CSCI X170 Final Project

UGA Computational Investing
By Nathan Brooks

An Extension of Project 4 - A Market Simulator

Project 4 overview:

- Tasked with reading premade order books and using historical data to calculate profit and loss

Project extension:

- Extend Project 4 to allow users to generate order books based on custom order strategies.

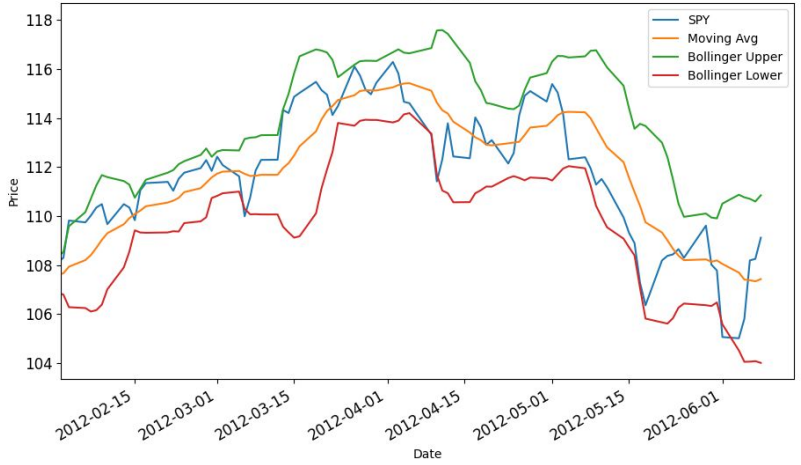
What has been added?

Indicators:

- I have added a folder where users can create and add their own indicators
- Indicators may include parameters to tune them to each use case
- Base package includes an RSI and Bollinger Bands Indicator

Indicator example

```python
def bollinger_bands_indicator(portvals, name, window=20):
    # add bollinger indicator details
    # add a moving average, upper bollinger band and lower bolinger band to portvals
    portvals['Moving Avg'] = portvals[name].rolling(window).mean()
    portvals['Bollinger Upper'] = portvals[name].rolling(window).mean() + (2* portvals[name].rolling(window).std())
    portvals['Bollinger Lower'] = portvals[name].rolling(window).mean() - (2* portvals[name].rolling(window).std())
    return portvals
```



Bollinger bands indicator

- function output
- graph output

What has been added? (continued)

Strategies:

- I have added a folder where users can create and add their own strategies
- Strategies may include parameters to tune them to certain use cases
- A strategy must extend the abstract strategy class
- Used to create rules around indicators and create buy or sell orders on the order book depending on those rules

By backtesting a strategy-generated orderbook, you can see how your strategies perform with different stocks and tune them with new parameter sets

Experiments

Create a strategy that can beat the market

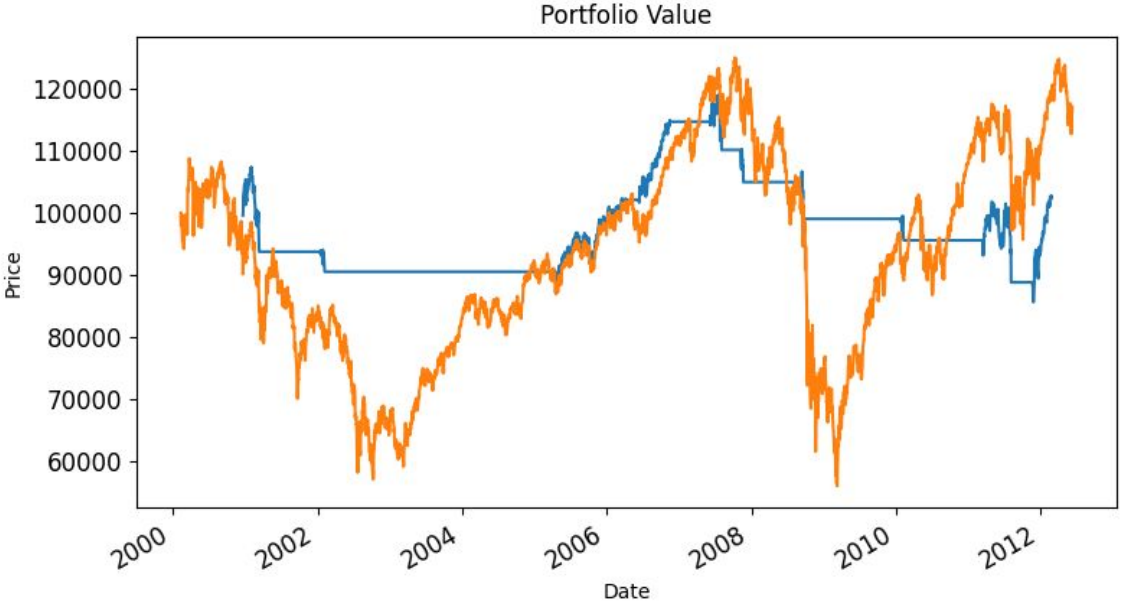
Test the strategy performance on different stocks

My custom strategy (Bollinger Bands and RSI)

My custom strategy rules:

1. Buy when the current price is less than the lowest bollinger band and the RSI is above a certain threshold (ex: 42)
2. It can open up to 3 positions at a time, but only open a max of one per day
3. It can close any position after either the stop loss or take profit threshold is met
    - Stop loss threshold set to 3% drawdown from the position open price
    - Take profit threshold set to 15% profit from the position open price

Using default parameters with SPY (2000- 2012)




```
default_params = {
    'bollinger_window': 20,
    'rsi_window': 14,
    'rsi_buy_threshold': 40,
    'max_positions': 1,
    'take_profit_%': 0.15,
    'stop_loss_%': 0.03
}
```

Initial portfolio = $100,000
Order size = $1,000

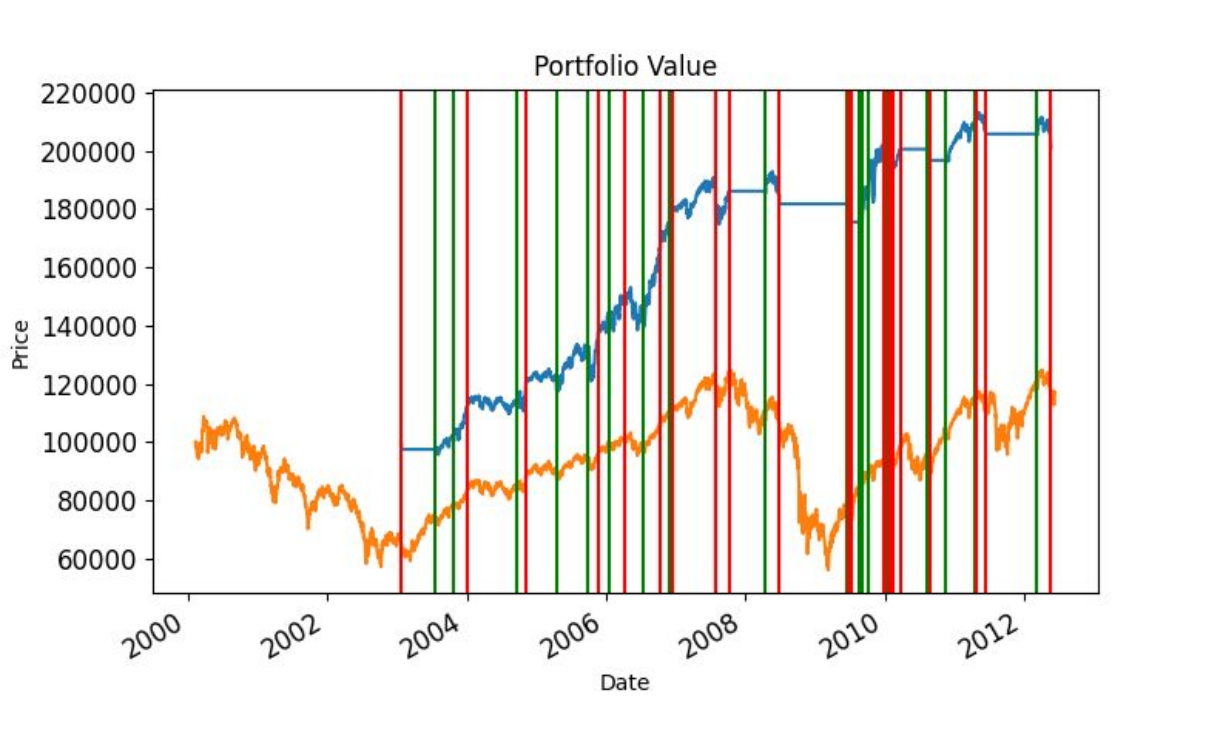
Final portfolio value:
$102,441 = $2,441 gain

Did not beat the market but obtained positive ROI

Seemed to hedge well against market downturns

Blue = Strategy performance; Orange = SPY performance

Optimizing parameters for SPY (2000 - 2012)



Blue = Strategy performance; Orange = SPY performance

```
new_params = {
    # = updated
    'bollinger_window': 9,#
    'rsi_window': 14,
    'rsi_buy_threshold': 42,#
    'max_positions': 3,#
    'take_profit_%': 0.15,
    'stop_loss_%': 0.03
}
```

Initial portfolio = $100,000
Order size = $1,000

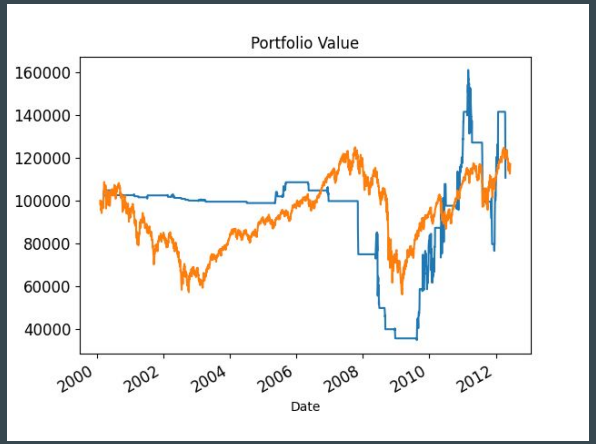
Increased max positions to capitalize on market runs

Final portfolio value:
$201,119 = $101,119 (>2x) gain

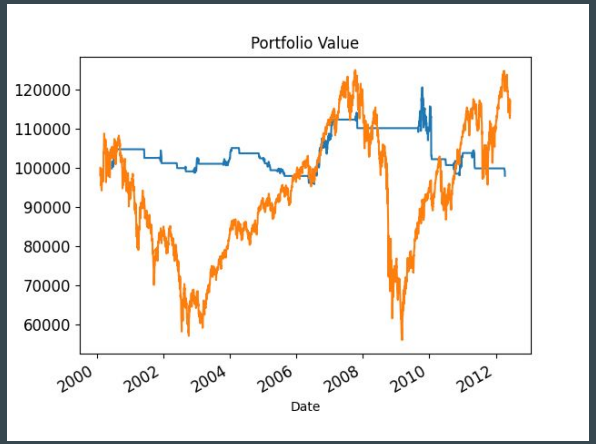
= Success?

More Results

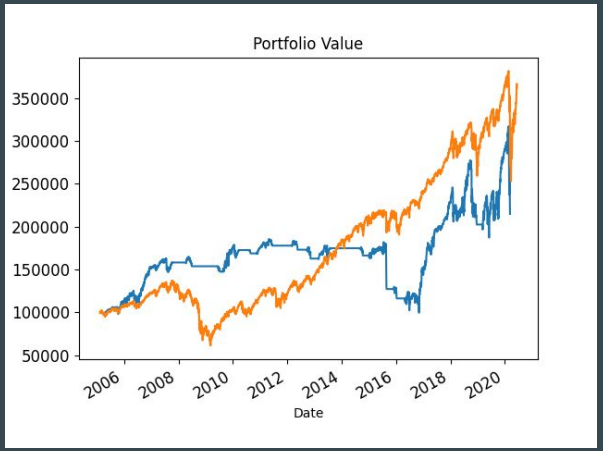
AAPL (2000-2012)



JPM (2000-2012)



SPY (2005-2020)



Obviously it's not perfect, but I think it's a good platform to continue experimenting with

Future Goals

Use machine learning to optimize strategy parameters for maximizing profit

When I originally created this project, I wanted to use SciPy's minimizer to optimize the strategy on specific time periods and stocks

However, many of the strategy parameters are integers. This makes it a non-linear, mixed-integer optimization problem since some parameters are floats and others ints

- Incompatible with scipy's optimizers
- Could not be guaranteed to be solved optimally unless I tried every possible combination (> ~100 billion with 7 parameters)
- Maybe it's possible to use a different machine learning approach such as reinforcement learning?

Thanks for listening