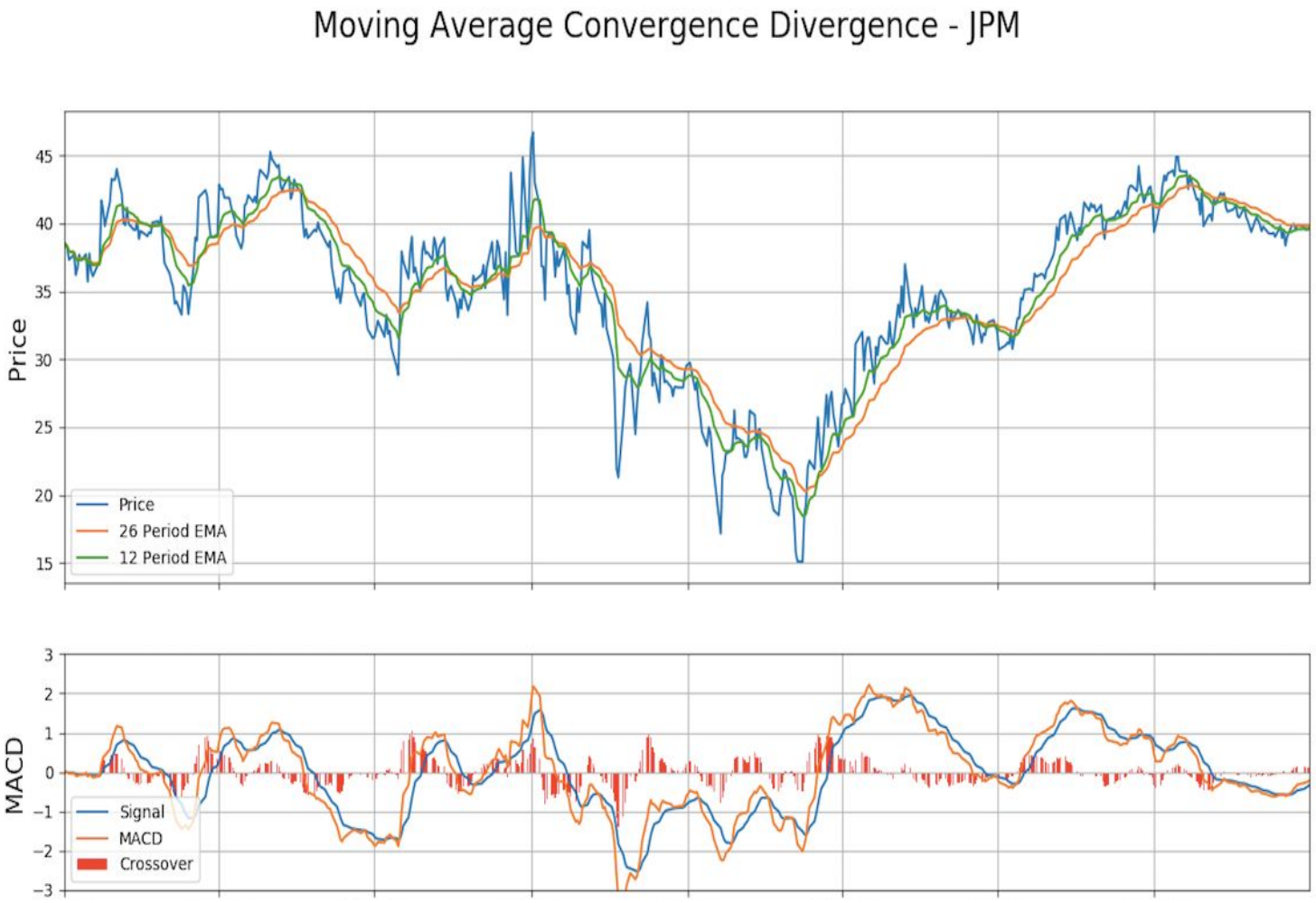
movements. The longer moving average (26-day) is slower and less reactive to price changes in the underlying stock.

Pseudo Code for MACD calculation:

1. *Calculate long Exponential Moving Average , using pandas series.ewm (exponential weighted functions) with rolling window of 26 days.*
2. *Calculate short Exponential Moving Average , using pandas series.ewm (exponential weighted functions) with rolling window of 12 days.*
3. *Calculate :*
   MACD = Short Exp. Moving Avg. of price(12 Days) - Short Exp. Moving Avg. of price (26 Days)
4. *Calculate :*
   Signal = Exp. Moving Avg. of MACD with 9 days rolling window

5. ***MACD crossover** = **MACD - Signal***

MACD for trading interpretation: The first subplot , shows the price of JPM stock with 12 days EMA and 26 days EMA lines.
The second subplot in the below figure shows MACD , which oscillates above and below the zero line (centerline) When 12 day EMA is larger than 26 days EMA(positive MACD) ,it indicates upside momentum is increasing .Conversely , when 12 day EMA is smaller than 26 day EMA (negative MACD) values , indicates downside momentum is increasing.Traders use 2 types of crossovers to interpret the trends and momentum.**A "zero crossover" event** occurs when the MACD series changes sign, that is, the MACD line crosses the horizontal zero axis. This happens when there is no difference between the fast and slow EMAs of the price series. A change from positive to negative MACD is interpreted as "bearish", and from negative to positive as "bullish". Zero crossovers provide evidence of a change in the direction of a trend.
Shown as orange curve in 2nd subplot in the figure below
A "signal-line crossover" occurs when the MACD and signal lines cross; that is, when the divergence (the bar graph) changes sign(Shown as red bar graph in the figure below).The standard interpretation of such an event is a recommendation to buy if the MACD line crosses up through the signal line (a "bullish" crossover), or to sell if it crosses down through the average line (a "bearish" crossover). These events are taken as indications that the trend in the stock is about to accelerate in the direction of the crossover.



Part 2: In Theoretically optimal strategy we have adopted a simple approach . Each day we are comparing prices with previous day, if price is showing increase from previous day then GO LONG and if on particular day ,price is showing drop from previous day then GO SHORT.

Following steps have been performed:

1. Fetch the prices for “JPM” symbol for the given date range in a prices dataframe.
2. Initialize 2 new data frames for trades and benchmark.
3. Added one more column “Diff” in the prices dataframe storing the difference between the today’s price and day before’s price.
4. Initialize the holdings to 0.

5. Populated trades dataframe for each trading day based on following rules :
   Pseudo code:

If (Today’s price is greater than yesterday’s i.e price[“diff”] >0 and holdings == 0):
Then add 1000 shares(BUY) in trades dataframe for that day
Set holdings to 1000 shares(GO LONG)

Else If (Today’s price is greater than yesterday’s i.e price[“diff”] >0 and holdings == -1000):
Then add 2000 shares in trades dataframe for that day.
Add 2000 shares to holdings ,making net holding values to 1000 shares (GO LONG)

Else if (Today's price is greater than yesterday's i.e price["diff"] >0 and holdings == 1000):
Since we are already holding 1000 shares(In long position). Do nothing.

Else if (Today's price is less than yesterday's i.e price["diff"] < 0 and holdings == 0):
Then add -1000 shares(SELL) in trades dataframe for that day
Set holdings to -1000 shares(Go SHORT)

Else if (Today's price is less than yesterday's i.e price["diff"] < 0 and holdings == 1000):
Then add -2000 shares(SELL) in trades dataframe for that day
Add -2000 shares to holdings,making net holding values to -1000 shares(Go SHORT)

Else if (Today's price is less than yesterday's i.e price["diff"] < 0 and holdings == -1000):
Since we are already going short with -1000 shares. Do nothing.

6. Invoke compute_portvals from "marketsimcode.py" to get the portfolio values on each trading day."marketsimcode.py" is modified to accept trades dataframe(resulting in step5) , instead of reading orders files.

7. Prepare benchmark dataframe with BUY of 1000 shares on first day and no further transactions. Compute benchmark port vals also by invoking compute_port_vals from marketsimcode.py.

8. Plot portfolio and benchmark curves in the graph. Calculate values of cumulative return , standard deviation of daily return and average daily return for portfolio and benchmark respectively.

Results :

Date Range: 2008-01-01 to 2009-12-31

Cumulative Return of Portfolio: 5.7861
Cumulative Return of Benchmark : 0.0123

Standard Deviation of Portfolio: 0.00454782319791
Standard Deviation of Benchmark : 0.0170043662712

Average Daily Return of Portfolio: 0.00381678615086
Average Daily Return of Benchmark : 0.000168086978191

Final Portfolio Value of Portfolio: 678610.0
Final Portfolio value of Benchmark :101230.0

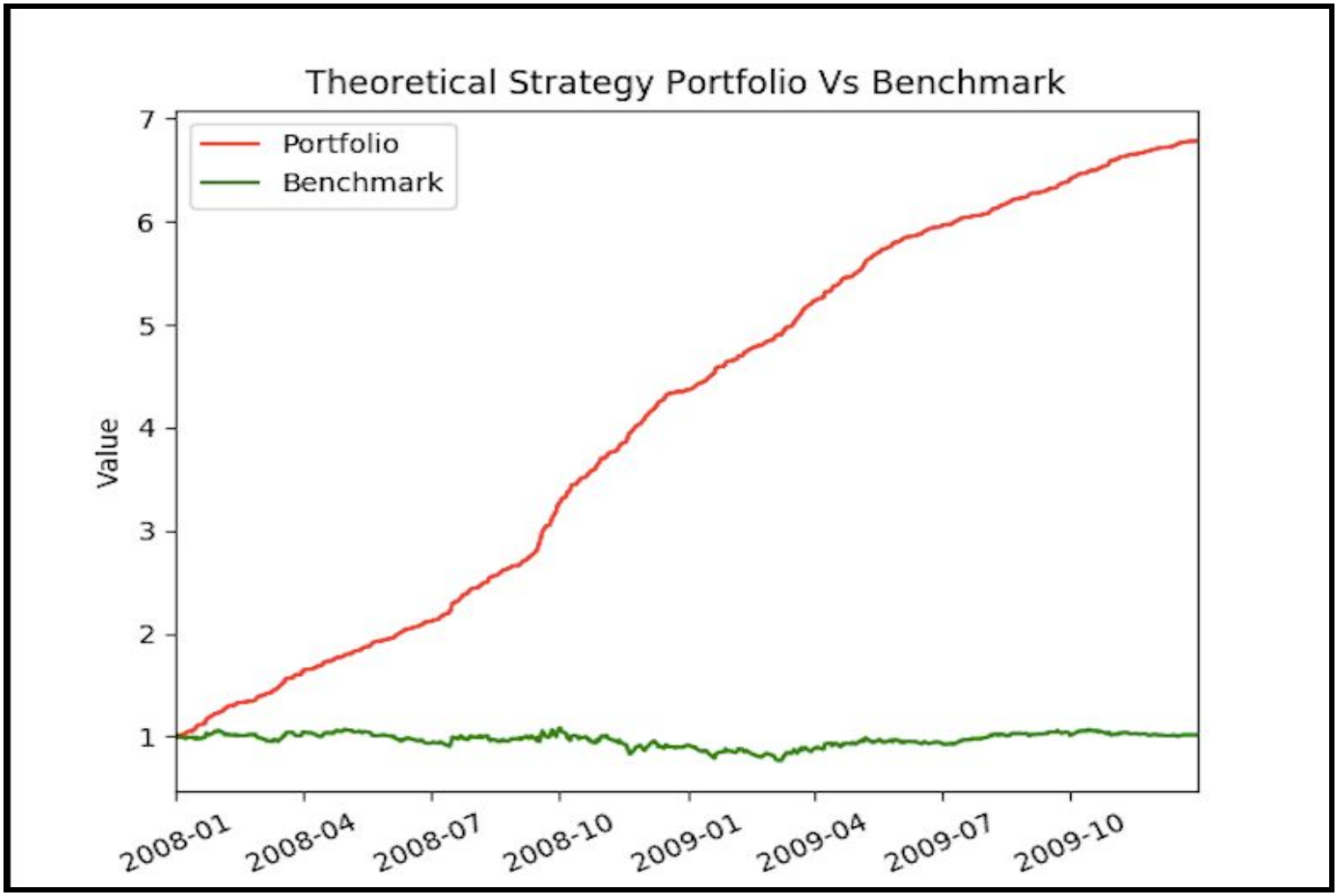


Figure : **Part2.png** generated by TheoreticallyOptimalStrategy.py

Part 3: In manual strategy , 3 indicators(Bollinger Bands , Price/SMA and MACD) as described in part 1 have been used to come up with trading strategy to decide the LONG and SHORT conditions.

1. Go LONG whenever any of these conditions are True:
   a).Price/SMA ratio is <0.90 AND BBP <0
   b). MACD crossed signal upwards i.e. macd_signal_crossover[day-1]<0 and macd_signal_crossover[day]>0

2. Go SHORT whenever any of these conditions are True:
   a).Price/SMA ratio is > 1.10 AND BBP > 0
   b). MACD crossed signal downwards i.e. macd_signal_crossover[day-1] > 0 and macd_signal_crossover[day] < 0

Following approach has been used :

1. Get the price for “JPM” symbol for the given date range in prices dataframe.

2. Read the past prices also to take lookback window in account for calculation on indicators.
3. Calculate the dataframes for Bollinger bands , MACD and Price/SMA for the given In-Sample date range by invoking these functions written in indicators.py
4. Initialize trades dataframe values and holdings to 0. Initialize 2 lists holding dates long and short entry points respectively.
5. Populate trades values with no. of shares traded as per the following strategy for each day:

Pseudocode :

If (priceoversma < 0.90 AND bbp <0) OR (MACDSignalCrossover for previous day < MACDSignalCrossover for current day) AND holdings == 0):

Then add 1000 shares(BUY) in trades dataframe for that day
Set holdings to 1000 shares(GO LONG)
Long_date.append (current day)

Else If (priceoversma < 0.90 AND bbp <0) OR (MACDSignalCrossover for previous day < MACDSignalCrossover for current day) AND holdings == -1000):

Then add 2000 shares in trades dataframe for that day.
Add 2000 shares to holdings ,making net holding values to 1000 shares (GO LONG)
Long_date.append (current day)

Else If (priceoversma < 0.90 AND bbp <0) OR (MACDSignalCrossover for previous day < MACDSignalCrossover for current day) AND holdings == 1000):

Since we are already holding 1000 shares(In long position). ALREADY LONG.Do nothing.

Else If (priceoversma >1.10 AND bbp > 0) OR (MACDSignalCrossover for previous day > MACDSignalCrossover for current day) AND holdings == 0):

Then add -1000 shares(SELL) in trades dataframe for that day
Set holdings to -1000 shares(Go SHORT)
Short_date.append (current day)

Else If (priceoversma >1.10 AND bbp > 0) OR (MACDSignalCrossover for previous day > MACDSignalCrossover for current day) AND holdings == 1000):

Then add -2000 shares(SELL) in trades dataframe for that day
Add -2000 shares to holdings,making net holding values to -1000 shares(Go SHORT)
Short_date.append (current day)

Else If (priceoversma >1.10 AND bbp > 0) OR (MACDSignalCrossover for previous day > MACDSignalCrossover for current day) AND holdings == -1000):

Since we are already going short with -1000 shares. ALREADY SHORT .Do nothing.

6. Invoke compute_portvals from "marketsimcode.py" to get the portfolio values on each trading day of In-Sample date range. Pass the commission and impact values as 9.95 and 0.005 respectively along with trades dataframe resulting in step 5.
."marketsimcode.py" is modified to accept trades dataframe , instead of reading orders files.

7. Prepare benchmark dataframe with BUY of 1000 shares on first day and no further transactions. Compute benchmark port vals also by invoking compute_port_vals from marketsimcode.py. Pass the commission and impact values as 9.95 and 0.005 respectively along with benchmark dataframe.

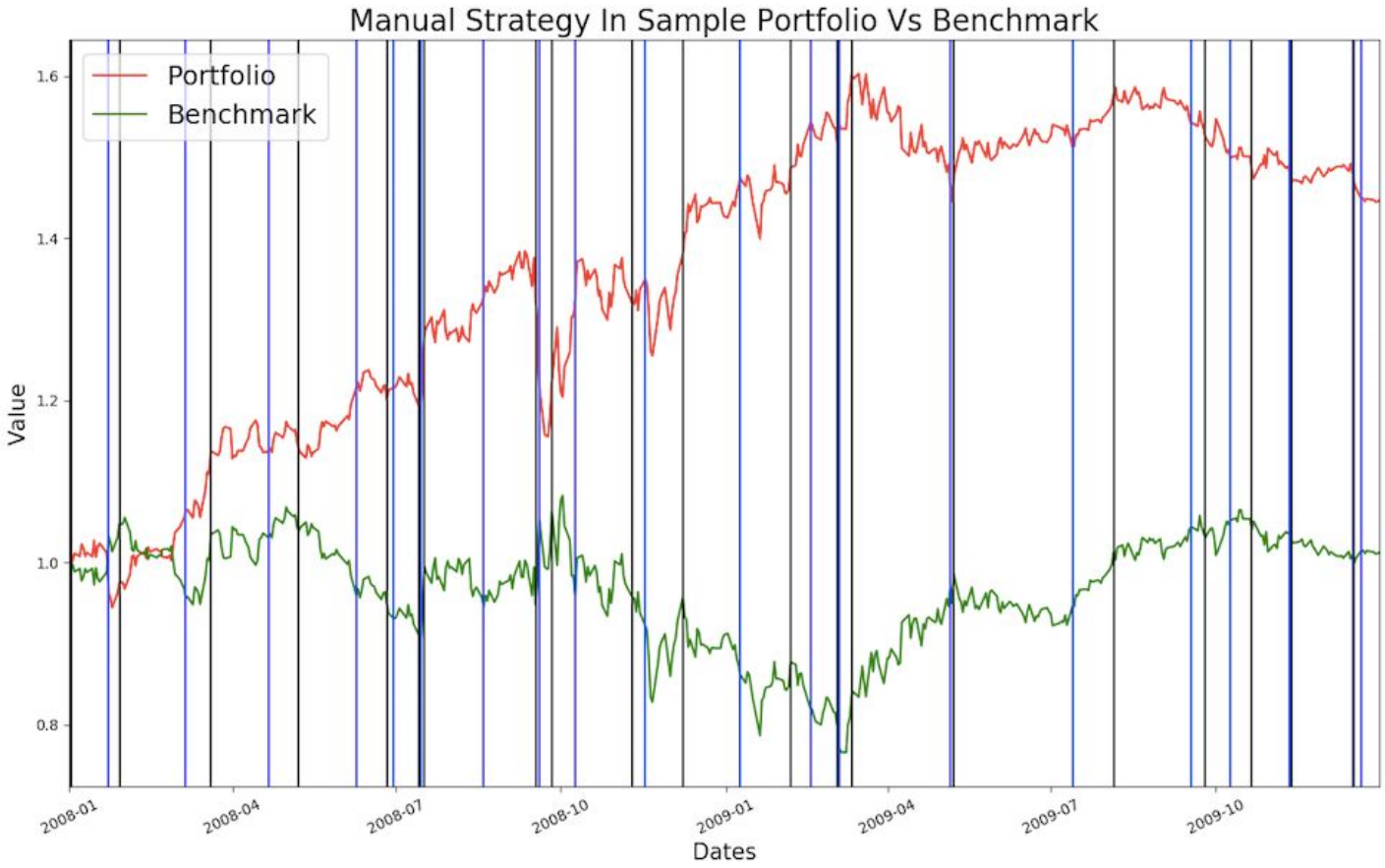
8. Plot portfolio and benchmark curves in the graph. Plot vertical blue line on entries in Long_date list and black on Short_date list.

9. Calculate values of cumulative return , standard deviation of daily return and average daily return for portfolio and benchmark respectively.

Strategy Selection : I tried many combinations of 3 indicators to reach at the desired , in-sample performance. Putting all the 3 indicator threshold condition with AND was giving very less trading dates and hence resulting in poor returns.
I tweaked my code to change indicator conditions as well as threshold in a way that it should avoid overfitting. Widening the threshold of “Price over SMA” as going LONG for < 0.90 and short for >1.10 helped reducing overfitting as well as reduction in no. of trading days resulting in saving commision and impact cost too. Looking at the resulted bollinger band dataframe , there were very few instances when “bbp” value was falling either >1 or less than -1 range , hence to improve the performance I also modified the BBP threshold range in my strategy, resulting in following graphs .

I believe that my strategy is performing reasonably well in in-sample data . Approx. 44% overall return shows that strategy is not suffering from overfitting and not too bad either.
Before changing the range of “price over sma”, my strategy was giving 65 % returns which made me suspicious of overfitting.After changing the threshold range from (lower 0.95 and higher 1.05) to (lower 0.90 and higher 1.10) , I have arrived at satisfactory in-sample performance.

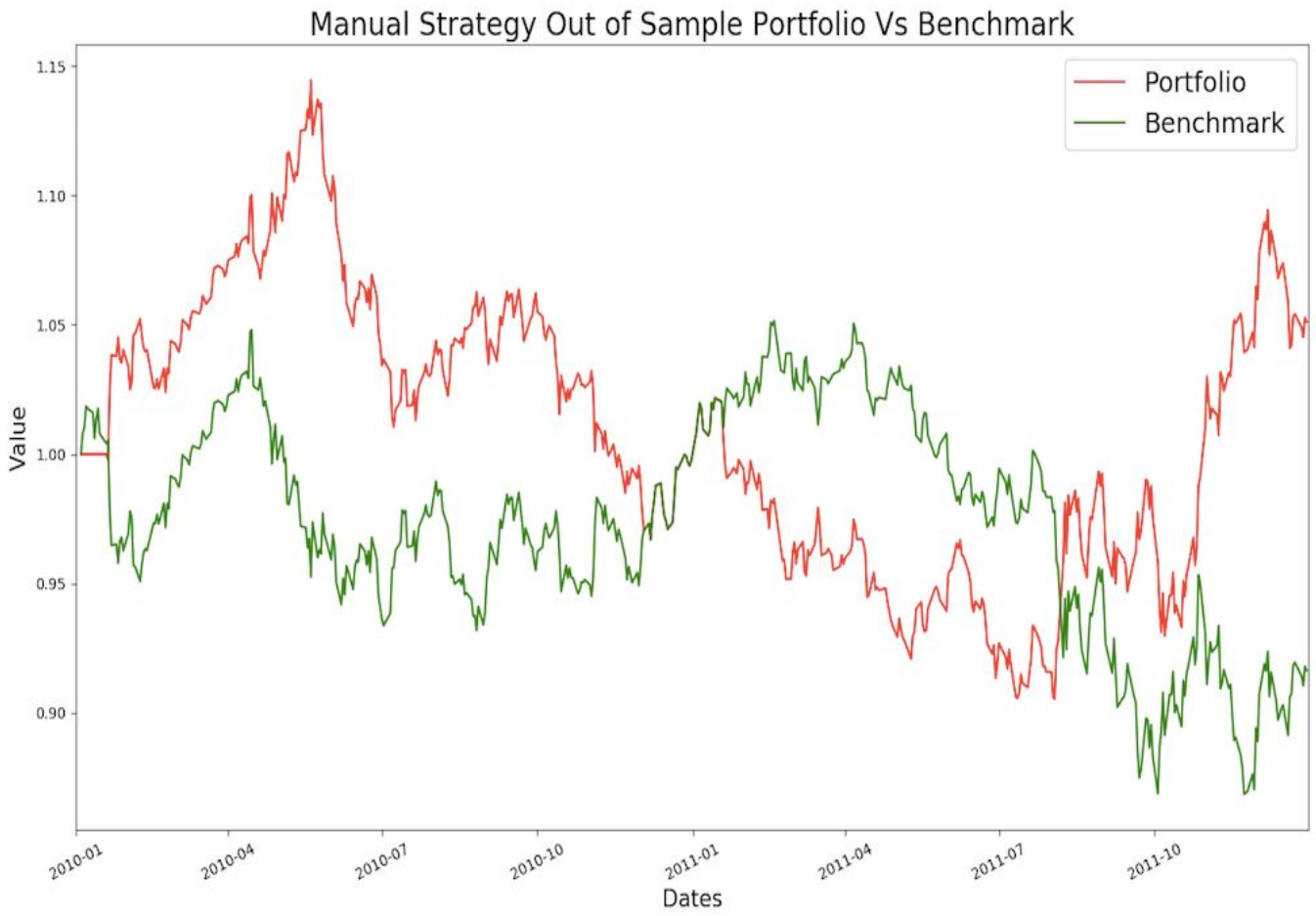
Figure : Part 3.png generated by manualstrategy.py



Part 4:

Calculated the out-sample results by populating trades dataframe using the same strategy described in Part 3 for JPM prices between the range 2010-01-01 to 2011-12-31.Then calculated port vals for benchmark as well as portfolio by passing trades and benchmark dataframes to marketsimcode.py.
Graphical and numerical results are shown below:

Figure : Part 4.png generated by manualstrategy.py



Out sample results don't show very great results, manual strategy gives just 5% results, however it still performs way better than out of sample benchmark returns . Benchmark results are showing -ve cumulative return , which means for benchmark it resulted in loss instead.

It is evident that our strategy would perform better for in-sample data than out sample , since we have developed our strategy and tweaked the parameters to get good in-sample results. Also , Manual Strategy performed better than benchmark for both in-sample and out-sample data in terms of cumulative returns and average daily returns.

However , standard deviation for benchmark is slightly more than Manual Strategy in both the scenarios . Which is also good and indicates that our strategy makes the daily returns less volatile

In-Sample Result Performance:
Date Range : 2008-01-01 to 2009-12-31
Starting Value = 100000$
Commision = 9.95 $
Impact = 0.005

	Manual Strategy	Benchmark
Cumulative Return	0.446792360946	0.0123249333401
Std. Deviation of Daily Return	0.0122067377814	0.0170412470682
Mean of Daily Return	0.000807572180922	0.000168759162146
Final Portfolio value	144386.55	101027.7

Out Sample Result Performance:
Date Range : 2010-01-01 to 2011-12-31
Starting Value = 100000$
Commision = 9.95 $
Impact = 0.005

	Manual Strategy	Benchmark
Cumulative Return	0.051134	-0.0835791100328
Std. Deviation of Daily Return	0.00818776016435	0.00850015832233
Mean of Daily Return	0.000132540860241	-0.000137429230389
Final Portfolio value	105113.4	91445.7

References:
https://www.investopedia.com/terms/m/macd.asp
https://en.wikipedia.org/wiki/MACD
https://www.fidelity.com/viewpoints/active-investor/how-to-use-macd