Bolling for Prophet

Combine FB Prophet with Bollinger Band strategy for Microsoft

Overview

Would combining an fbProphet model with a Bollinger Mean Reversion (BMR) and Bollinger Breakout Strategy be a viable combination for a short term trading strategy?

Process:

Gather and clean data.

Perform simple financial analysis on it - establish a baseline “buy and hold” strategy to compare to.

Generate fbProphet prediction.

Backtest BMR strategy on the historical data set.

Generate buy-sell signals based on Prophet prediction

Final Analysis - how does BMR compare to baseline? Does Prophet prediction improve outcomes?

Data cleaning and initial analysis

Step 1: imports, functions, constants.

Step 2: get data

```
ticker='MSFT'

# Get current closing prices for ticker
stock_df = fetch_yahoo(ticker, start_date, end_date)

# Display sample data
stock_df
```

```
Start date = 2020-05-01
End date = 2022-05-01
```

	high	low	close
Date			
2020-05-01	175.006481	170.470650	171.019272
2020-05-04	175.359157	170.264928	175.202408
2020-05-05	179.914560	176.240835	177.083344
2020-05-06	180.453403	177.935684	178.827164
2020-05-07	180.796305	178.866373	179.865631

Step 3:

Validate data is clean.

Save for use by ML model.

```
validate_data(stock_df)
```

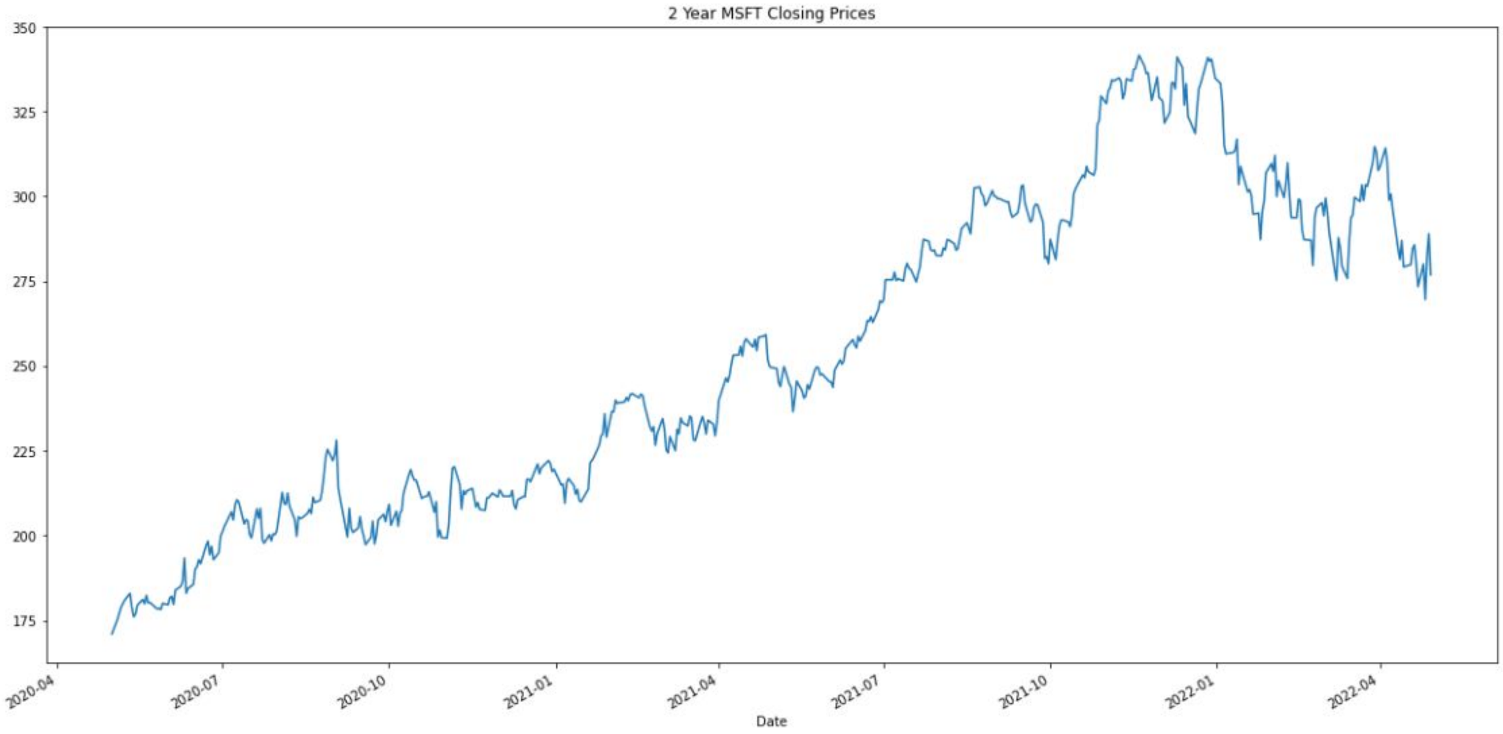
```
DATA TYPES:
high     float64
low      float64
close    float64
dtype: object

Total duplicated values: 0

Total null values:
high     0
low      0
close    0
dtype: int64
```

```
# Save a copy of the clean, raw data
stock_df.to_csv('Resources/msft_hlc.csv')
```

Step 4:

Initial examination of data

Visualize



```python
# Calculate the cumulative returns
cumulative_returns = (1 + stock_df["daily_return"]).cumprod()
cumulative_returns[-1]
```

```
1.6189703012664525
```

The cumulative product of 1.6189703 means that if we bought $100,000 worth of MSFT stock at the beginning of our period, we'd have $161,897.03 at the end.

Variance ¶

Variance measures the price volatility of a single asset around its mean over time.

```python
daily_var = stock_df["daily_return"].var()
daily_var
```

```
0.0002807718490664985
```

A variance of .00028 is low and therefore relatively stable.

Sharpe Ratio

The Sharpe ratio assesses risk and reward by measuring the excess return (that is, the reward, or profit) for the risk that someone assumes when investing in the asset. The greater the value of the Sharpe ratio, the more attractive the risk-adjusted return for that asset is.

```
# Calculate Sharpe Ratio by dividing the average annual return by the annual standard deviation
sharpe_ratio = average_annual_return / annual_std_dev
# Display result
sharpe_ratio
```

```
1.040869408505323
```

```python
ticker='SNP'

# Get current closing prices for M
sp_df = fetch_yahoo(ticker, start_date, end_date)

# Calculate daily returns
sp_daily = sp_df['close'].pct_change().dropna()
```

```python
sp_cum_return = (1 + sp_daily).cumprod()
sp_cum_return[-1]
```

```
1.1994573401673354
```

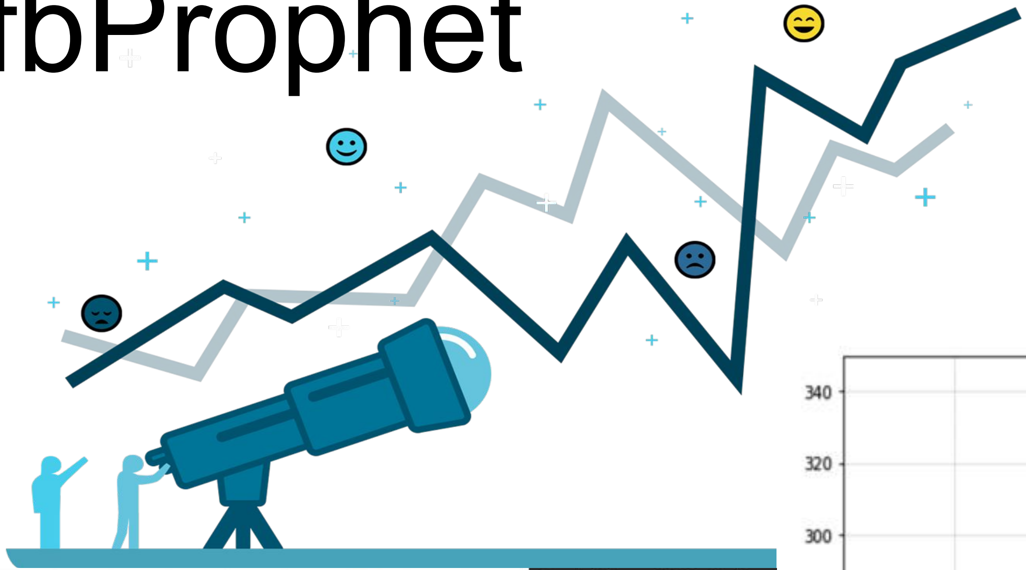
The cumulative return for the S&P500 is 1.199457, meaning that if we invested $100,000 initially, we'd have $119,945.72.

```python
# Calculate beta of all daily returns
# Divide the covariance versus the S&P 500 by the variance of the S&P 500
beta = stock_df["daily_return"].cov(sp_daily) / sp_daily.var()

# Display the beta
beta
```

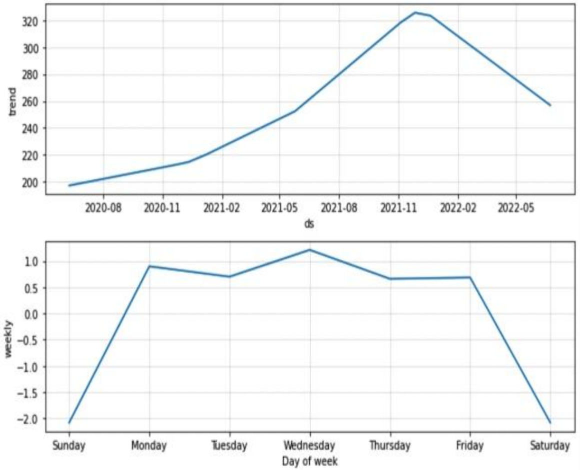
```
0.07116721674627714
```

The beta measures how much an asset's return value is likely to change relative to changes in the overall market's return value. MSFT's beta of .07 means that it will move significantly less than the market.

fbProphet



- ➔ **What is it?**
- ➔ **Components?**
- ➔ **Usefulness?**

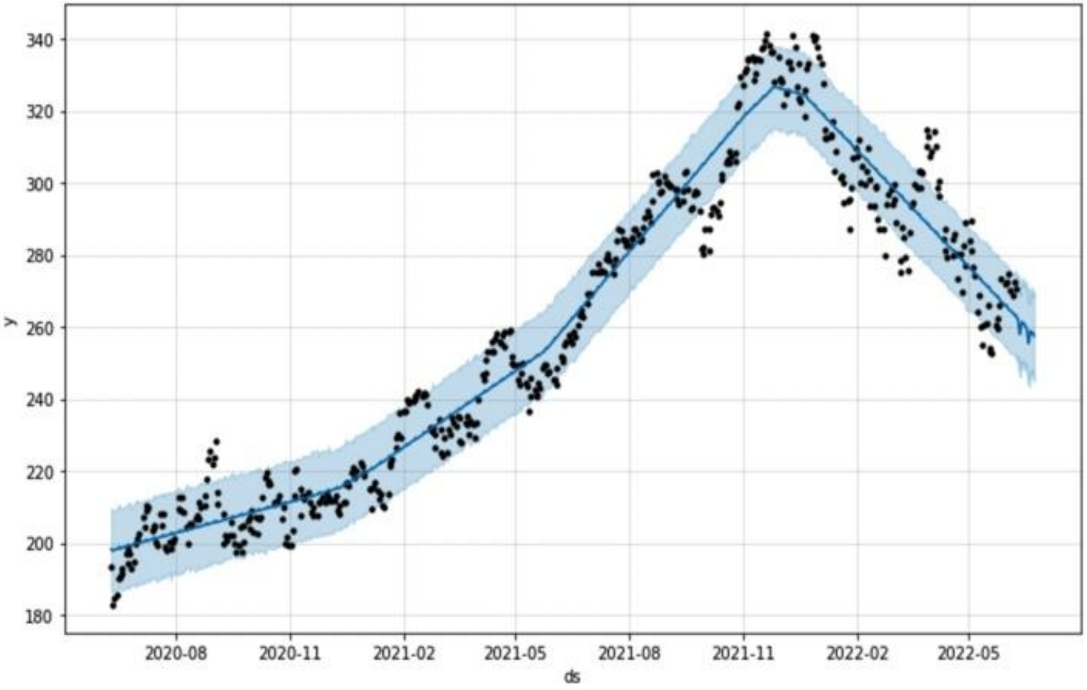



```
Start date = 2020-06-10
End date = 2022-06-09
```

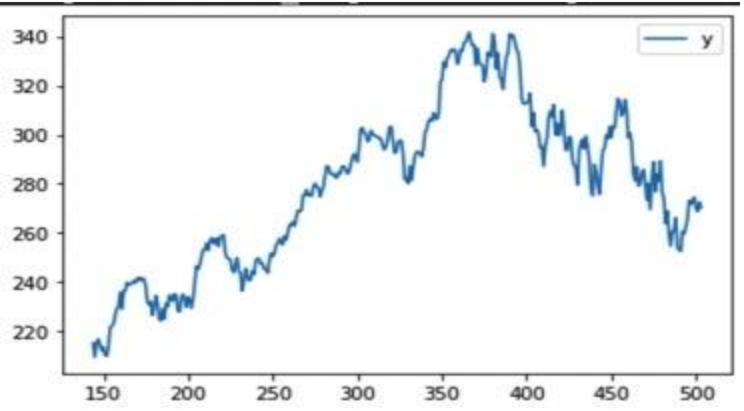
	ds	y
0	2020-06-10	193.373337
1	2020-06-11	182.989532
2	2020-06-12	184.433640
3	2020-06-15	185.612488
4	2020-06-16	190.160950

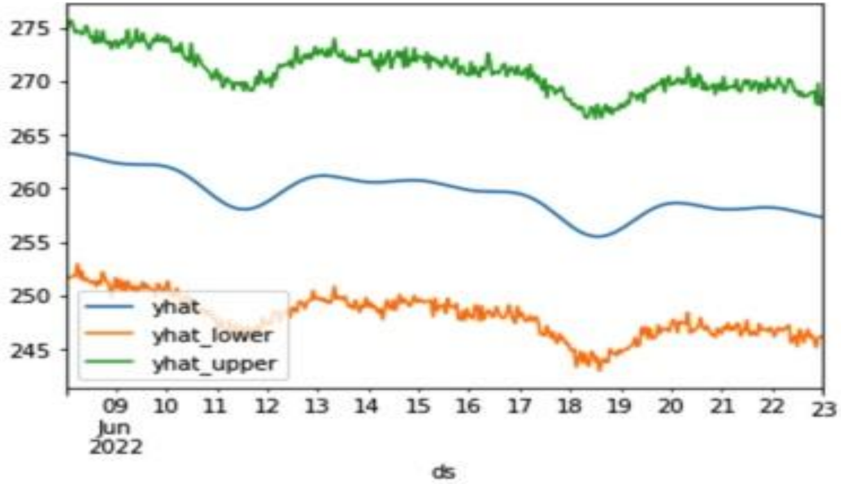
	ds	y
499	2022-06-02	274.579987
500	2022-06-03	270.019989
501	2022-06-06	268.750000
502	2022-06-07	272.500000
503	2022-06-08	270.410004

```
(504, 2)
```



Two Week vs Thirty Day Results



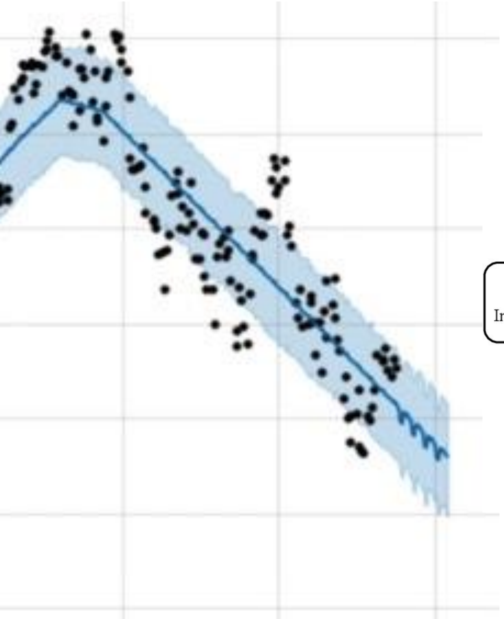


ds	Best Case	Worst Case	Most Likely Case
2022-06-08 01:00:00	274.815099	251.645973	263.288875
2022-06-08 02:00:00	275.544014	251.640658	263.271218
2022-06-08 03:00:00	275.693323	251.767520	263.249580
2022-06-08 04:00:00	275.125883	251.810190	263.224048
2022-06-08 05:00:00	274.543673	251.900335	263.194765
...	...	...	...
2022-06-22 20:00:00	269.065374	245.723783	257.444881
2022-06-22 21:00:00	267.870058	245.183337	257.405022
2022-06-22 22:00:00	269.818030	246.122557	257.367933
2022-06-22 23:00:00	267.754686	246.115556	257.333861
2022-06-23 00:00:00	268.455372	246.225405	257.302997

360 rows × 3 columns

```
Best Case           270.927778
Worst Case          248.019668
Most Likely Case    259.469915
dtype: float64
```

Thirty Day Forecast



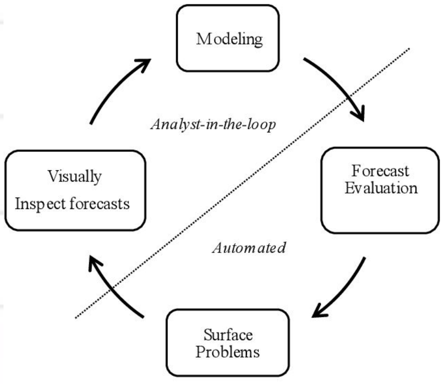
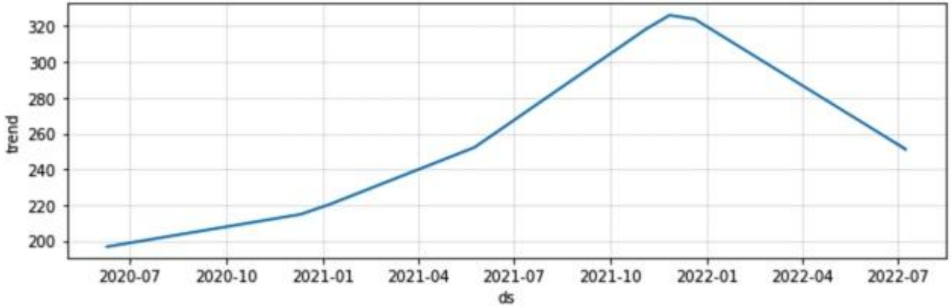
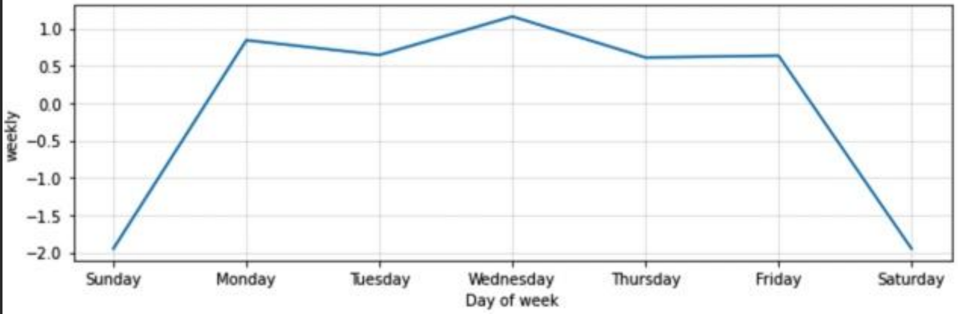


Fig. 1: Prophet workflow

```
Best Case          283.247430
Worst Case         260.414963
Most Likely Case   271.824712
dtype: float64
```



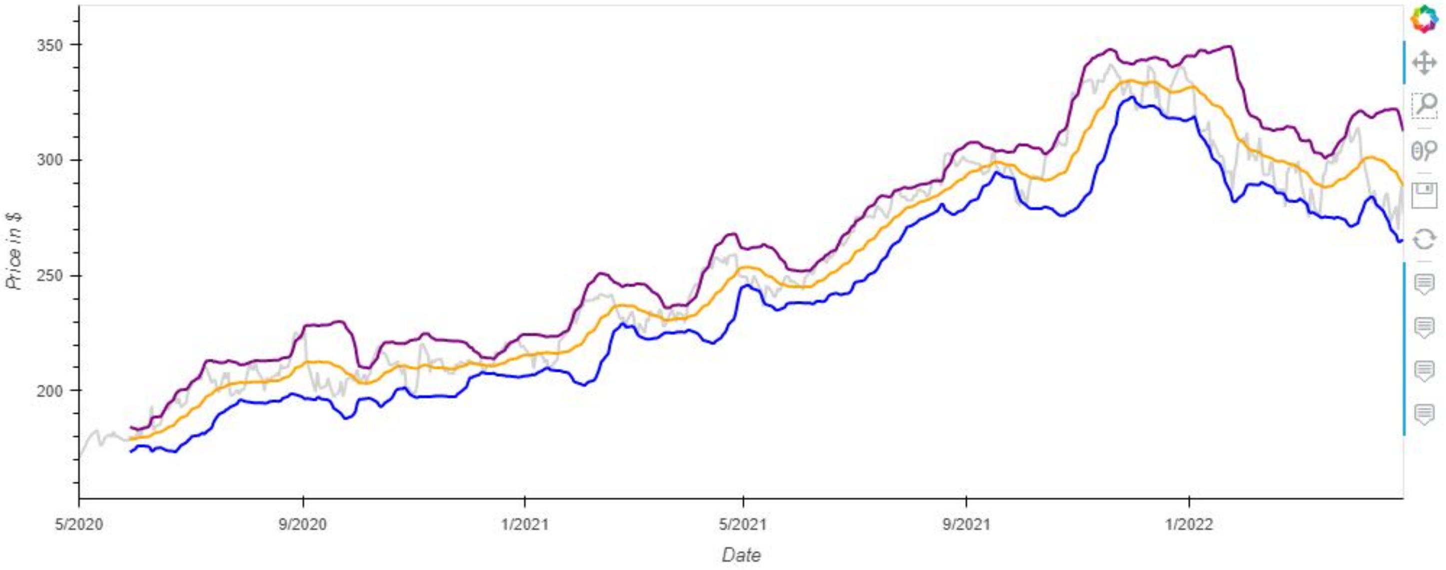


ds	Best Case	Worst Case	Most Likely Case
2022-06-09	274.045364	250.667229	262.390434
2022-06-10	273.694428	250.630918	262.052140
2022-06-11	271.999078	247.811082	259.102425
2022-06-12	269.818604	247.145408	258.739036

ds	Best Case	Worst Case	Most Likely Case
2022-07-04	264.477557	242.380305	253.537512
2022-07-05	263.449823	241.182433	252.976118
2022-07-06	264.435202	241.237202	253.127651
2022-07-07	263.995076	240.184904	252.215559

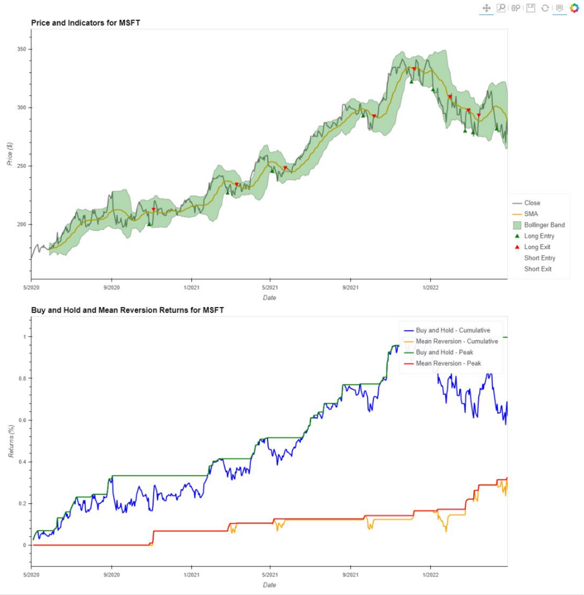
Bollinger Bands

A Bollinger Bands is a technical analysis tool defined by a set of trendlines plotted two standard deviations (positively and negatively) away from a simple moving average (SMA) of a security's price, but which can be adjusted to user preferences. They were designed to discover opportunities that give investors a higher probability of properly identifying when an asset is oversold or overbought.



Bollinger-Mean Reversion Algorithmic Strategy

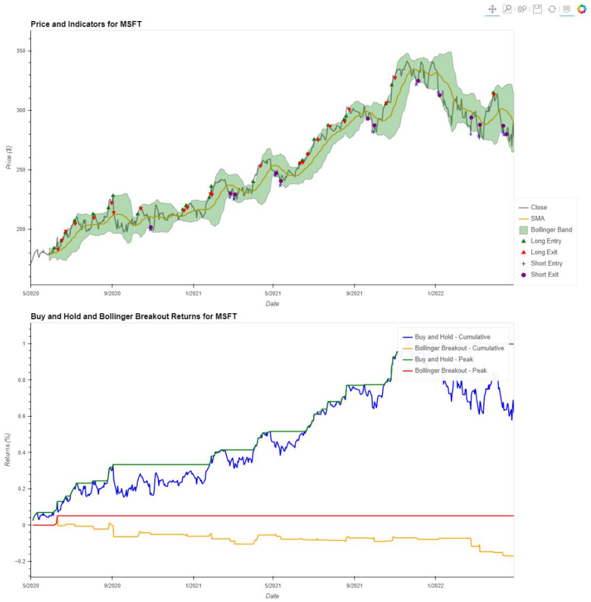
We apply the Bollinger Breakout algorithmic strategy, with Shorting (take Short position on the price reaching upper band)



	tot_returns	annual_returns	annual_volatility	sortino_ratio	sharpe_ratio
Buy and Hold	0.61897	0.272997	0.266358	1.334663	0.949838
Mean Reversion	0.26785	0.126254	0.134898	0.563245	0.787660

Bollinger Breakout Algorithmic strategy

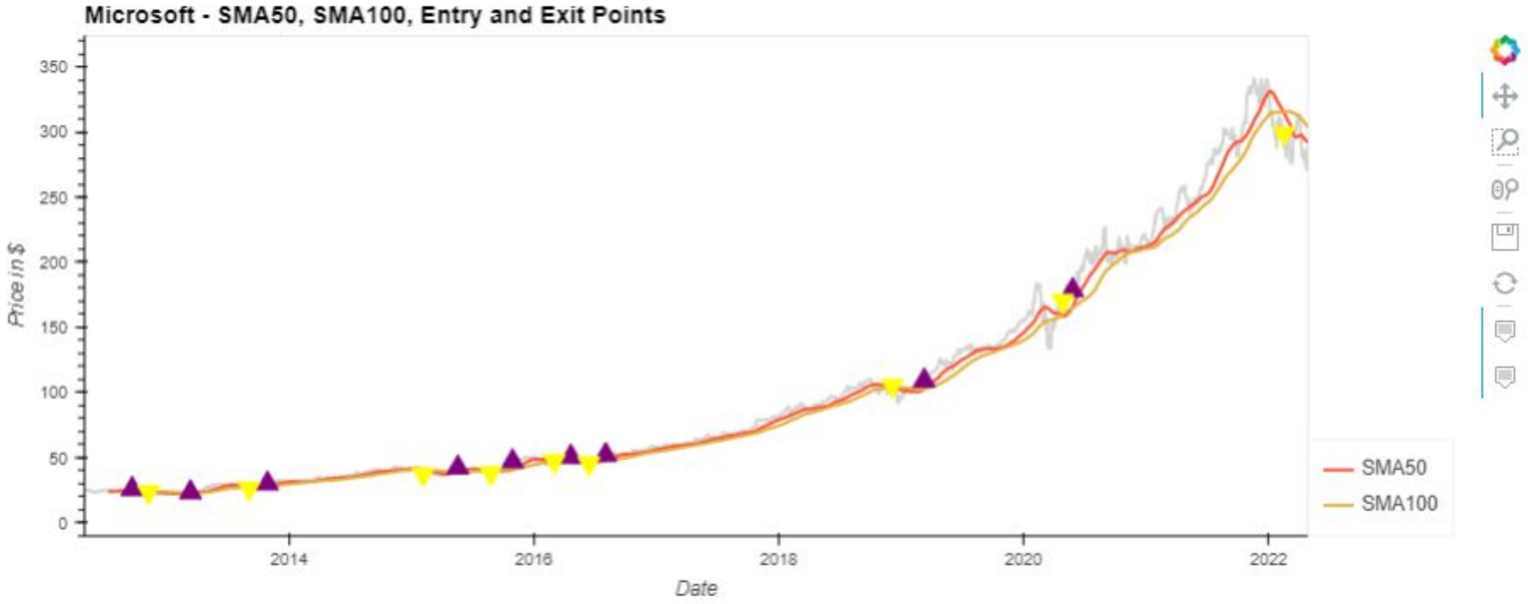
We apply the Bollinger Breakout algorithmic strategy, with Shorting (take Short position on the price reaching lower band)



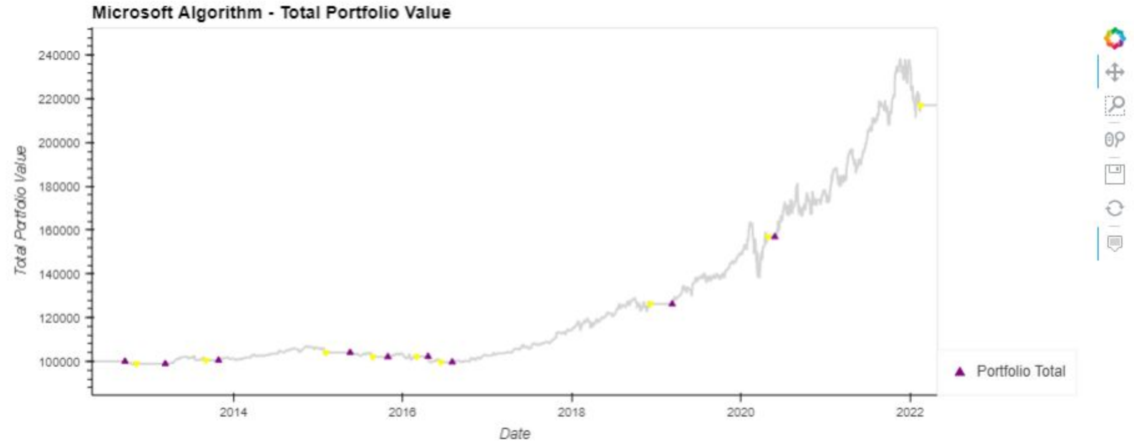
	tot_returns	annual_returns	annual_volatility	sortino_ratio	sharpe_ratio
Buy and Hold	0.618970	0.272997	0.266358	1.334663	0.949838
Bollinger Breakout	-0.169307	-0.088744	0.104089	-0.417148	-1.044728

Backtest the Algorithm

Using 10 years of historic data



Portfolio Evaluation



	Backtest
Annualized Return	0.082913
Cumulative Returns	1.169931
Annual Volatility	0.102686
Sharpe Ratio	0.807436
Sortino Ratio	1.166911

Trade Evaluation Metrics

	Stock	Entry Date	Exit Date	Shares	Entry Share Price	Exit Share Price	Entry Portfolio Holding	Exit Portfolio Holding	Profit/Loss
0	MSFT	2012-09-20	2012-11-08	500.0	25.848677	23.678865	12924.338341	11839.432716	-1084.905624
1	MSFT	2013-03-13	2013-09-04	500.0	23.327494	26.435738	11663.746834	13217.868805	1554.121971
2	MSFT	2013-10-29	2015-02-05	500.0	30.096069	37.232998	15048.034668	18616.498947	3568.464279
3	MSFT	2015-05-20	2015-08-26	500.0	42.302677	38.223240	21151.338577	19111.619949	-2039.718628
4	MSFT	2015-10-30	2016-03-03	500.0	47.110085	47.504986	23555.042267	23752.492905	197.450638
5	MSFT	2016-04-21	2016-06-14	500.0	50.617527	45.534485	25308.763504	22767.242432	-2541.521072
6	MSFT	2016-08-03	2018-12-06	500.0	52.058990	105.123352	26029.495239	52561.676025	26532.180786
7	MSFT	2019-03-11	2020-04-27	500.0	109.091713	170.509857	54545.856476	85254.928589	30709.072113
8	MSFT	2020-05-27	2022-02-16	500.0	178.608063	298.804047	89304.031372	149402.023315	60097.991943

Analysis

- Testing different algorithms/strategies

- Using an API that collects more data (more time series)

- Will data with higher volume of trade signals be more profitable?


Q&A