Trading the Index

October 28, 2020

0.1 Trading the Index

- Hypothesis: it's advantageous to trade the index based on the relationship between closing price and the moving average

[1]:
```
import pandas as pd
import numpy as np
import datetime as dt
from pandas_datareader import data as pdr

import matplotlib
%matplotlib inline
import matplotlib.pyplot as plt
```

```
//anaconda3/lib/python3.7/site-packages/pandas_datareader/compat/__init__.py:7:
FutureWarning: pandas.util.testing is deprecated. Use the functions in the
public API at pandas.testing instead.
  from pandas.util.testing import assert_frame_equal
```

[2]:
```
start = '2000-01-01'
end = dt.datetime.now()
spy = pdr.DataReader('SPY', 'yahoo', start, end)
spy.head()
```

[2]:

Date	High	Low	Open	Close	Volume	Adj Close
2000-01-03	148.25000	143.875000	148.25000	145.4375	8164300.0	98.564461
2000-01-04	144.06250	139.640625	143.53125	139.7500	8089800.0	94.709984
2000-01-05	141.53125	137.250000	139.93750	140.0000	12177900.0	94.879379
2000-01-06	141.50000	137.750000	139.62500	137.7500	6227200.0	93.354584
2000-01-07	145.75000	140.062500	140.31250	145.7500	8066500.0	98.776245

0.1.1 Trading based on Moving Average indicator

[3]:
```
# Array of moving average windows to test
moving_avgs = np.arange(10,251,5)

# Data Frame of Adjusted Close for the SPY ETF for each day since 1/03/2000
prices = pd.DataFrame(spy['Adj Close'])

# List to store returns
returns = []

# Data frame to combine returns with their respective moving average
rtdf = pd.DataFrame({})

# For loop to test strategy on each moving average window
for p in range(len(moving_avgs)):

    # Position:
    pos = 0
    num = 0
    pctcng = []
    prices[p] = prices['Adj Close'].rolling(window=moving_avgs[p]).mean()

    # For loop to test the strategy for a given window
    for i in prices.index:

        close = prices.loc[i, 'Adj Close']
        moving_avg = prices.loc[i, p]

        if(close > moving_avg):
            if(pos==0):
                pos = 1
                bp = close

        elif(close < moving_avg):
            if(pos==1):
                pos = 0
                sp = close
                pc = (sp/bp - 1) * 100
                pctcng.append(pc)

    rtdf.loc[p, 'moving average window'] = moving_avgs[p]
    rtdf.loc[p, 'return'] = sum(pctcng)
rtdf.head()
```

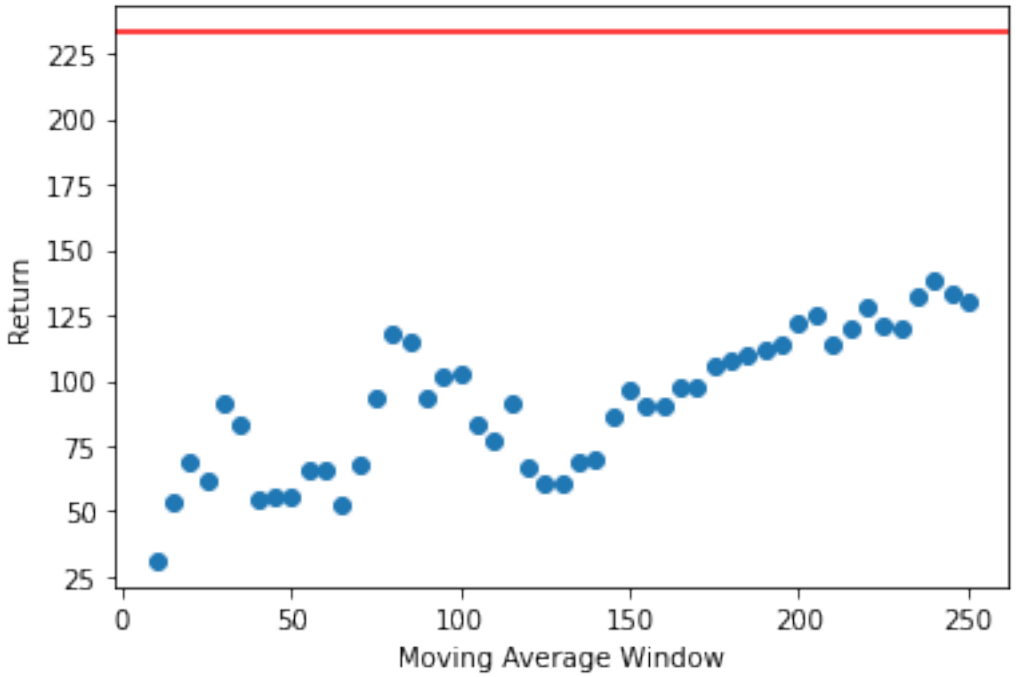
[3]:

	moving average window	return
0	10.0	31.181745

```
1                    15.0   53.780702
2                    20.0   69.033773
3                    25.0   61.637035
4                    30.0   90.981458
```

[4]:
```
price = np.array(prices['Adj Close'])
basic_ret = (((price[-1] - price[0]) / price[0]) * 100)
```

[5]:
```
plt.scatter(rtdf['moving average window'], rtdf['return'])
plt.axhline(basic_ret, c='r')
plt.xlabel('Moving Average Window')
plt.ylabel('Return')
plt.show()
```



The graph above shows the total return for each moving average window used in the strategy. The general trend implies that the bigger the window, the bigger the return; however, it is clear that the return from simply holding a position from the beginning of the testing period to the end (red line) is far greater than any of the strategies I tested.

0.1.2 Improvements:

- add economic predictors
- Finite State Theory