When Investing Meets Algorithmic Trading

QUANTITATIVE EQUITY TRADING STRATEGIES
FINAL PROJECT
GROUP 3

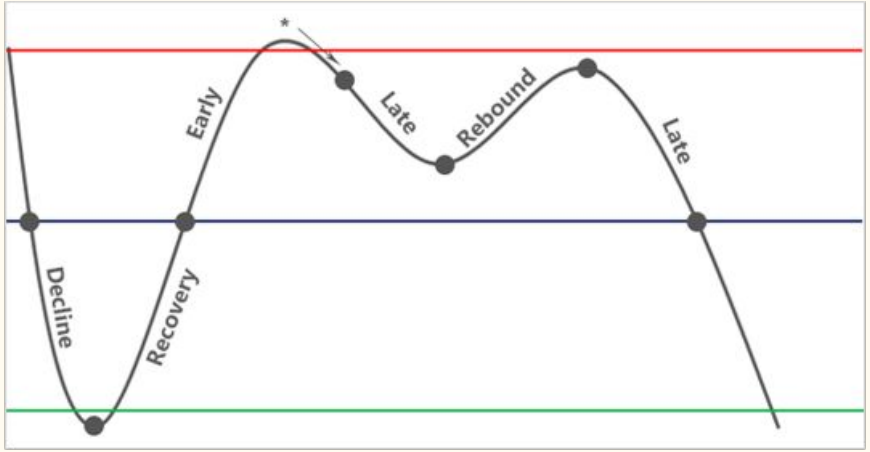
STRATEGY

- Identify business cycle by Leading indicators published by The Conference Board.
- Identify the sectors that have historically performed well or poorly in given cycle using sectoral analysis
- Perform Algorithmic Trading on the same sector ETF where we expect high volatility and opportunity for short term returns
- Long a sectoral ETF that performs well given/after the current market cycle, to hedge our short term losses, if any

TOP LEVEL APPROACH

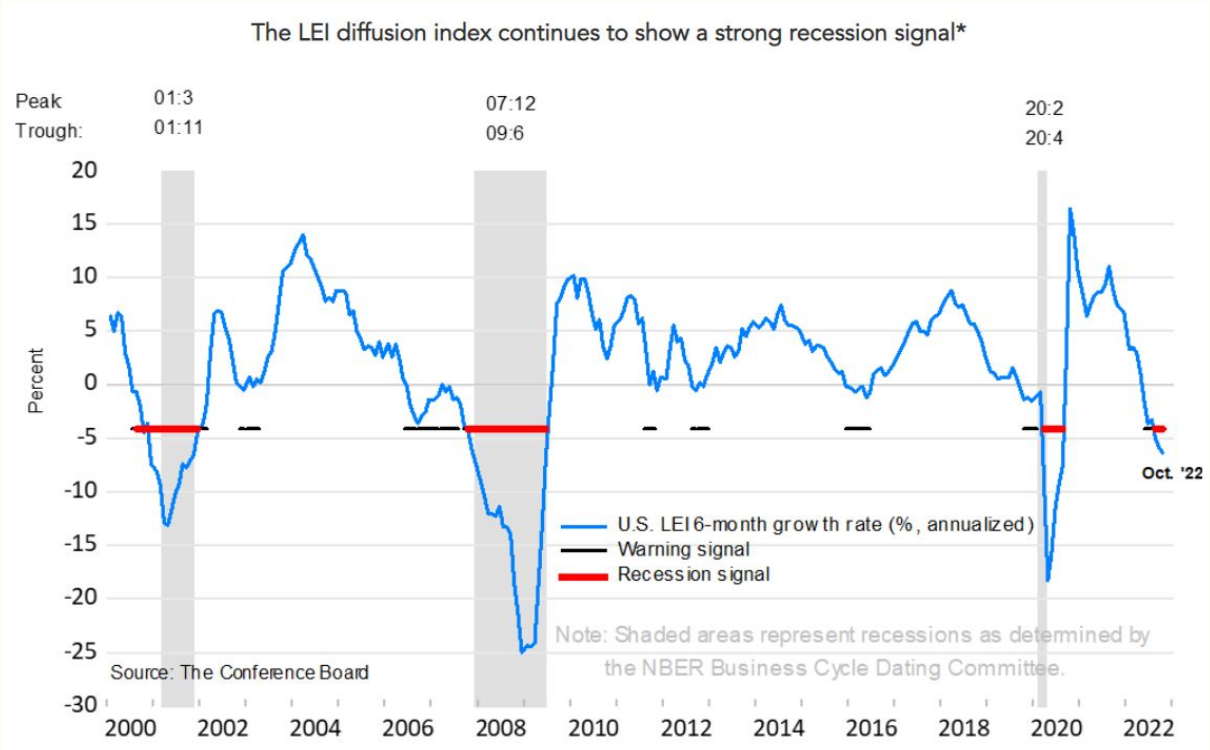
Macroeconomic data is processed and used to determine which part of the market cycle we are in at a given time.

Historical data helped determine which sectors outperform and which sectors underperform during a certain phase.



Sector																		
Market Cycle	Consumer Staples		Healthcare		Utilities		Energy		Industrials		Financials		Consumers Discretionary		Information Technology		Materials	
	Long	Short	Long	Short	Long	Short	Long	Short	Long	Short	Long	Short	Long	Short	Long	Short	Long	Short
Decline	✓						✓			✓		✓				✓	✓	
Recovery		✓	✓			✓							✓			✓	✓	
Early				✓				✓	✓		✓			✓			✓	
Late	✓		✓		✓			✓								✓		✓
Rebound		✓	✓			✓			✓			✓			✓			

CURRENT BUSINESS CYCLE: LEI



- The Composite Index of Leading Indicators is another name for the U.S. Conference Board Leading Economic Index (LEI)
- Its is geared toward predicting the direction of the overall economy over the next few quarters.
- The Index consists of 10 components that indicate the short-term future course of various sectors of the economy, combined into a composite indicator of general economic performance.

Source: Conference Board; Investopedia

ETF SELECTION STRATEGY

The given results show the LEI data and the reflecting business cycle

In [36]:
```
data_dir = '/Users/rishabhsingh/Desktop/untitled folder 2/data/'

lei = pd.read_excel(data_dir + 'LEI Calculation.xlsx')
```

In [ ]:
```
```

In [37]:
```
lei = lei.set_index('Month')
lei.loc['2012-01-01':].tail(10)
```

Out[37]:

Month	LEI (2016=100)	LEI Y/Y	High Boundary	Low Boundary	Business Cycle
2022-01-01	118.5	0.062780	0.07	-0.03	Late
2022-02-01	119.4	0.071813	0.07	-0.03	Late
2022-03-01	119.3	0.059503	0.07	-0.03	Late
2022-04-01	118.7	0.043058	0.07	-0.03	Late
2022-05-01	117.9	0.027003	0.07	-0.03	Late
2022-06-01	117.1	0.013853	0.07	-0.03	Late
2022-07-01	116.6	0.000000	0.07	-0.03	Decline
2022-08-01	116.4	-0.008518	0.07	-0.03	Decline
2022-09-01	115.8	-0.015306	0.07	-0.03	Decline
2022-10-01	114.9	-0.027096	0.07	-0.03	Decline

ETF SELECTION STRATEGY (Continued)

In [3]:
```
staples = ['XLP', 'VDC', 'IYK', 'FSTA', 'RHS', 'FXG', 'PBJ', 'PSL']
healthcare = ['XLV', 'VHT', 'IYH', 'IHI', 'XHE', 'IHF', 'GERM', 'CNCR']
utilities = ['XLU', 'VPU', 'FUTY', 'IDU', 'RYU', 'FXU', 'PUI', 'UTES']
energy = ['XLE', 'FXN', 'XES', 'PXJ', 'AMLP', 'IEZ', 'IEO', 'XOP']
industrials = ['XLI', 'VIS', 'FXR', 'IYJ', 'FIDU', 'RGI', 'AIRR', 'PRN']
financials = ['XLF', 'VFH', 'KBE', 'IYF', 'FNCL', 'IYG', 'FXO', 'RYF']
discretionary = ['XLY', 'VCR', 'FDIS', 'IYC', 'RCD', 'FXD', 'PEZ', 'PSCD']
info_tech = ['VGT', 'XLK', 'IYW', 'FTEC', 'RYT', 'QTEC', 'FXL', 'XNTK']
materials = ['XLB', 'FXZ', 'RTM', 'PYZ', 'XME']

sector_list = [staples, ### each of these is a list of str tickers
               healthcare,
               utilities,
               energy,
               industrials,
               financials,
               discretionary,
               info_tech,
               materials,
               ['SPY']]
```

In [ ]:
```
```

In [4]:
```
volume_all_data_df = pd.DataFrame()

for sector_name in sector_list:

    price_data = yf.download(sector_name, start = "2012-01-01")['Volume']

    price_data.index = pd.to_datetime( price_data.index)

    volume_all_data_df = pd.concat([volume_all_data_df, price_data], axis=1)
```

In [14]:
```
### long/short lists of market cycles
decline_long = [staples, energy, materials]
decline_short = [industrials, financials, info_tech]

recovery_long = [healthcare, industrials, financials]
recovery_short = [staples, utilities, info_tech]

early_long = [industrials, financials, materials]
early_short = [healthcare, energy, discretionary]

late_long = [staples, healthcare, utilities]
late_short = [energy, info_tech, materials]

rebound_long = [healthcare, industrials, info_tech]
rebound_short = [staples, utilities, financials]
```

In [15]:
```
decline_long
```

Out[15]:
```
[['XLP', 'VDC', 'IYK', 'FSTA', 'RHS', 'FXG', 'PBJ', 'PSL'],
 ['XLE', 'FXN', 'XES', 'PXJ', 'AMLP', 'IEZ', 'IEO', 'XOP'],
 ['XLB', 'FXZ', 'RTM', 'PYZ', 'XME']]
```

In [17]:
```
materials
```

Out[17]:
```
['XLB', 'FXZ', 'RTM', 'PYZ', 'XME']
```

ETF CONSIDERED

For the investing part of our strategy, we are pursuing a long only strategy which would serves as a hedge component for our algorithmic trading since our screened ETF is expected to perform well after the declining phase.

To take advantage of the market volatility, we are performing algorithmic trading on the same ETF using Supertrend indicator and Bollinger Bands.

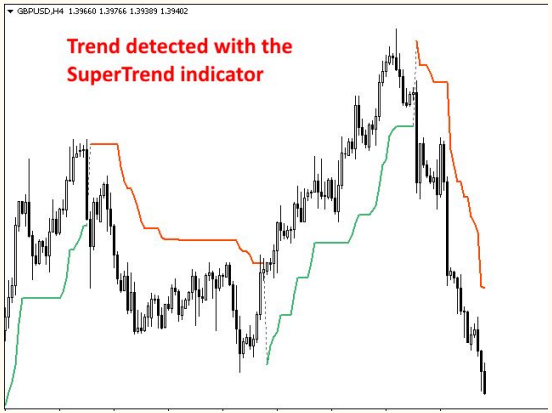
ETF Selected: Materials Select Sector SPDR Fund (XLB)

BOLLINGER BANDS

A Bollinger Band 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.

SUPERTREND

A Supertrend is a trend following indicator similar to moving averages. It is plotted on price and the current trend can simply be determined by its placement vis-a-vis price. The indicator is constructed with the help of two parameters- period and multiplier.



Up = (high + low / 2 + multiplier x ATR

Down = (high + low) / 2 – multiplier x ATR

Calculation of Average True Range –

[(Prior ATR x 13) + Current TR] / 14

Average True Range (ATR) is the average of true ranges over the specified period. ATR measures volatility, taking into account any gaps in the price movement. Typically, the ATR calculation is based on 14 periods, which can be intraday, daily, weekly, or monthly.

TRADING SIGNALS GENERATED FOR XLB

TRADING RESULTS

We invested $1M in XLB using the supertrend strategy and a 2% profit was generated.

```
st_investment_ret_df = pd.DataFrame(st_investment_ret).rename(columns = {0:'investment_returns'})

total_investment_ret = round(sum(st_investment_ret_df['investment_returns']), 2)
profit_percentage = floor((total_investment_ret/investment_value)*100)
print(cl('Profit gained from the trading strategy by investing $1m : {}'.format(total_investment_ret), attrs = ['bold']
print(cl('Profit percentage of the strategy : {}%'.format(profit_percentage), attrs = ['bold']))
```

```
Profit gained from the trading strategy by investing $1m : 23009.5
Profit percentage of the strategy : 2%
```

```
In [129]: SharpeRatio = (profit_percentage - benchmark_profit_percentage)/res
          SharpeRatio

Out[129]: 0.01821130484085287
```

LONG ONLY RESULTS

For our long-only position we invested $1M and we bought XLB at 74.8799 on 1st of November 2022.

We exited our position on 1st of December at 82.76997.

```
In [119]: #Long Only Profit
          buypos = 1000000/74.87999
          buy*82.769997 - 1000000

Out[119]: 105368.69729817007
```

PORTFOLIO RETURNS

```
In [123]: #return percentage of combined strategy
          2000000/(lo + total_investment_ret)

Out[123]: 15.578969342861372
```

BENCHMARK RETURNS

In [106]:
```
# SPY ETF COMPARISON
def get_benchmark(start_date, investment_value):
    spy = yf.download('SPY', start_date)['Close']
    benchmark = pd.DataFrame(np.diff(spy)).rename(columns = {0:'benchmark_returns'})

    investment_value = investment_value
    number_of_stocks = floor(investment_value/spy[-1])
    benchmark_investment_ret = []

    for i in range(len(benchmark['benchmark_returns'])):
        returns = number_of_stocks*benchmark['benchmark_returns'][i]
        benchmark_investment_ret.append(returns)

    benchmark_investment_ret_df = pd.DataFrame(benchmark_investment_ret).rename(columns = {0:'investment_returns'})
    return benchmark_investment_ret_df

benchmark = get_benchmark('2022-01-01', 1000000)
investment_value = 1000000
total_benchmark_investment_ret = round(sum(benchmark['investment_returns']), 2)
benchmark_profit_percentage = floor((total_benchmark_investment_ret/investment_value)*100)
print(cl('Benchmark profit by investing $1m : {}'.format(total_benchmark_investment_ret), attrs = ['bold']))
print(cl('Benchmark Profit percentage : {}%'.format(benchmark_profit_percentage), attrs = ['bold']))
print(cl('Strategy profit is {}% higher than the Benchmark Profit'.format(profit_percentage - benchmark_profit_percenta
```

```
[*********************100%***********************]  1 of 1 completed
Benchmark profit by investing $1m : -215010.62
Benchmark Profit percentage : -22%
Strategy profit is 24% higher than the Benchmark Profit
```

FUTURE SCOPE

REFERENCES

- Bloomberg
- Yahoo finance
- Investopedia
- Elearnmarkets
- The Conference Board
- https://www.niftytradingacademy.com/blog/how-to-use-supertrend-indicator