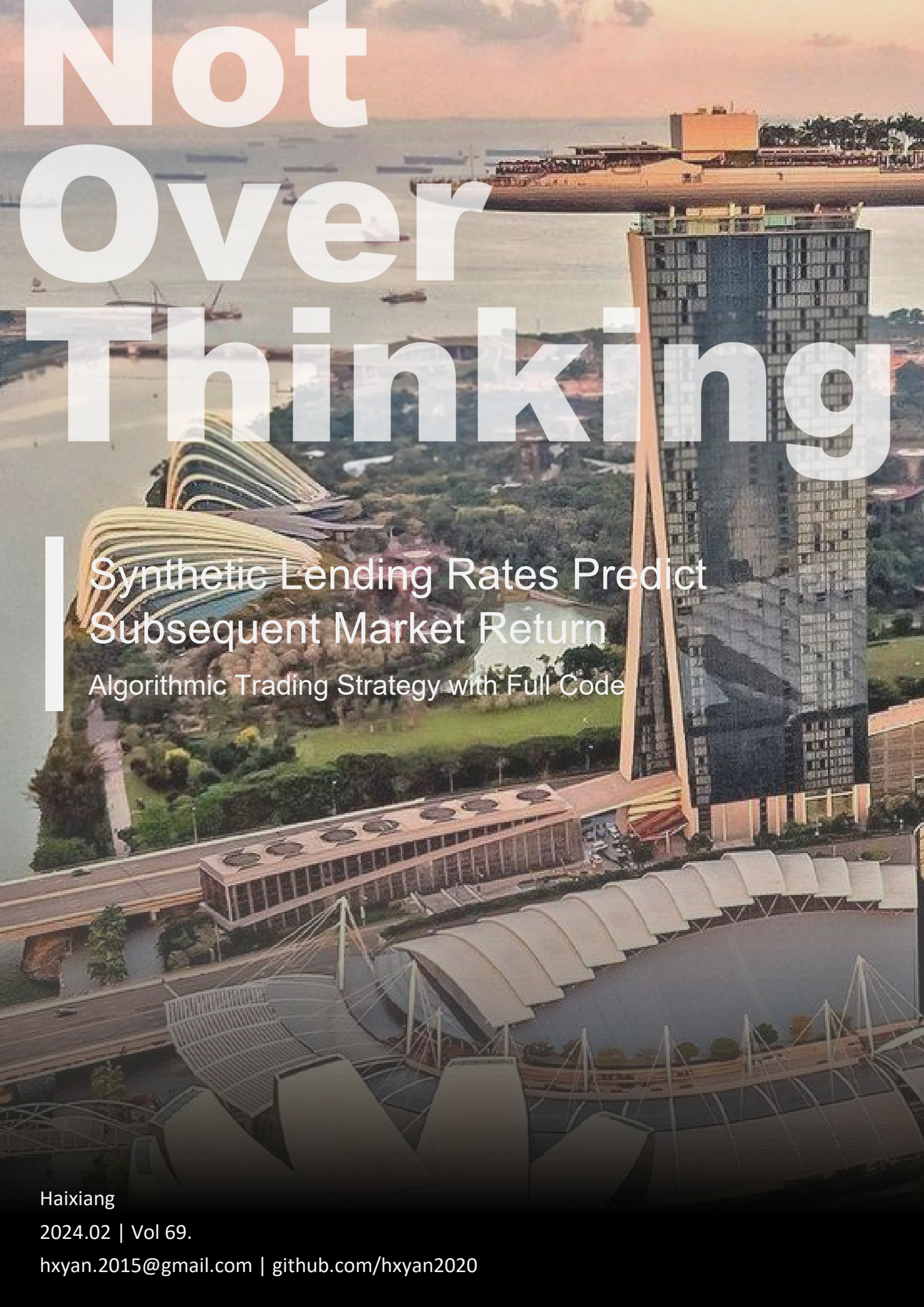

Not Over Thinking

Synthetic Lending Rates Predict Subsequent Market Return

Algorithmic Trading Strategy with Full Code

Haixiang

2024.02 | Vol 69.

hxyan.2015@gmail.com | github.com/hxyan2020

STRATEGY & ECONOMIC RATIONALE

The investment universe consists of SPY ETF. Synthetic shorting costs data are obtained from Bo rrow Intensity Indicators by the CBOE (and includes 4877 stocks/ETFs). The paper utilizes the c onstant maturities of 45 days. Intraday SPY data are obtained from FirstRate Data. The aggregat e (mean) borrow intensity is calculated as equally weighted borrow intensity of each stock/ETF in the sample at day t. The shorting costs data are estimated at a timestamp of 15:57. Calculat e the change in the aggregate intensity at day as the difference of aggregate borrowing inten sity at day t and t-1. Buy the SPY ETF at 15:59 if the difference is positive and short the SPY if the difference is negative. The positions are held for one day and are closed at 15:58 at n ext day.

BUY	SELL
Buy the SPY ETF at 15:59 if the difference is positive	short the SPY if the diffe rence is negative

PARAMETER & VARIABLES

PARAMETER	VALUE
MARKETS TRADED	Equity
FINANCIAL INSTRUMENTS	ETFs
REGION	United States
PERIOD OF REBALANCING	Daily
NO. OF TRADED INSTRUMENTS	1
WEIGHTING	Equal weighting
LOOKBACK PERIODS	N/A
LONG/SHORT	Long & short

ALGORITHM

```
from AlgorithmImports import *#endregion
class SyntheticLendingRatesPredictSubsequentMarketReturn(QCAlgorithm):

    def Initialize(self):
        self.SetStartDate(2016, 1, 1)
        self.SetCash(100000)

        self.spy_symbol:Symbol = self.AddEquity('SPY', Resolution.Minute).Symbol

        self.lending_data_symbol:Symbol = self.AddData(
            QuantpediaLendingRates,
            'lending_rate',
            Resolution.Minute).Symbol

        self.last_lending_mean = None

    def OnData(self, data: Slice):
        curr_time:datetime.datetime = self.Time

        # liquidate at 15:58
        if curr_time.hour == 15 and curr_time.minute == 58:
            self.Liquidate(self.spy_symbol)

        # lending rate data came in
```

```
        if self.lending_data_symbol in data and data[self.lending_data_symbol]:
            curr_lending_mean:float = data[self.lending_data_symbol].Value

            if self.last_lending_mean:
                # calculate daily change in lending rate
                diff:float = curr_lending_mean - self.last_lending_mean

                if diff > 0:
                    self.SetHoldings(self.spy_symbol, 1)
                else:
                    self.SetHoldings(self.spy_symbol, -1)

            self.last_lending_mean = curr_lending_mean
# Quantpedia data.# NOTE: IMPORTANT: Data order must be ascending (datewise)class QuantpediaLen
dingRates(PythonData):
    def GetSource(self, config, date, isLiveMode):
        return SubscriptionDataSource("data.quantpedia.com/backtesting_data/options/lending_rat
es_day_close_matur_45_days.csv".format(config.Symbol.Value), SubscriptionTransportMedium.Remote
File, FileFormat.Csv)

    def Reader(self, config, line, date, isLiveMode):
        data:QuantpediaLendingRates = QuantpediaLendingRates()
        data.Symbol = config.Symbol

        if not line[0].isdigit(): return None

        split:list = line.split(';')

        datetime_str:str = split[0] + ', 15:59'

        data.Time = datetime.strptime(datetime_str, "%Y-%m-%d, %H:%M")
        valid_values:list = list(filter(lambda value: value != '', split[1:]))
        valid_values:list = list(map(lambda str_value: float(str_value), valid_values))
        data.Value = np.mean(valid_values)

        return data
```

BACKTESTING PERFORMANCE

Fig 1. Overall Performance

Total Trades	2505	Average Win	0.73%
Average Loss	-0.64%	Compounding Annual Return	7.879%
Drawdown	30.800%	Expectancy	0.075
Net Profit	68.208%	Sharpe Ratio	0.476
Probabilistic Sharpe Ratio	4.191%	Loss Rate	50%
Win Rate	50%	Profit-Loss Ratio	1.14
Alpha	0.061	Beta	0.018
Annual Standard Deviation	0.132	Annual Variance	0.018
Information Ratio	-0.134	Tracking Error	0.203
Treynor Ratio	3.427	Total Fees	$5518.89
Estimated Strategy Capacity	$61000000.00	Lowest Capacity Asset	SPY R735QTJ8XC9X

Fig 2. Performance Metrics