FIN 566 Final Project Presentation

Khavya Chandrasekaran
Xuehui Chao
Yong Xie
Kushal Goenka

Overview

- Devise a trading strategy based on real market data from IEX (Investors Exchange)
- Analyse daily trading data from October 2019 from the raw feeds from IEX
- Data format included Various Timestamps, Sequence Number, Tick Type, Price, Size and Symbol (Stored in

	COLLECTION_TIME	SOURCE_TIME	SEQ_NUM	TICK_TYPE	MARKET_CENTER	PRICE	SIZE
0	2019-10-08 13:30:01.488931072	2019-10-08 13:30:01.488905602	98015	T	IEX	178.200	100
1	2019-10-08 13:30:35.962691072	2019-10-08 13:30:35.962669101	155172	T	IEX	178.040	100

Step in the strategy development process

1. Analyse data, plot correlations, conduct regression analysis and chose a handful of interesting Symbols

2. Analyse the symbols against a popular ETF (Exchange Traded Fund)

3. Come up with a strategy to leverage statistical indicators in the training data

4. Tune the parameters of the strategy to get the best results for the given data during backtesting

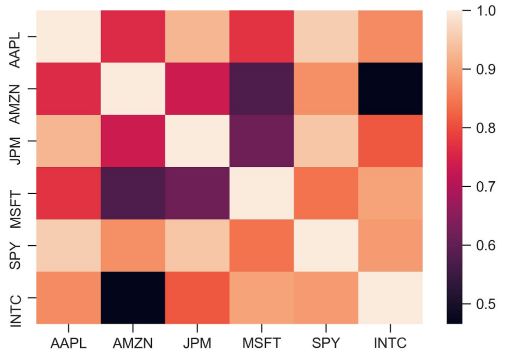
5. Choose the best parameters and backtest on newer market data to test our strategy on unseen data

Initial Analysis and Thoughts

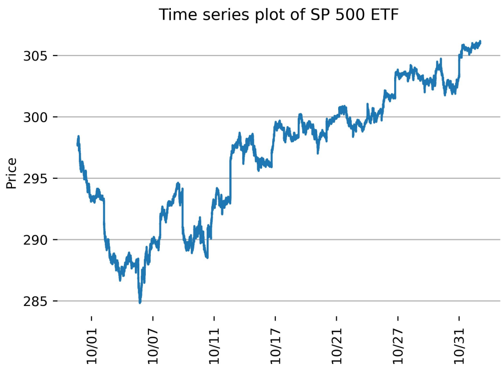
- We initially planned to look at DIA, which is an ETF that tracks the Dow Jones Index, and its major components
- Realised that DIA is not as liquid as its components and hence switched to using the very popular S&P 500 ETF, SPY
- Looking at some of the largest components of the Index was a good starting point to make analysis simpler
- We confirmed that these symbols were heavily traded on any given trading day, and hence made for good liquid targets.
- Correlation analysis and plots led us to a few chosen symbols to further analyse, and base our trading strategy on

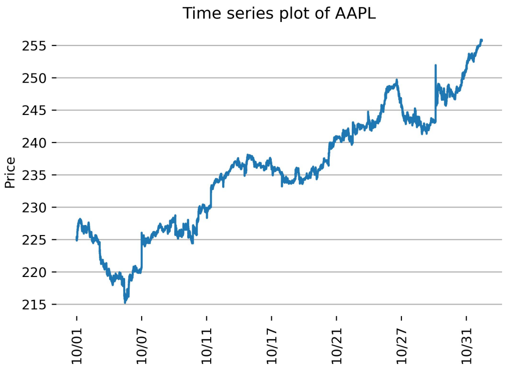
Final Components chosen

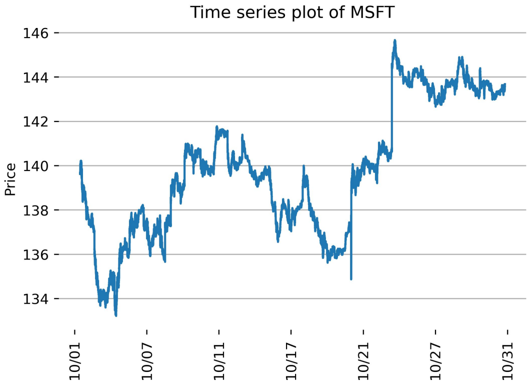
- We decided on the following symbols:
    1. AAPL - Apple Inc.
    2. MSFT - Microsoft Corporation
    3. JPM - JPMorgan Chase & Co
    4. INTC - Intel Corporation
- The correlation heatmap provides a visual justification for choosing specific symbols
- Closer the number to 1, more correlated the symbol on the x axis is with one on the y axis
- Also considered the fluctuations in these symbols for the most strongly correlated symbols to SPY

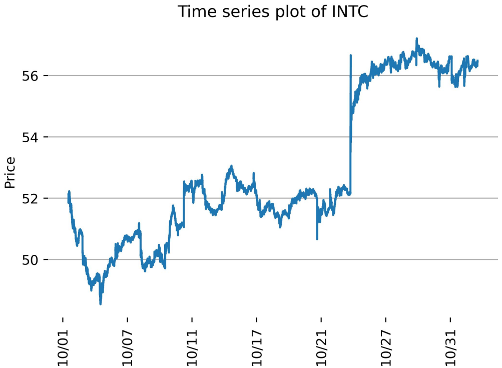


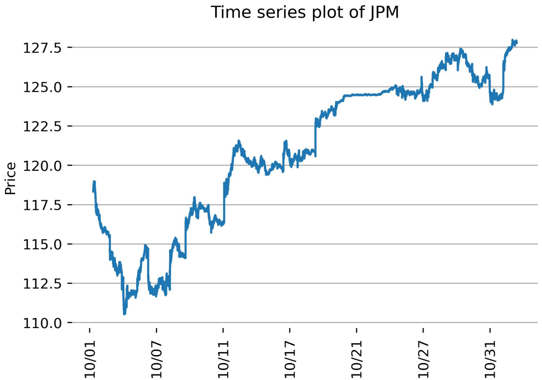
What were these companies trading at?











Regression between Components and SPY seconds as time interval

	MSFT	AAPL	INTC	JPM
SPY-1s	0.2660***	0.0642***	0.3655***	0.3891***
SPY-5s	0.1380***	0.1626***	-0.0112***	0.0584***
SPY-10	0.0046***	-0.0394**	-0.0217	0.0478***
SPY-30	0.0020	0.0034	0.0397**	0.0735***
COMP-1s	-0.1840***	-0.1114***	-0.1520***	-0.0888***
COMP-5s	-0.1220***	-0.0147	0.0155	-0.0767***
COMP-10s	0.0105	-0.0125	-0.0365***	-0.0678***
COMP-30s	-0.0326***	-0.0062	0.0137	0.0169*

Note: *** represents the coefficient is significant at 0.001 level; ** 0.01 represents level; * represents 0.05 level. COMP stands for the ticker of dependent variable. For example, COMP-1s in the MSFT column represent MSFT-1s.

Using SPY to predict components has larger coefficient

Regression between Components and SPY milliseconds as time interval

	MSFT	AAPL	INTC	JPM
SPY-1ms	-0.0127	-0.0021	2.312e-05	-0.0063
SPY-5ms	0.0784***	0.0110	-0.0686**	0.0202
SPY-10ms	0.0014	0.0205	0.1058***	0.0331
SPY-30ms	-0.0464***	-0.0361**	0.0068	0.0124
SPY-60ms	0.0535***	0.0422**	0.0432*	-0.0189
SPY-90ms	0.0871***	0.0957***	0.0513***	0.0484***
COMP-1ms	0.0955***	0.0894***	-0.0376***	-0.0103*
COMP-5ms	-0.0838***	-0.0322**	-0.0086	0.0182
COMP-10ms	-0.0049	-0.0133	-0.0021	0.0011
COMP-30ms	-0.0089	-0.0218**	-0.0094	-0.0064
COMP-60ms	-0.0105	0.0537***	-0.0275***	-0.0019
COMP-90ms	-0.0041	-0.0347***	0.0279***	-0.0016

Note: *** represents the coefficient is significant at 0.001 level; ** 0.01 represents level; * represents 0.05 level. COMP stands for the ticker of dependent variable. For example, COMP-1s in the MSFT column represent MSFT-1s.

All the coefficients are so close to zero, the prediction might not be good

Regression filtering out the unchanged price instance using microseconds

	AAPL	JPM	INTC
SPY-1ms	0.6365***	0.0979	0.4034
SPY-5ms	0.6365***	0.0979	0.4034
SPY-10ms	-1.5391***	-1.2193	-1.0022
SPY-30ms	0.0225	0.3896	-0.5379
SPY-60ms	-0.4848	-0.5751	0.8031
SPY-90ms	1.3971***	1.8756	0.8031

When filtering out the unchanged price movement instance, the using SPY 1ms, 5ms, 10ms, and 90ms to predict AAPL is good. But the coefficient between SPY, JPM, INTC is not slightly different than zero

Interesting Discoveries

- Counter intuitively, the Index usually leads its component
- ETF is an effective short term predictor for prices of its most highly weighted components
- Only a single past trade was not a good predictor, and hence we needed more historical data to factor into our strategy
- Threshold values as a percentage don’t work well, as the changes are very small and absolute. We therefore considered absolute thresholds for our signal

High level Strategy Description

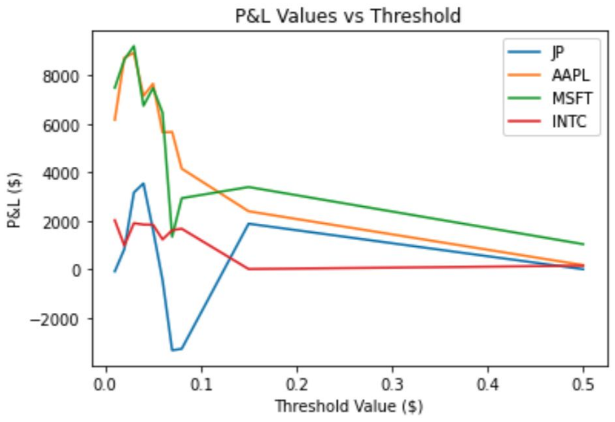
- We define a Signal, and an associated Component to trade on
- We define an up and down threshold parameter to determine when to place a BUY or SELL order (the default is $0.05)
- We have another tuning parameter, the number of past trades to factor into the decision making (the default is 2)
- On every trade, if the current price of the signal is greater than the price traded in the last 2 trades, by the threshold amount, we recognize the upward trend, and send a BUY order for the last traded quantity
- On every trade, if the current price of the signal is lesser than the price traded in the last 2 trades by the threshold amount, we recognize the downward trend and send a SELL order or the entire quantity of the component that is held (Close out the strategy, or liquidate our position)

Executing the Trade, Stop-loss and Profit realization

- We saw how we recognize and set the ‘signal’ to buy or sell our current holdings
- Our default aggressiveness is $ 0.00, trades are sent at exactly the current Market price
- The strategy then sends an order to the trading engine and waits to hear back about a completed order before once again analysing the past trade messages for a suitable signal
- We also alternate between buying the component and selling all of it, we buy on an upward momentum and hold until we see a downward trend
- Furthermore, to prevent large losses, we have logic that exits our current position if we see a 5% drop from our last successful order to BUY (Average Cost).
- We also exit the strategy when we have gained 1% from our last successful order to BUY

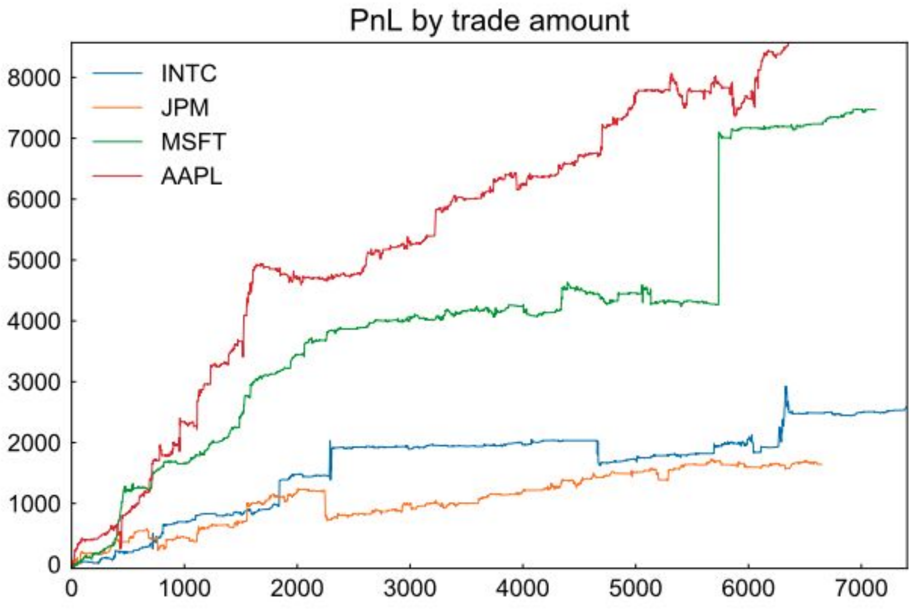
Parameter Tuning Results

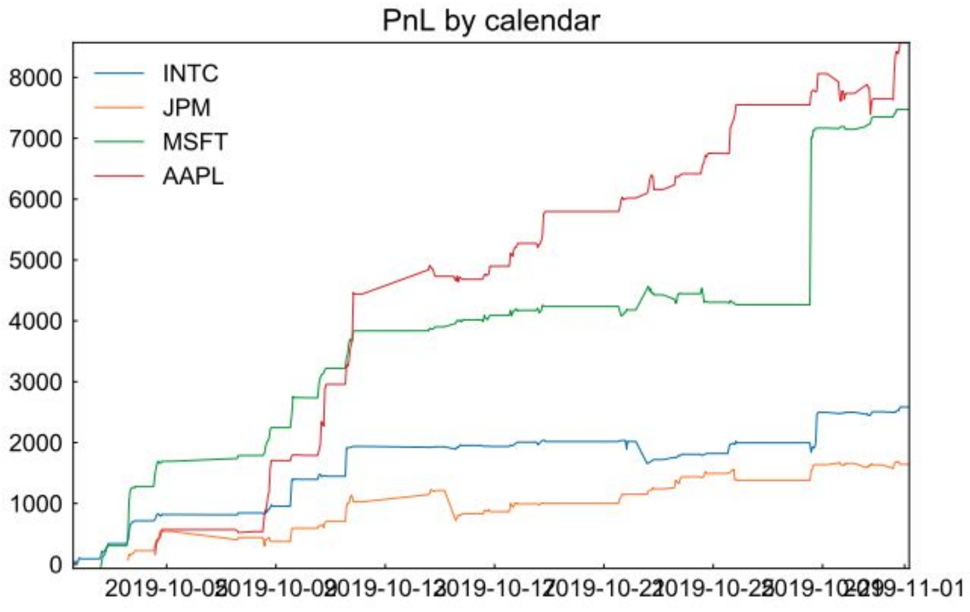
- Varying the threshold value and running the backtest for all the symbols, we see a 'sweet spot' in the $0.03 - $0.05 range where the profits were maximized. Any greater and the profitability reduced exponentially
- Negative aggressiveness, leads to losses (Hypothesis: Not Filled)
- Best Aggressiveness for all symbols was $0.00, always decreased P&L with more aggressive strategies



Backtesting result analysis

- PnL





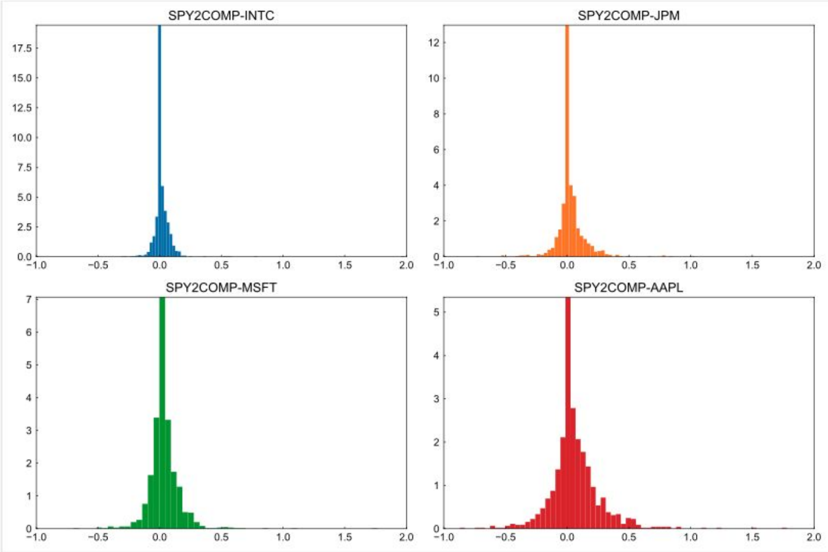
Backtesting result analysis

- Trade Efficiency

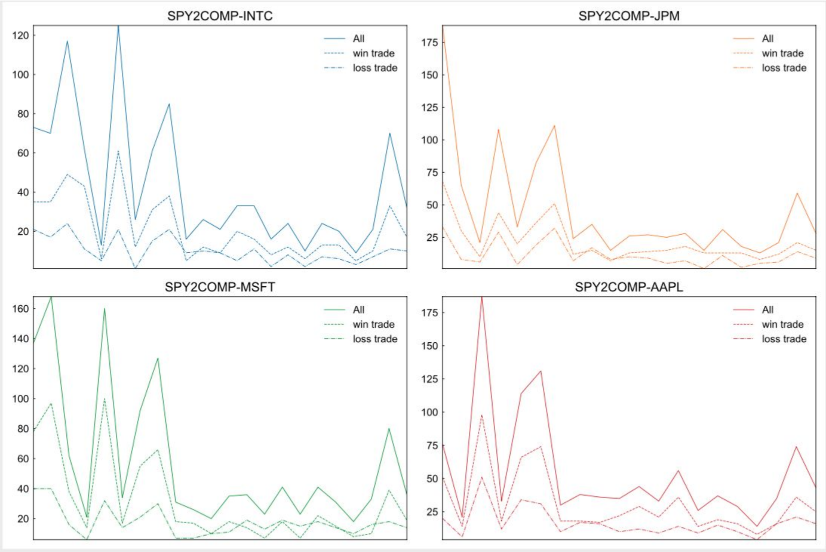
	PnL	Trade amount	Winning rate	Ave. wining PnL	Ave losing PnL
MSFT	$7472	1275	53.9%	$0.141	$-0.066
AAPL	$8576	1092	56.4%	$0.193	$-0.148
INTC	$2584	987	49.5%	$0.057	$-0.049
JPM	$1642	973	46.0%	$0.093	$-0.089

Backtesting result analysis

- Average PnL distribution

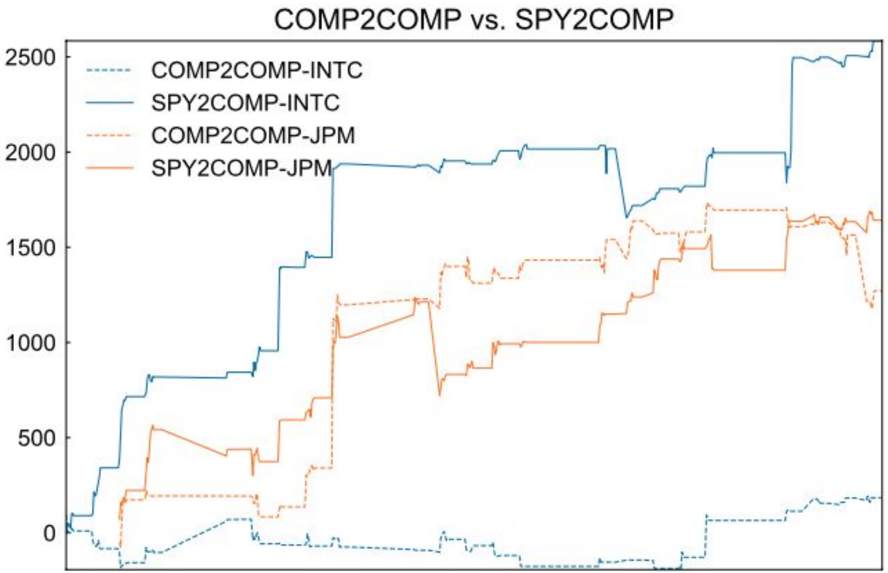


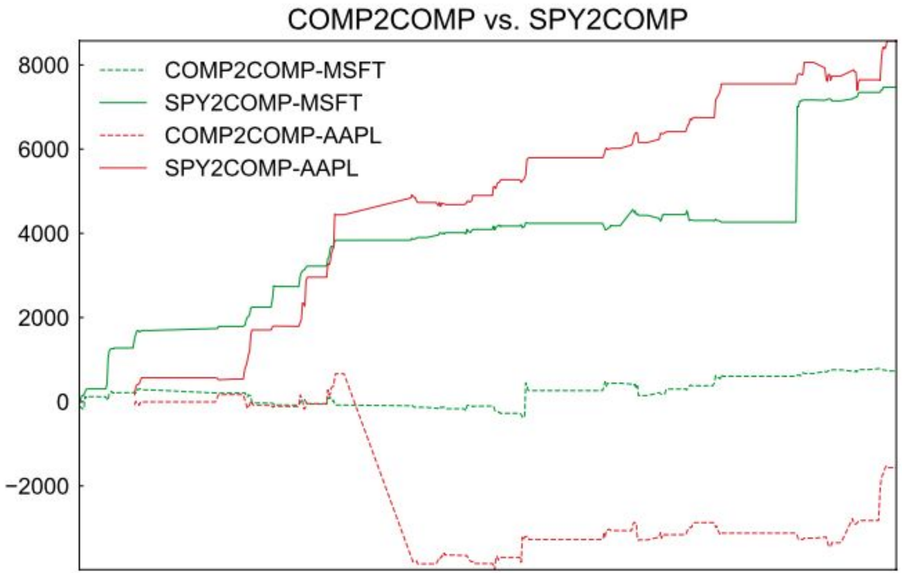
- Trades' temporal distribution



SPY2COMP entails that we use SPY as an indicator to predict the Component Symbol

Backtesting result analysis

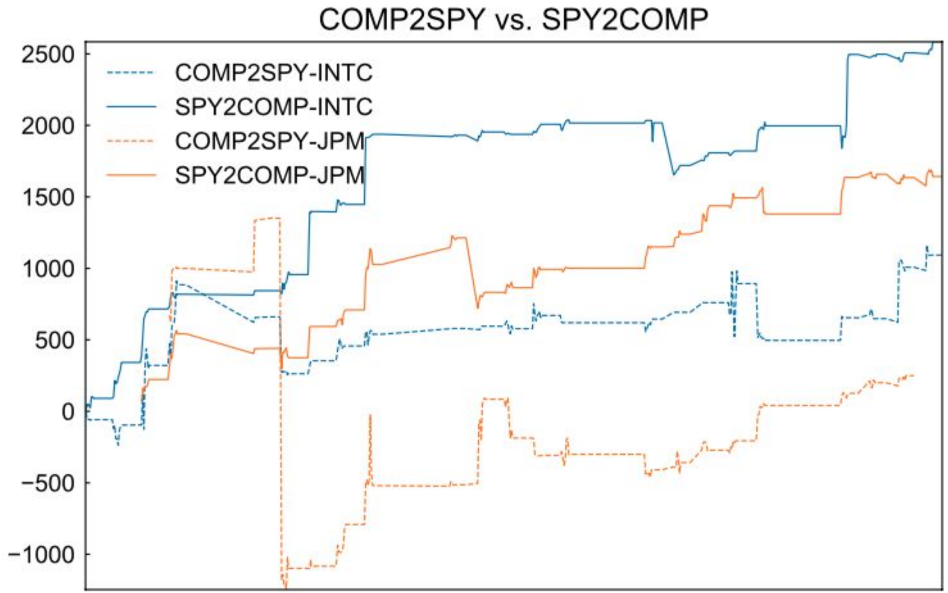


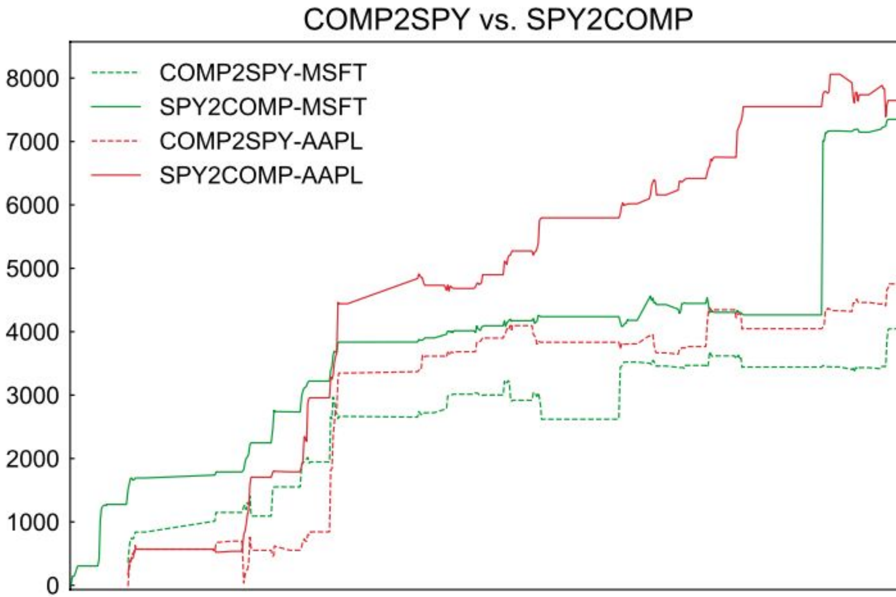


COMP2COMP entails that we use the component as an indicator to predict the component itself, and the definition of SPY2COMP follows from before

Backtesting result analysis

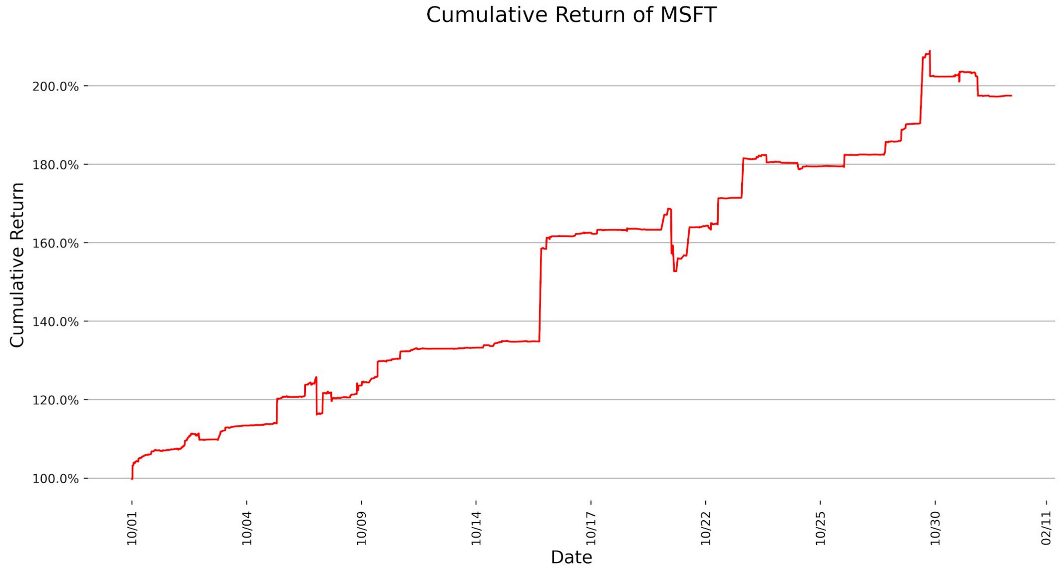
- Reverse leading effect





COMP2SPY entails that we use the component as an indicator to predict SPY, and the other definitions follow from the previous slides

Backtesting result analysis



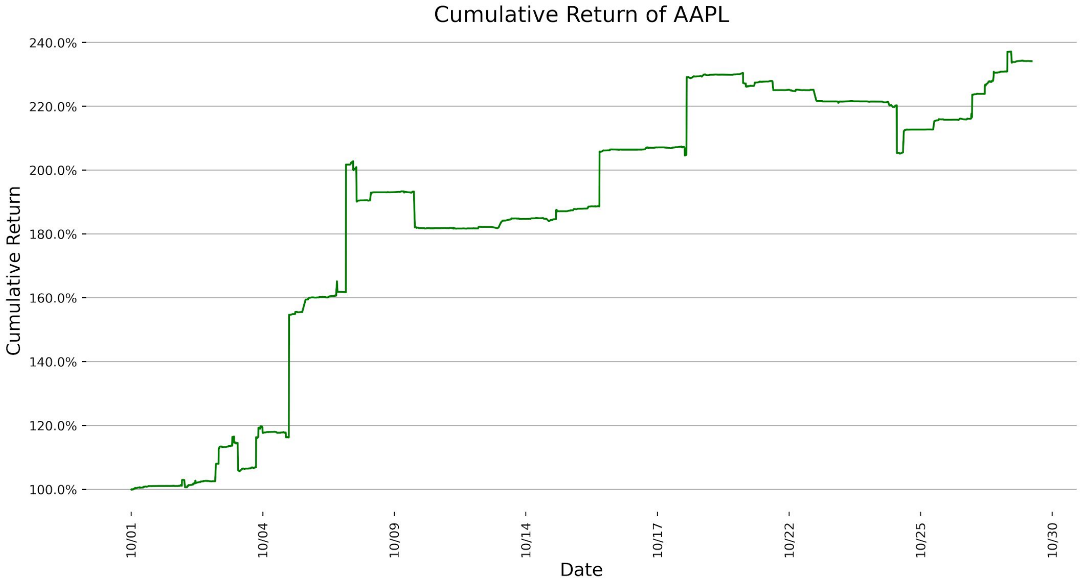
Cumulative Return	Annualized Return	Volatility	Sharpe	Sortino	T-Stat	P-Value
97.507%	352267.064%	2.640%	133412.171	296663.347	2.686	0.007

The cumulative return is calculated using the simple return rate (from fill csv file)compounded among all the transactions.

When a transaction is made, which is we have a complete buy and sell, we calculate the simple return rate by dividing the pure return with the total cost of buying the stocks

MSFT's simple return rate is significantly different than zero

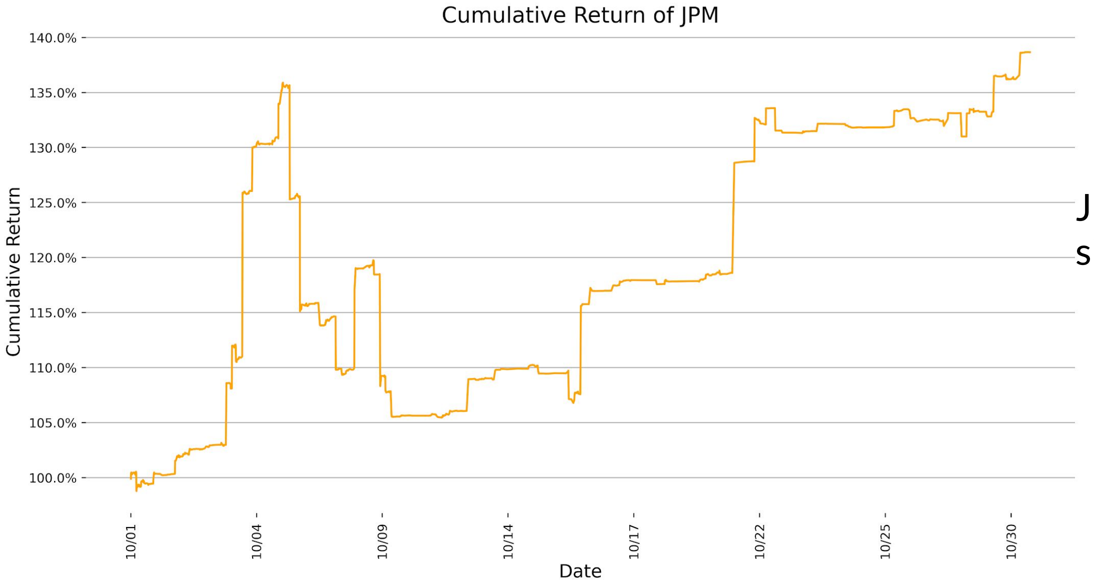
Backtesting result analysis



Cumulative Return	Annualized Return	Volatility	Sharpe	Sortino	T-Stat	P-Value
134.135%	2713744.804%	5.026%	539891.413	1867062.713	1.984	0.047

AAPL’s simple return rate is significantly different than zero

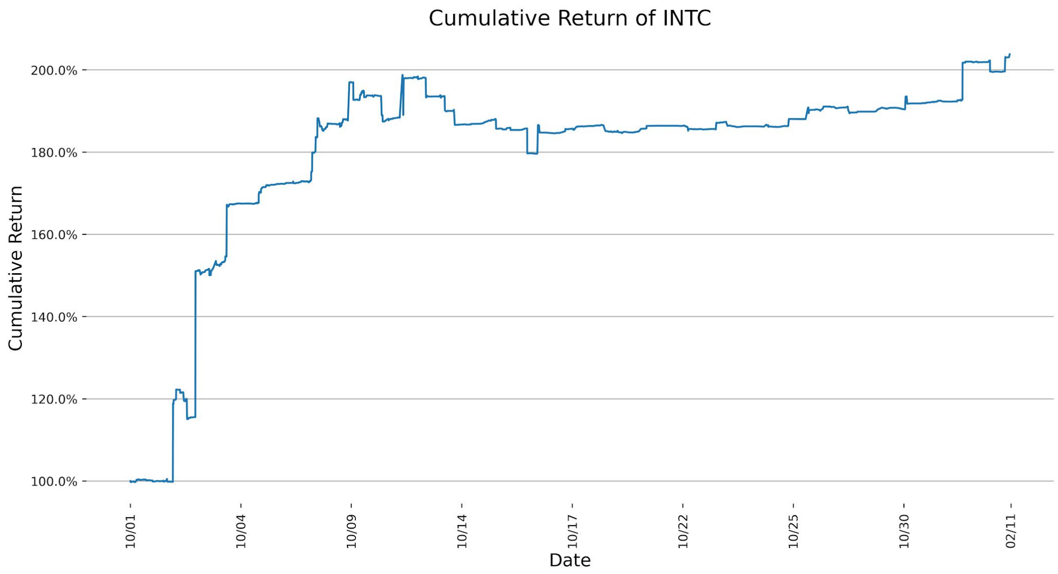
Backtesting result analysis



Cumulative Return	Annualized Return	Volatility	Sharpe	Sortino	T-Stat	P-Value
38.663%	4952.688%	2.914%	1699.846	2855.516	1.369	0.171

JPM's simple return rate is not significantly different than zero

Backtesting result analysis

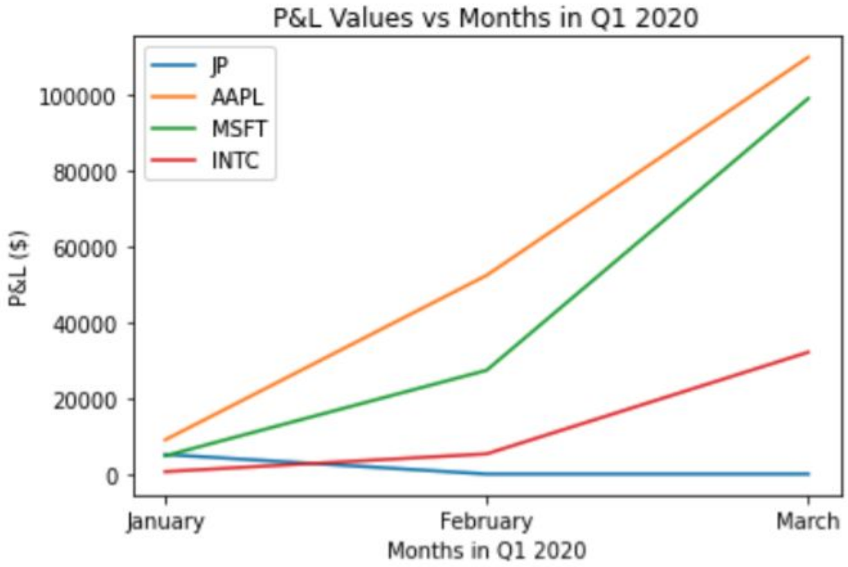


INTC's simple return rate is not significantly different than zero

Cumulative Return	Annualized Return	Volatility	Sharpe	Sortino	T-Stat	P-Value
103.804%	513427.848%	4.298%	119457.607	540330.730	1.955	0.051

Testing on New Data

- We were additionally provided data from the first quarter of 2020, January, February and March.
- We ran the backtest with our default parameters and saw positive gains.
- AAPL Approached over $100k in March
- Hypothesize that this was due to large volatility, and heavy trading in Technology stocks



Conclusion

- Under the ideal conditions, we see that our strategy does generate profits for the chosen components(especially for MSFT and AAPL)
- We even tested our strategy on new unseen data, and it performs well. Indicating very minimal overfitting
- We could improve and handle more complicated real world situations

Improvement

- Tune Parameters such as Aggressiveness, Number of past trades, threshold, stop-loss, profit realization parameters
- Try more combination of future return and historical return time windows.
- Currently we remove those price unchanged instances in order to have a detail look at the regression in microseconds level. Probably we can set up a threshold to remove those tiny price movement to have a better correlation
- Try to use the pnl dataset to generate the cumulative rate, currently we find it’s hard to use pnl cumulative return to generate cumulative return rate