



Algorithmic Trading

With Machine Learning

By Hunter Spence, Andre Montgomery, Nicholas Pamboukis



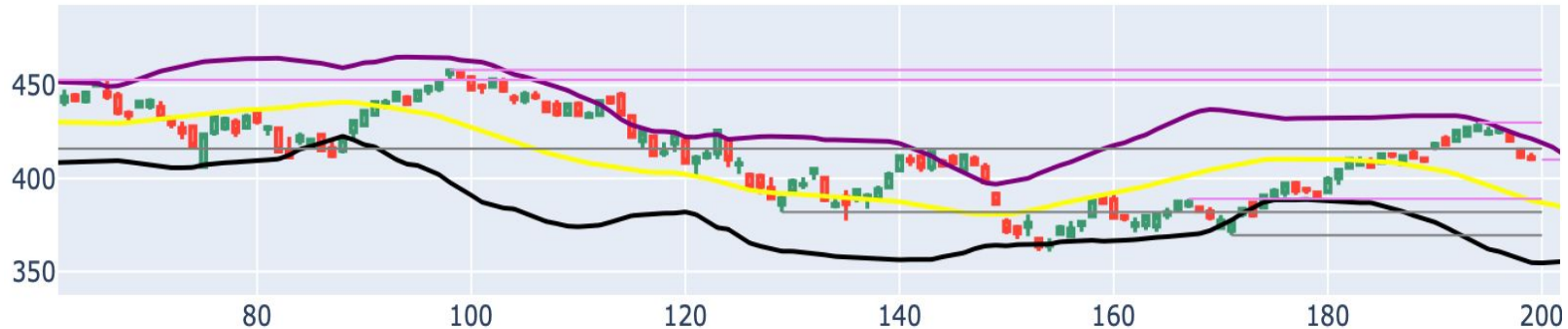
Background

Algorithmic trading is a complex trading tool. In this project, we have built a trading program that allows the user to get started in Algo trading. The only thing the user needs to know, is what stock do you want to trade. Since the program uses an API (yfinance), the program can pull in any data for the stock you choose.

To better understand our model, we will use the SPY data.

Trading Strategy (Bollinger Bands)

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”(). Approximately 90% of the price action of a stock occurs between the two bands. Our strategy takes Short and Long positions, based on where the price is in the Bollinger Band. Below you can see the Bands and SMA of the SPY. Including Support and Resistance levels.<https://www.investopedia.com/terms/b/bollingerbands.asp>



Trading Strategy (Code)

```
# Bollinger Bands Trading Strategy

# Set SMA and standard deviations
SMA = 30
dev = 2

# Creating column to hold SMA
data["SMA"] = data["Close"].rolling(SMA).mean()

# Creating the lower band
data["Lower"] = data["SMA"] - data["Close"].rolling(SMA).std() * dev

# Creating the upper band
data["Upper"] = data["SMA"] + data["Close"].rolling(SMA).std() * dev

# Creating a column for distance between closing price and the SMA
data["distance"] = data.Close - data.SMA # helper Column

# Creating a column to add a 1 if oversold to go long
data["position"] = np.where(data.Close < data.Lower, 1, np.nan) # 1. oversold -> go long

# Adding a -1 to the 'position' row if overbought to go short
data["position"] = np.where(data.Close > data.Upper, -1, data["position"]) # 2. overbought -> go short

# Adding a 0 to the 'position' column to close position when passes SMA
data["position"] = np.where(data.distance * data.distance.shift(1) < 0, 0, data["position"])

# If 1-3 isn't met the previous position is held
data["position"] = data.position.ffill().fillna(0) # where 1-3 isn't applicable -> hold previous position

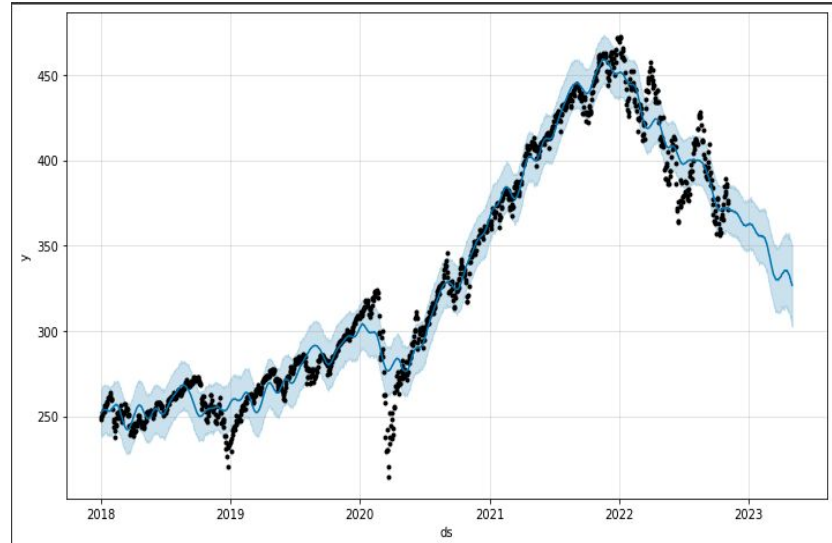
# Creating a column for strategy returns
data["strategy_returns"] = data.position.shift(1) * data["daily_returns"]
```

Fb Prophet Model

FB Prophet is a machine learning model that can make predictions of a stocks price. In our setup, we have the model predict the next 180 days. This is changeable, but it helps identify the trends over the next 6 months.

The Cumulative Sum of projected Daily Percent Changes, over a 180 day period, is:

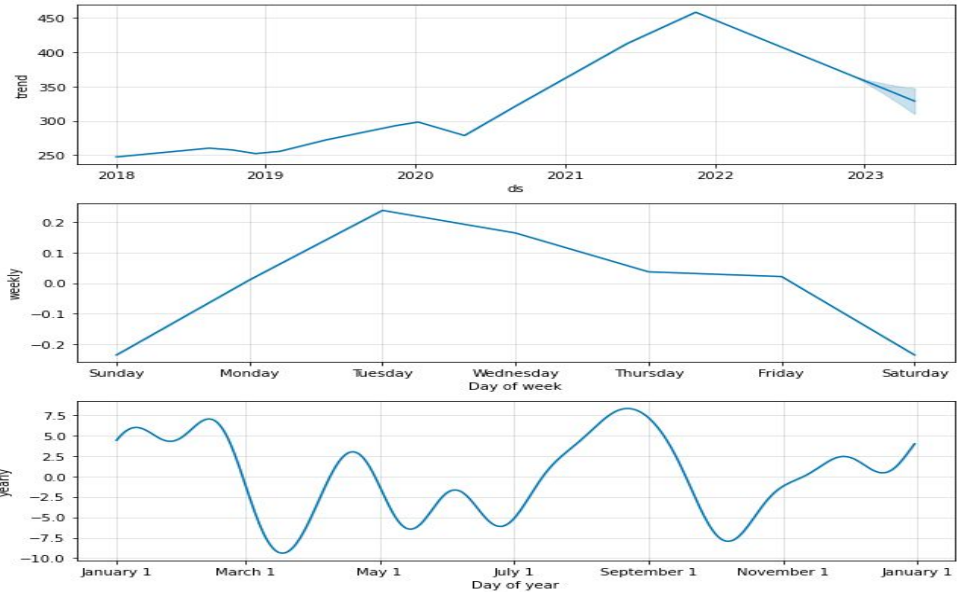
% -12.739



Fb Prophet Model - Trends

The model also gives a analysis of the trends over a daily and monthly time-frame. This is important in understanding how the price of the stock moves during time periods.

For example, SPY performs the best during Sunday - Tuesdays, and during the months of July - September.



Trading Strategy Performance

The graph on the right, displays the strategies returns vs a buy and hold strategy. During 2022-09-19 : 2022-11-04, our trading strategy performed better than a buy and hold strategy.

*The green is our strategy returns, while the blue is the SPY price.





Trading Strategy Performance

During the time period, if the investor would've invested \$100k in the SPY, they would've made **\$7968.92**. A total profit percentage gain of **7%**. In 53 days. This is all without including leverage into our model. Depending on the leverage size, the profits could be higher.

Profit gained from the BB strategy by investing \$100k in SPY : 7968.92
Profit percentage of the BB strategy : 7%

SKlearn SVM Classification Report

The Classification Report indicates that the **training model** was accurate at predicting. The F1 score was 77-81. The Zero Loss score was: 0.207

(Top)

The Classification report indicates the **testing model** was less accurate at predicting. The F1 score was 46-53. The Zero Loss score was: 0.516

(Bottom)

	precision	recall	f1-score	support
-1.0	0.80	0.82	0.81	239
0.0	0.74	0.81	0.77	340
1.0	0.90	0.74	0.81	202
accuracy			0.79	781
macro avg	0.81	0.79	0.80	781
weighted avg	0.80	0.79	0.79	781

	precision	recall	f1-score	support
-1.0	0.52	0.54	0.53	1423
0.0	0.61	0.38	0.47	2130
1.0	0.37	0.59	0.46	1287
accuracy			0.48	4840
macro avg	0.50	0.50	0.49	4840
weighted avg	0.52	0.48	0.48	4840



SKlearn Machine Learning Model

The Machine Model we used was SKlearn SVM Classifier. The model produced these results:

From 09-26-2022 – 09-30-2022, the model Profit Percentage was 3%. With a return of \$3,379.71 if invested \$100,000. Without leverage

Sharpe Ratio : 2.48

```
Profit gained from the ML strategy by investing $100k in SPY : 3379.71
Profit percentage of ML strategy : 3%
```



Future Improvements

Trading Features

- Add Resistance and Support levels into Strategy
- Develop better ML model for strategy data.
- Incorporate automatic buy/sell into trading platform (Alpaca, etc.)
- Add fb Prophet predictions into analysis

This project is in it's developing stages. As the project continues to be developed, we want to add more features to the program. Including an easier UI to interact with. Our trading model also has room for improvement and optimization. Eventually the program should be able to screen/trade automatically on multiple stocks.