Sujit Kanti Biswas
GT User ID: sbiswas67
GT ID: 903549376

Project 6: Indicator Evaluation

Part 1: Technical Indicators

1. Simple Moving Average (SMA)

What is SMA ?

A SMA is a moving average that is calculated by summing the adjusted close price for a "n" number of time periods and then dividing by the number of time periods (n).

$$SMA_t^{(n)} = \frac{1}{n}\sum_{i=0}^{n-1} Price_{t-i}$$

It is an indicator that smoothens the price curve volatility and demonstrates the lagged characteristics of the stock price movement. It is also called a rolling mean across data points.

For example:

1) If the last five closing prices of a stock are: 28.93+28.48+28.44+28.91+28.48 = 143.24
2) We divide the total of the closing prices by the number of periods. By this we get a 5-day SMA. Here 5 is the moving average "window" length.
3) So, 5-day SMA = 143.24/5 = 28.65

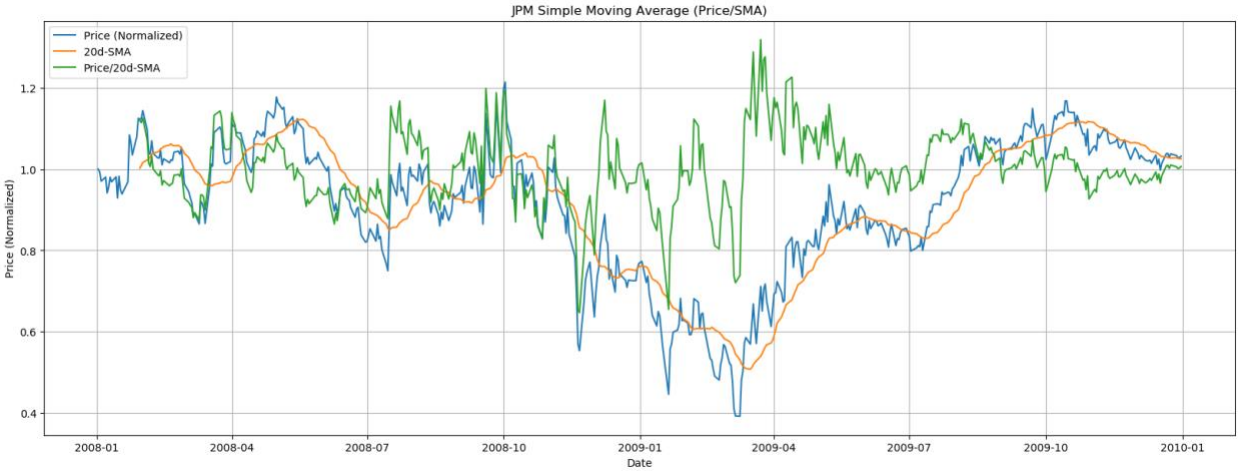
Some of the major popular moving averages used by most traders are the 10, 20, 50, 100 and 200.

Indicator to be used:
In this project, I have considered a 20-day SMA and then eventually a "price per 20-day SMA" calculation. Please refer to the plot below.

"Price/20-day SMA" represents today's adjust close price of the stock divided by SMA of the past 20 days.
According to David Byrd's strategy explained in the class, one of the uses of this ratio for a strategy can be to determine when to close a position by classifying in 3 ways,

- If the ratio is greater than 1 then we are above the SMA,
- Equals 1 indicates we are at the SMA and
- Less than 1 indicates being below the SMA

We can use Price/SMA in our trading strategy (as explained by David Byrd in the class) where Price/SMA dropping below 0.95 can be used to enter a long position while Price/SMA moving above 1.05 can be a signal to enter a short position.



2. Bollinger Band and BB%:

What is Bollinger Band ?
Bollinger Band is another Divergence based market indicator where it shows divergence between the stock and the market. It is a technical analysis tool developed by John Bollinger for generating oversold or overbought signals.

There are three lines that compose Bollinger Bands: A simple moving average (middle band) and an upper and lower band.

- Upper band (2 standard deviations above SMA)
- Middle band (n-Day SMA)
- Lower band (2 standard deviations below SMA)

It is believed that the closer the prices move to the upper band, the more overbought the market, and the closer the prices move to the lower band, the more oversold the market.

BBs provides a visual of volatility (bands are highly separated) , and to determine “breakout” prices, where the price rises above the upper band, or drops below the lower band, signifying potential purchases or selling points.

Indicator to be used:
In our project we can use a different aspect of the Bollinger Band called Bollinger Band Percentage (BB %) as it returns a quantitative measure that can be used to build a trading strategy.

The BB% quantitively points how far the price is from the Lower Band to the Upper Band.

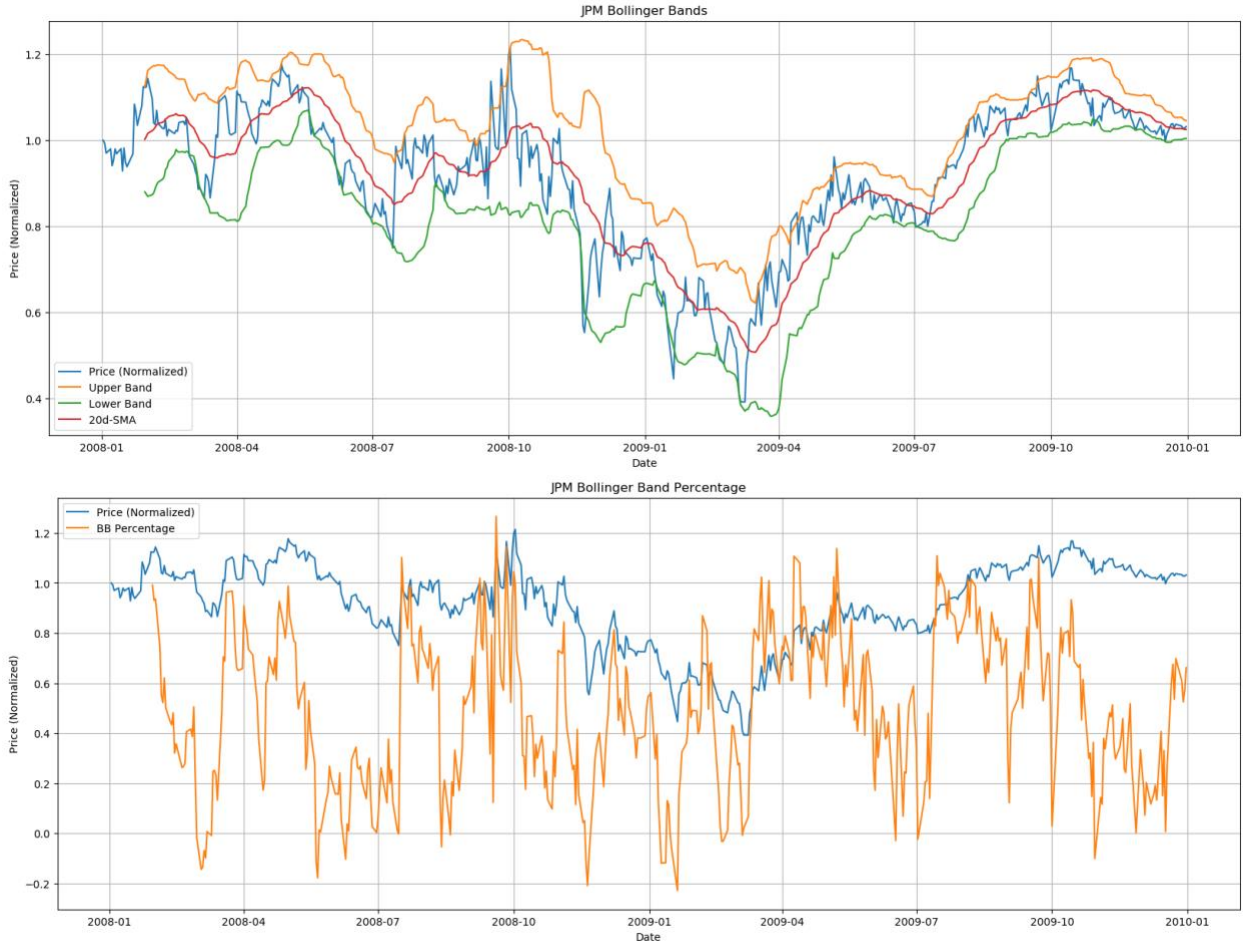
- 0 means at the bottom band,
- 0.5 at the SMA and

- +1 at the top band

As pointed out in David Byrd's lecture, we can build an effective trading strategy by determining some cut-off on the BB% as shown below:

- If BB% greater that 1 then the stock is overbought.
- If BB% lower that 0 then the stock is oversold.

Below we have plotted the Bollinger Bands across 20-Day SMA and subsequently the BB% on a separate plot for better readability.



3. Moving Average Convergence Divergence (MACD):

What is MACD ?

MACD stands for Moving Average Convergence Divergence. It is a trend-following, trend-capturing momentum indicator, that shows the relationship between two moving averages (MAs) of prices. MACD is a lagging indicator.

Computing MACD:

1. Compute the 12-period EMA of prices
2. Compute the 26-period EMA of prices

3. Subtract 2 from 1. That is the fast MACD line.
4. Compute the 9-period EMA of 3. That is the slow Signal line which is plotted on top of the MACD, usually marking triggers for buy and sell signals.

Indicator to be used:
We shall use Crossovers strategy of the MACD line with the Signal Line.
When the fast MACD line rises above the slow signal line, it means a bull cycle has begun line (a "bullish" crossover). If the fast MACD line crosses under the slow signal, a corrective period has started (a "bearish" crossover).

This upward and downward Crossover points can be identified in Python by creating datasets from MACD and MACD-Signal data with a shift() of 1-day, 2-day, etc. and comparing them respectively.

Refer to the grapy plot below. Crossover points are intersection of the MACD (orange line) with Signal line (green line).

4. Stochastic Oscillator:

What is SO ?
A stochastic oscillator is a momentum indicator that compares the closing price of a stock to the range of its prices over a certain period of time.

This indicator is calculated using the following formula:

%K = 100(Close – L14) / (H14 – L14)

Where:
Close = the most recent closing price
L14 = the low of the 14 previous trading sessions

H14 = the highest price traded during the same 14-day period
%K= the current market rate for the currency pair
%D = 3 period moving average of %K

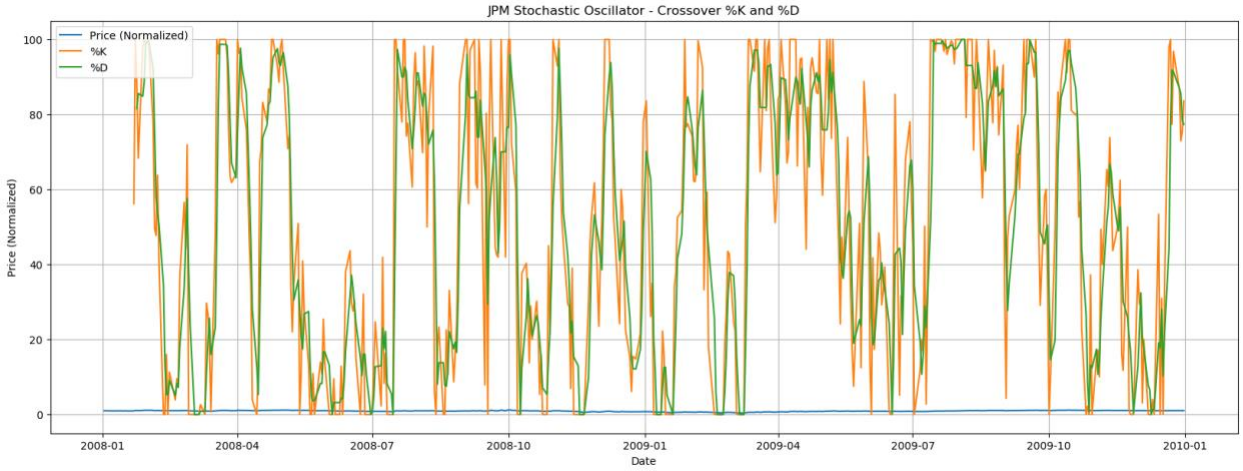
Indicator to be used:

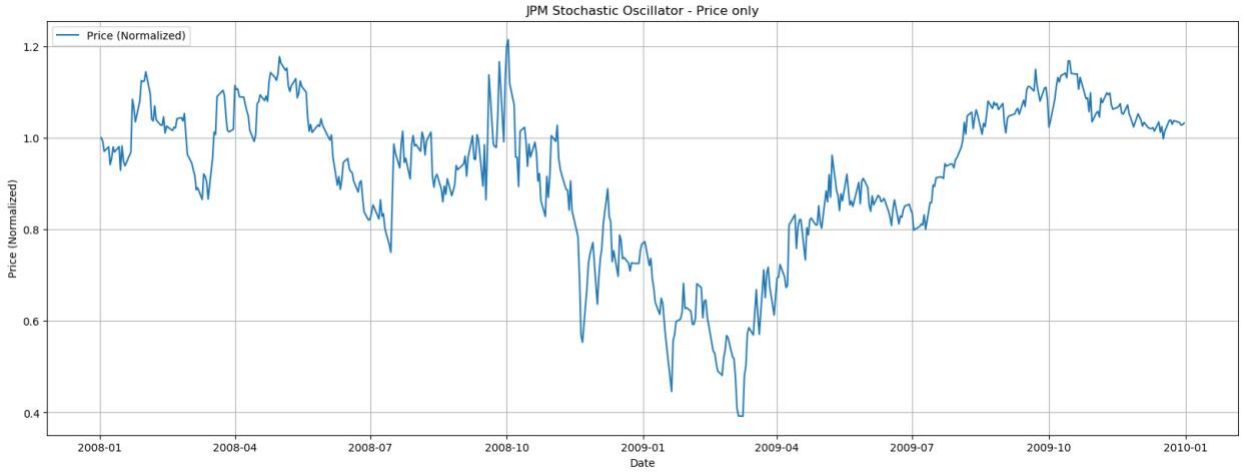
We can implement the following SO indicator strategy programmatically in Python that can give us the following three signals as explained below:

1) A SELL ENTRY signal when the %K line crosses down through the %D line and the %K line is above 80.
2) The SELL EXIT signal for this short position shall be given as soon as the %K line crosses back upward through the %D line, irrespective of the actual value of the %K.
3) A BUY ENTRY signal when the %K line passes upward through the %D line, and the %K line is under 20 at that time.
4) The BUY EXIT for this long position shall be given as soon as the %K line crosses back downward through the %D line, irrespective of the actual value of the %K line when this happens.
5) All other conditions shall result in FLAT signal (which means no position).

This upward and downward Crossover points can be identified in Python by creating datasets from %K and %D data using a shift() function of 1-day, 2-day, etc. and comparing them respectively.

Please refer to the plot below:
In the first plot the Blue colored “price” line looks skewed at the bottom since the “%K” value can reach as high as 100. That’s why a separate second plot is used to show the price (normalized) variation.





5. Commodity Channel Index (CCI):

What is CCI ?

The Commodity Channel Index (CCI) is a versatile indicator which can be employed to identify a new trend or warn sign of extreme conditions. CCI can be used to measure the current price level which is relative to an average price level over a given period of time.

- CCI shall be relatively high when prices are far above their average level (overbought).
- On the other hand, CCI shall be relatively low when prices are quiet below their average level (oversold).
- CCI can be used in this way to identify overbought and oversold levels.

This indicator measures the difference between a stocks' price change and its average price change.

- High positive readings point that prices are well above their average, which is a sign of strength.

- Low negative readings point that prices are well below their average, which is a show of weakness.
- This indicator can be used as either a coincident or a leading indicator.

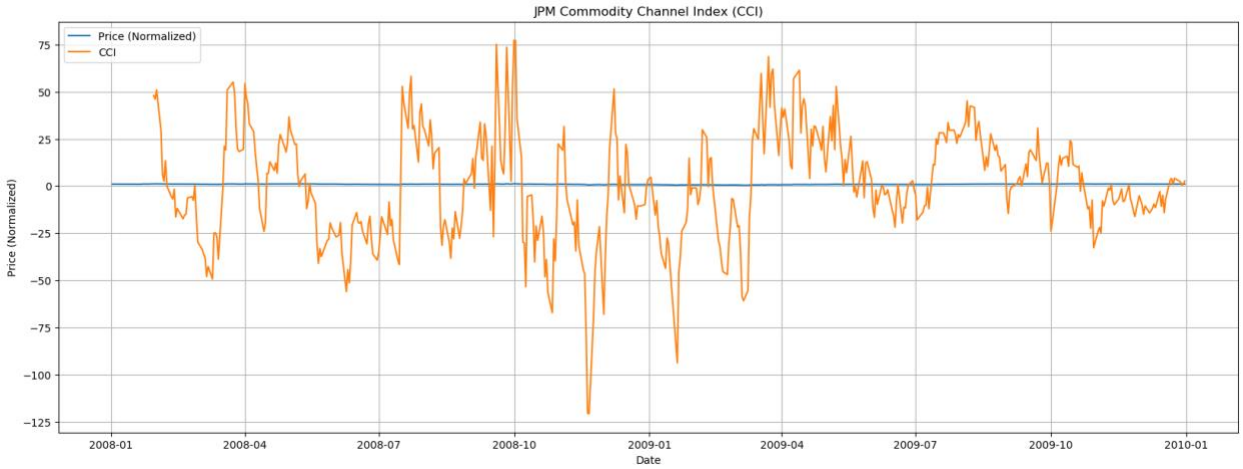
In case of CCI the definition of overbought or oversold can be varied. In theory, there are no upside or downside limits in this case which makes an overbought or oversold assessment subjective. Also, symbols can continue to move higher after an indicator becomes overbought, as well, can continue moving lower after an indicator becomes oversold.

The CCI Indicator can be calculated using the below formula:

CCI = (Prices – 20-DAY SMA) / (.015 * Standard_Dev(Prices))

<u>Indicator to be used:</u>

The value of CCI indicator can be used as a coincident indicator, surges above +100 reflect strong price action that can signal the start of an uptrend (Bullish).
If plunges below -100 reflect weak price action that can signal the start of a downtrend (Bearish).



Part2: Theoretically Optimal Strategy

Assumptions:

In the Theoretically Optimal Strategy, we assume that we can peek into the future. However, we are constrained by the portfolio size and order limits as mentioned below:

1. Net holdings are constrained to -1000, 0, and 1000.

2. Trade amount are constrained at
   - +1000.0 indicating a BUY of 1000 shares
   - -1000.0 indicating a SELL of 1000 shares
   - 0.0 indicating nothing
   - +2000.0 indicating a BUY of 2000 shares as long as we maintain the Net holding constraints
   - -2000.0 indicating a SELL of 2000 shares as long as we maintain the Net holding constraints.

3. The transaction costs like, commission = 0.00 and impact = 0.00
4. Price Date Range: 2008-1-1 to 2009-12-31

Approach:

A. Portfolio:

As part of my Theoretically Optimal Strategy, I followed the following approach to build the trade dataframe (df_trades):

Step 1: Loop till the second last day [len(df_prices.index) - 1] of the available price dates.

(This is because the trade amount on last day shall be zero as we cannot see the future prices.)

Step1a:

If price_today < price_tomorrow and net_holding = 0 then BUY 1000 shares
If price_today < price_tomorrow and net_holding = -1000 then BUY 2000 shares
If price_today < price_tomorrow and net_holding = 1000 then HOLD
If price_today > price_tomorrow and net_holding = 0 then SELL 1000 shares
If price_today > price_tomorrow and net_holding = 1000 then SELL 2000 shares
If price_today > price_tomorrow and net_holding = -1000 then HOLD
If any other conditions then HOLD

Step 2: On the last available price day:

Step 2a: SELL “remaining shares”

Step 3: The trade dataframe (df_trades) is then passed to the compute_portvals() function that

was implemented in market simulator program (marketsimcode.py) to compute the daily holdings and eventually the daily portfolio values.
Also generated the graph and below mentioned statistics:

a) Cumulative return of the benchmark and portfolio

b) Standard Deviation of daily returns of benchmark and portfolio

c) Mean of daily returns of benchmark and portfolio

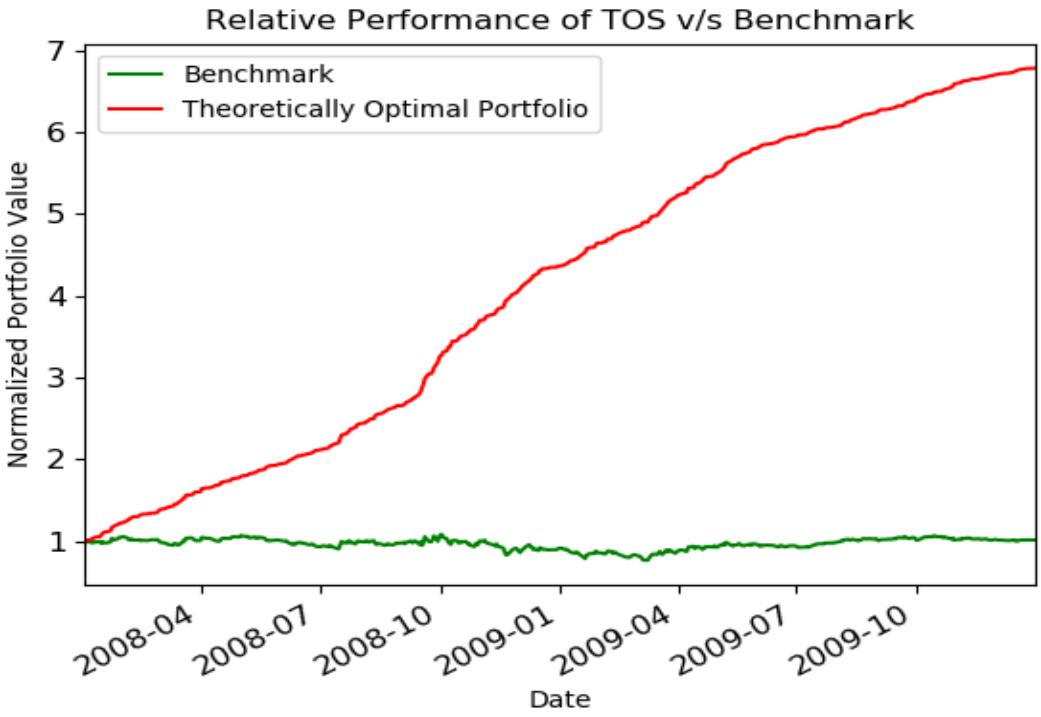
B. Benchmark:
Benchmark is defined as the performance of a portfolio of "JPM" stock starting with $100,000 cash and investing in 1000 shares and holding that position.
The purpose of the benchmark is to set a checkpoint which should be attained at the least by the trading strategy and provides a baseline for comparison.

To implement this, I create a dataframe (df_trades) for benchmark with just 2 rows, 1 each for the
starting and ending dates of transactions.

- On the first price day, execute a BUY 1000 shares of JPM.
- On the last day is SELL the 1000 shares of JPM.

Chart with performance illustration:

A comparison of the benchmark (in green line) with the value of the theoretically optimal portfolio (in red line) both portfolio values are normalized to 1.0 at the start as shown in the graph below:



TOS Portfolio versus Benchmark portfolio performance statistics:

Cumulative Return of TOS: 5.7861
Cumulative Return of Benchmark : 0.012299999999999978

Standard Deviation of TOS: 0.004547823197908003
Standard Deviation of Benchmark : 0.017004366271213763

Average Daily Return of TOS: 0.0038167861508578197
Average Daily Return of Benchmark : 0.00016808697819094035