Project 6: Indicator Evaluation

Saad Rasool
srasool7@gatech.edu

Abstract**—**The research below represents 2 main concepts: theoretical optimal trading strategy and implementation of five stock indicators. Understanding these two concepts will helped us what key parameters a machine learning algorithm could use as an input and what maximum possible gains it could achieve.

Introduction

For the optimal strategy, we were able to peak in the future to analyze the stock price and make a decision to maximize returns. For indicator implementation we have to research several indicators to understand which ones could potentially be most useful in a machine learning algorithm. In the later sections we will go deeper into each indicator and implementation of optimal trading strategy. In this report all the data represented is from 'JPM" stock from 2008-01-01 until 2009-12-31 period. All the code for implemented using python programming language.

Optimal Trading Strategy:

In this section we will walk through implementation of the optimal trading strategy and its performance.

<u>Creation and Assumption Used:</u>

The main assumption we made to implement or create an optimal trading strategy was that we could analyze the price of the stock from the future. This means that we could essentially see the price of the stock in the future, which for obvious reason is not possible in reality. Without making this assumption we would not be able to implement the optimal trading strategy.

To create the strategy we implemented the testPolicy(symbol, sd, ed, sv) function which take following arguments:

- list of stock = ['JPM']

- sd = datetime start date

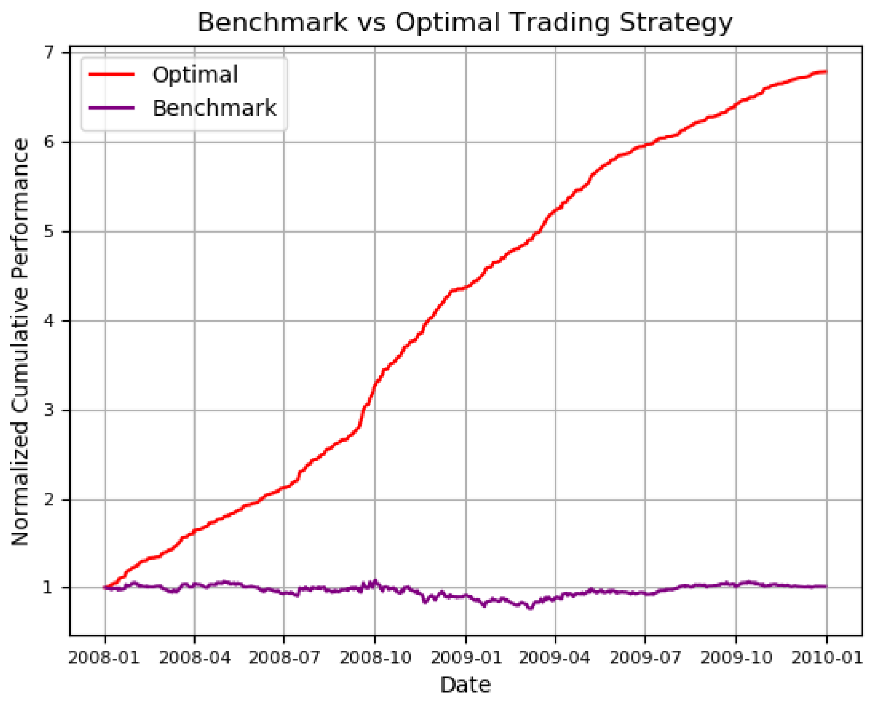
-ed = datetime end data

- sv = Initial cash

To obtain the data of the stock, testPolicy function calls the getdata function implemented in the lectures. Once we have the data of the stock for the time ranges we are trying to create the strategy for we can dive into how the function actually buys and sells the stock to maximize returns. To code the strategy first we need to understand if we have all the future prices how would we actually or in reality would trade. After looking at the price graph it makes most sense to trade maximum number of stocks everyday. As we can see the price 1 day ahead we know if the stock would either go up , down or stay the same. If we take advantage of change in the price of the stock daily we would have the maximum capital. The following trades were essentially implemented:

```
- if the next day price > today price:
        then buy the stock , with max number of shares
- while buy = true
        if the next day < then today's price:
                sell the stock, this would give us the max return on that trade
- if the next day price < today price:
        then short the stock ( buy negative shares)
- while shorting = True
        if the next day price > today price:
                sell the short buy, this will give us the maximum return on the shorting of the
                stock
```

the sudo code described above essentially will give you the maximum return as we are taking advantage of every single price change. The code generates an orders_df for each timestamp, with is used in the marketsimulator code to calculate cumulative returns. Below is the plot showing cumulative return for the optimal strategy when compared with benchmark (holding 1000 shares through start until the end of the period) As we can see the strategy works and our cumulative returns are significantly higher.



To compare the performance of the strategy we also generated the following performance metrics.

Metrics	Portfolio	Benchmarks
Cumulative return $	678,610	101,230
Mean daily Return	0.003817	0.00017
Stdev Daily Return	0.004548	0.017004

As you can see from the above table, the cumulative returns are higher for the portfolio, along with the mean daily return. The standard dev is lower meaning there was less variance in the returns compared to benchmark.

To run the code please use the following cmd:

```
PYTHONPATH=../:. python testproject.py
```

Indicators:

Indicators 1 : Golden and Death Cross

Golden and death cross are both indicator of long term bull or bear market, both indicator use a 50-day moving and 200-day moving average of the stock price. As the short term moving average moves faster than the long term moving average. In our case 50 day moving average represents short and 200 day moving average long term.

Golden Cross occurs when the 50 day moving average is initially lower than the 200 day average and once the 50 day moving average cross the 200 day moving average causing it to be higher is the point referred as golden cross. In our case we are using golden cross as a buy indicator.

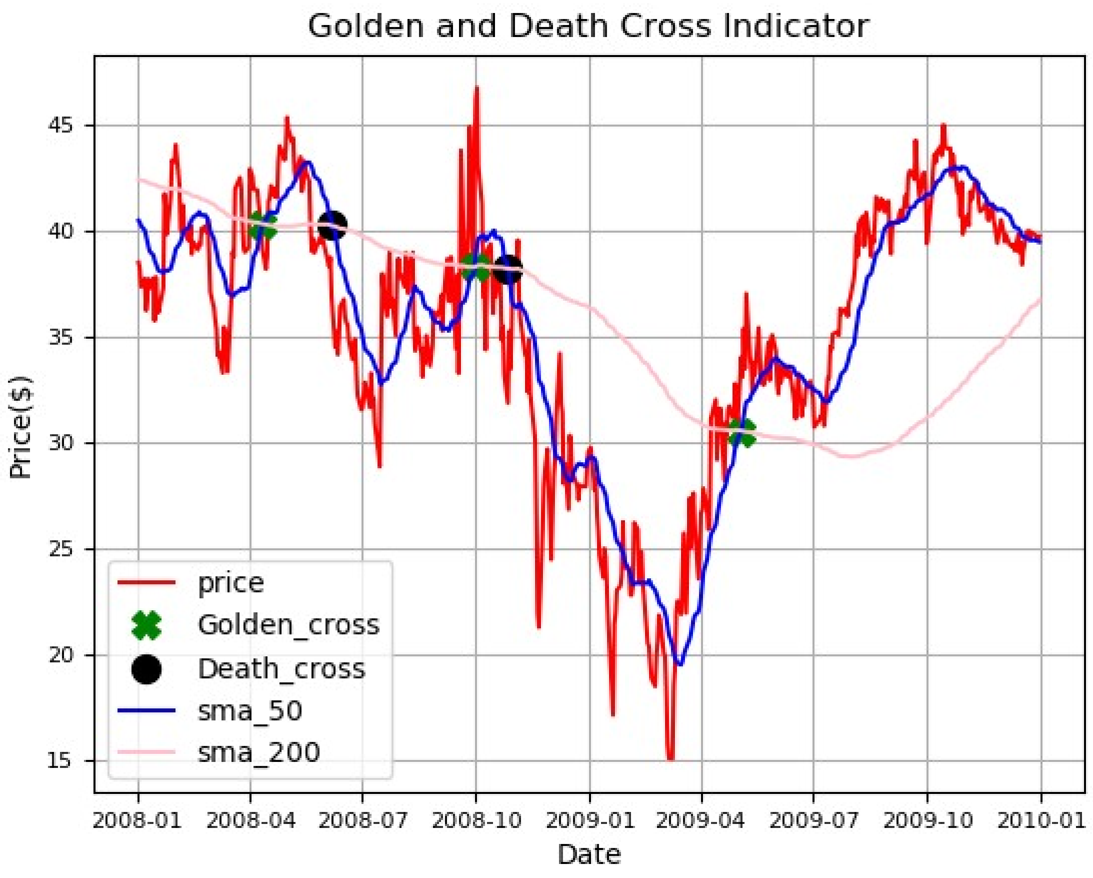
On the other hand if the 50 day moving average is initially higher than 200 day moving average and the 50 day moving average cross the 200 day moving average in a downward trend then the cross of the two lines are refereed as death cross. In our case any occurrence of death cross represents a sell indicator

The golden and death cross are exact opposites, a golden cross indicates a long-term bull market going forward where as a death cross refers to as a long term bear market, hence the buy and sell signal is achieved

In pseudo code terms:

```
if current_50_day_moving_avg < current_200_day_moving_avg:
    if next_day_50_day_moving_avg > next_day_200_day_moving_avg:
        golden_cross = True
if current_50_day_moving_avg > current_200_day_moving_avg:
    if next_day_50_day_moving_avg < next_day_200_day_moving_avg:
        death_cross = True
elif:
    there is no signal
```

The graph below indicated the buy and sell signals for the 'JPM' stock using the golden_death_cross indicator implemented.



Indicators 2: Rsi

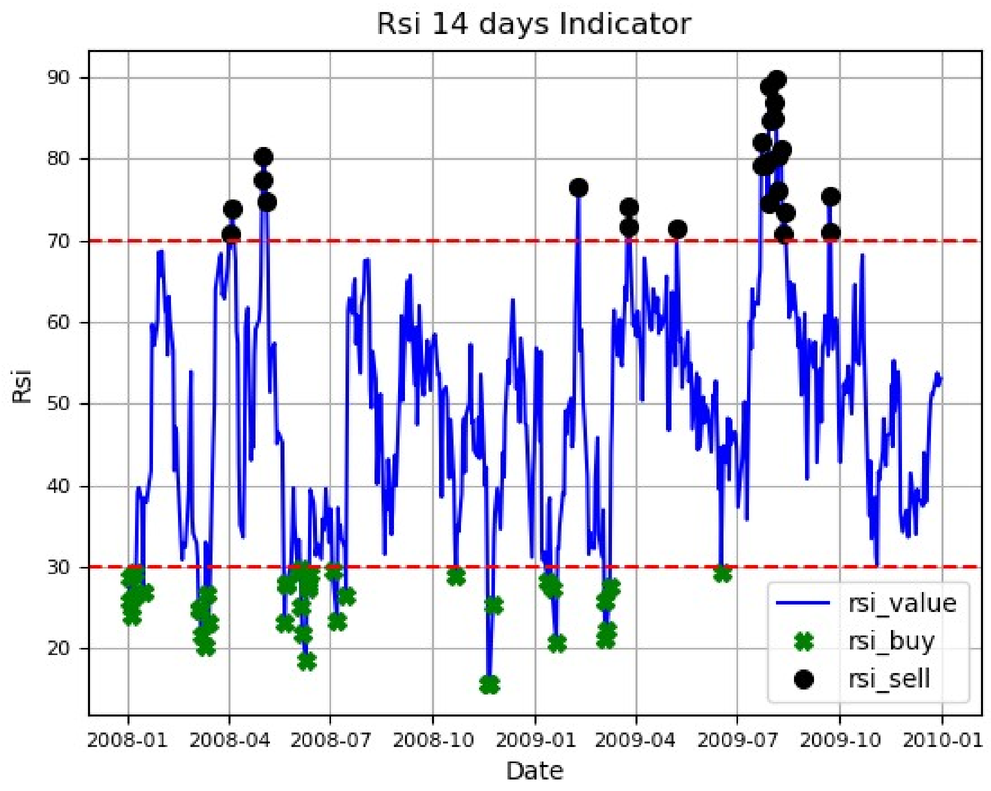
Rsi (relative strength index) is a momentum indicator, which measures the speed and magnitude of price changes to evaluate if the stock is overvalued or undervalued. Based on the oversold or the overbought of the stock, we can generate buy or sell signals. Rsi does require a period of window for its calculations, in our case we used 14 day period. As we are using a smaller period of 14 days, this would mean that it is a short term buy or sell signal.

An rsi value can range from 0 – 100, if the rsi value crosses 70 it can indicate an overbought stock, hence a sell signal can be achieved. On the other hand if the rsi value is lower than 30, it could indicate an oversold or undervalued condition hence buy signal is achieved.

Below is the pseudo code used to generate the rsi value

```
calculate the prices difference = price.diff()
calculate  up_bound and lower_bound = price_change.clip()
calculate the sma_up and sma_down via using a rolling().mean function
rsi = sma_up / sma_down
rsi = 100 – 100 /(1 + rsi)
```

below is the rsi graph along with the buy and sell signals



Indicators 3: Ema Crossover

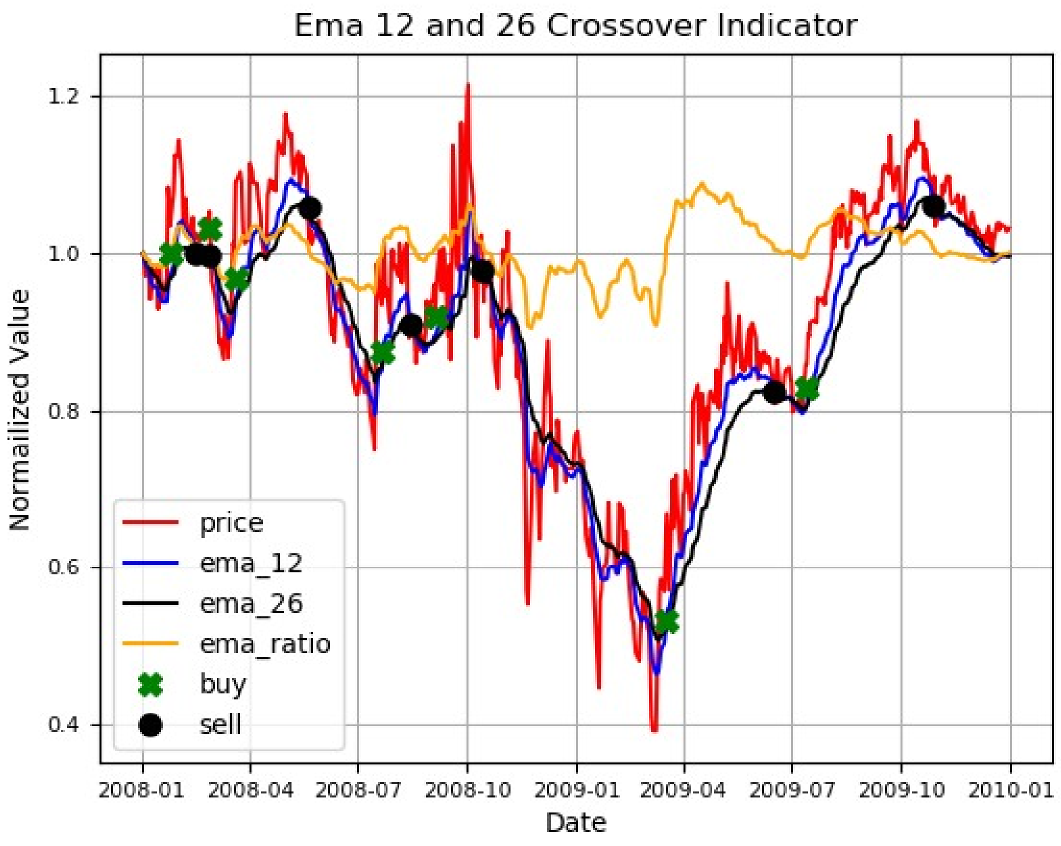
Ema (exponential moving average) crossover indicator is very similar to golden and death cross indicator. The main idea behind using ema is for mover short term trading strategy. For our ema indicator we are using ema of 12 and 26 days. Ema is mostly used as it puts more weight to recent price changes hence the response is much faster then sma (simple moving averages)

We can generate buy or sell signals when the 12 day ema cross the 26 day ema line. If the 12 day ema is higher than 26 day ema and cross over to be lower then that would be a sell signal. On the other hand is the 12 day ema is lower than 26 day ema , and crosses over to be higher then that would be a buy signal.

Ema ratio was also calculated = ema_12/ema_26, if > 1 means ema_12 is greater than ema_26, if < 1 it mean ema_26 is greater than ema_12.

In pseudo code:

```
ema_12 = stock_price.ewm(window = 12).mean()

ema_26 =stock_price.ewm(window =26).mean()

if current_12_day_ema < current_26_day_ema:

    if next_day_12_day_ema > next_day_26_day_ema:

        buy = True

if current_12_day_ema > current_26_day_ema:

    if next_day_12_day_ema < next_day_26_day_ema:

        sell = True

elif:

    there is no signal
```



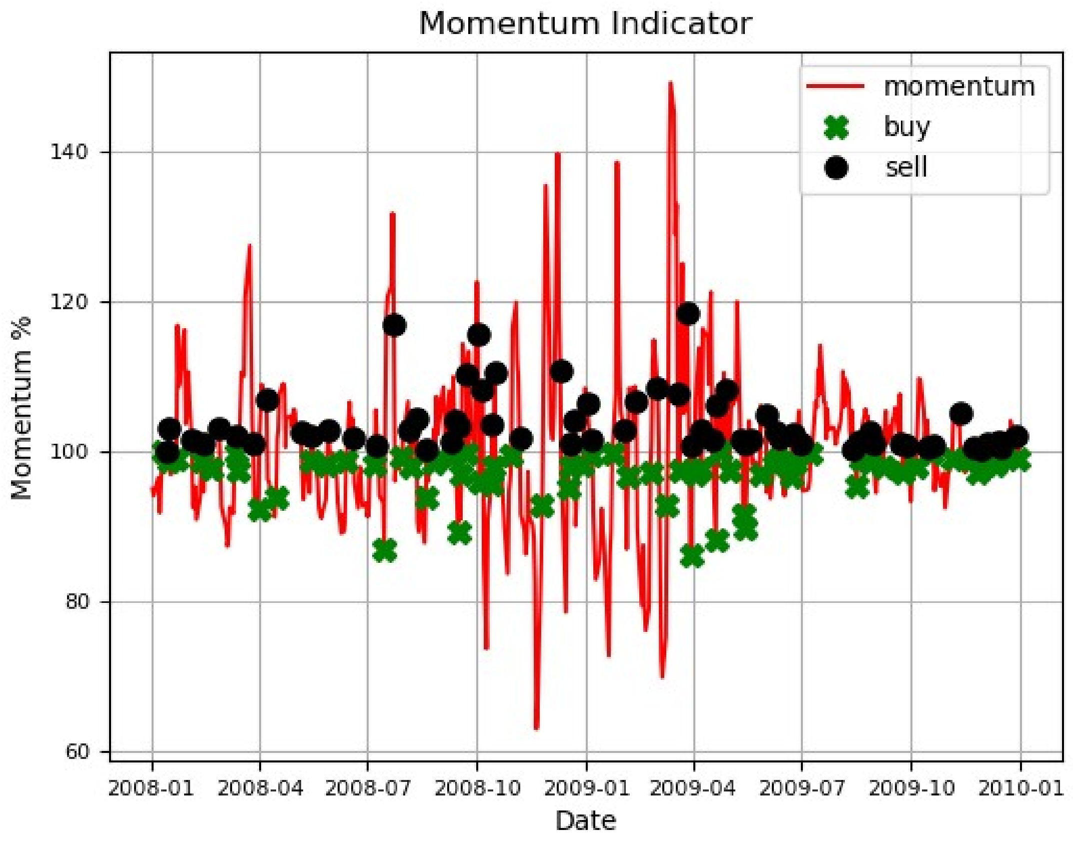
Indicators 4: Momentum

Momentum indicator demonstrates the speed at which is the price is changing, the indicator is used for more short term trades. A buy signal can be generated as the momentum of the prices is increasing and sell indicator can be generating is the momentum of the stock is decreasing. The indicator, should be used with other trading indicator. Using this indicator is like following others and what the market is doing. I think including this indicator with others mentioned above could significantly help develop a reasonable trading strategy. Momentum does require a window, in our case we are using 5 days as I wanted to include some short term trading strategies.

Momentum = (current price / price – 5 days) * 100

```
    if current_momentum  > 100:
        if next_momentum < 100: #momentum is shifting in negative direction
            Sell = True
    elif: current_momentum < 100:
        if next_momentum > 100 # momentum is shifting in positive direction
            buy = True
```

Below is the chart for 'JPM' stock



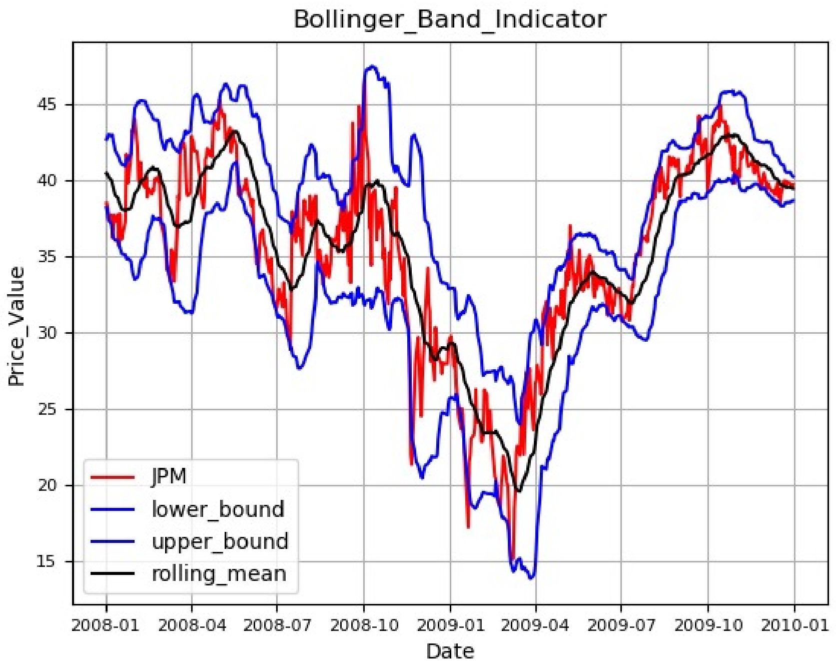
Indicators 5: Bollinger bands percent

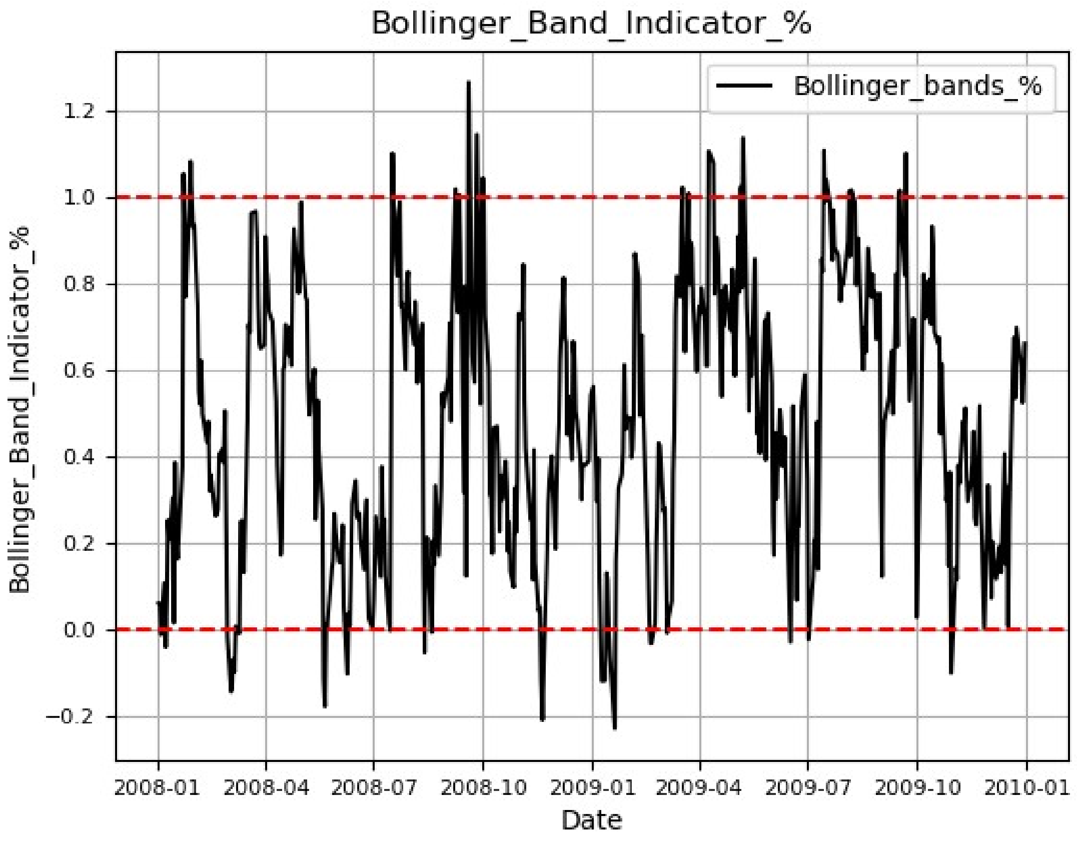
Bollinger bands is a long term trading strategy, it consists of a simple moving avg line, along with the upper and lower bound. The indicator does give us buy and sell signals. If the price of the stock continually touches the upper bound, it could mean the stock is overvalued and should be sold. On the other hand if the price touches the lower bound it could indicate that the stock is undervalued and could be a buy signal. Bollinger bands along with some of the short term trading indicators mentioned above, could produce reasonable results to maximize returns. Upper and lower bands are calculated based on the volatility of the stock price, via using standard deviation.

pseudo code:

```
sma_rolling = price.rolling(window = 20 days).mean()

std_rolling = price.rolling(window = 20 days).std()

upper_bound =  sma_rolling + std_rolling *2

lower_bound = sma_rolling - std_rolling *2

bollinger% = (price – lower_band) /( upper_band – lower_band)

if bollinger% >= 1 = buy

If bollinger% <= 0 = sell
```

below are the charts for bollinger band and bollinger band pct





Conclusion:

In conclusion, the indicators mentioned above, can help with short and long term strategy for trading. Implementing them together should definitely provide us with reasonable and profitable trades. It should be super exciting to see how the above indicators would perform in a machine learning algorithm.

Reference:

"Investopedia." Investopedia, 12 March 2023, www.investopedia.com.