Using Language Processing to Predict Stock Performance

Jonathan Khalfayan, Justin Kahl, Santiago Rodriguez, Matthias Schmitz, Sam Sklar, Juan Pablo Villarreal
Aka "The Young Ones"

Inspiration

- “Leveraged Small Value Equities” by Daniel Rasmussen, Brian Chingono
- “Forcasting Debt Paydown Among Leveraged Equities” by Daniel Rasmussen, Brian Chingono

FORECASTING DEBT PAYDOWN AMONG LEVERAGED EQUITIES

Analysis of U.S. Stocks from 1964 to 2012

Brian Chingono and Dan Rasmussen*

Working Draft: May 2016

Abstract

Using a random sample of 60% of our cross-sectional data on U.S. stocks from 1964 to 2012, we trained four machine learning algorithms to forecast debt paydown over a one-year horizon. An evaluation of these candidate models on half of the hold-out sample (20% of the original dataset) showed that a boosted trees algorithm can forecast debt paydown with up to 70% precision over the next year. This boosted trees model achieved similar results in a second out-of-sample test on the remaining 20% of original data. While information on one-year-ahead equity returns was not used in training or evaluating any of the models, our results show that stocks with a higher estimated probability of paying down debt in the next year also earn higher average returns in that one-year-ahead period. A back-test of the boosted trees model's forecasts of debt paydown between 1965 and 2012 shows a 10.3 percentage point spread between the average annual returns of portfolios formed from the 10th decile of estimated debt paydown probability versus annual portfolios formed from the 1st decile of estimated debt paydown probability. When the 10th decile is combined with a value investment strategy to focus on cheap

LEVERAGED SMALL VALUE EQUITIES

Brian Chingono, *The University of Chicago Booth School of Business*
brian@verdadcap.com

Daniel Rasmussen, *Stanford Graduate School of Business*
dan@verdadcap.com

Working Paper, August 2015

Abstract

The size premium and value premium are well documented in academic studies. We contribute to this literature by finding that leverage – as defined by long term debt divided by enterprise value – enhances the average returns of a small-value investment strategy. At the company level, our results indicate that there is a positive interaction between leverage and value. We test a variety of quality and technical factors to develop a theory of what works in leveraged small-value equity investing. We develop a ranking system for creating annual portfolios of leveraged small-value stocks in the United States. This ranking system prioritizes smaller, cheaper and more leveraged stocks that are already paying down debt and exhibit improving asset turnover. Annual portfolios of the top 25 stocks in this ranking system have a 25.1% average annual return between 1965 and 2013. At a standard deviation of 39.4%, the Sharpe Ratio of these annual

Roadmap

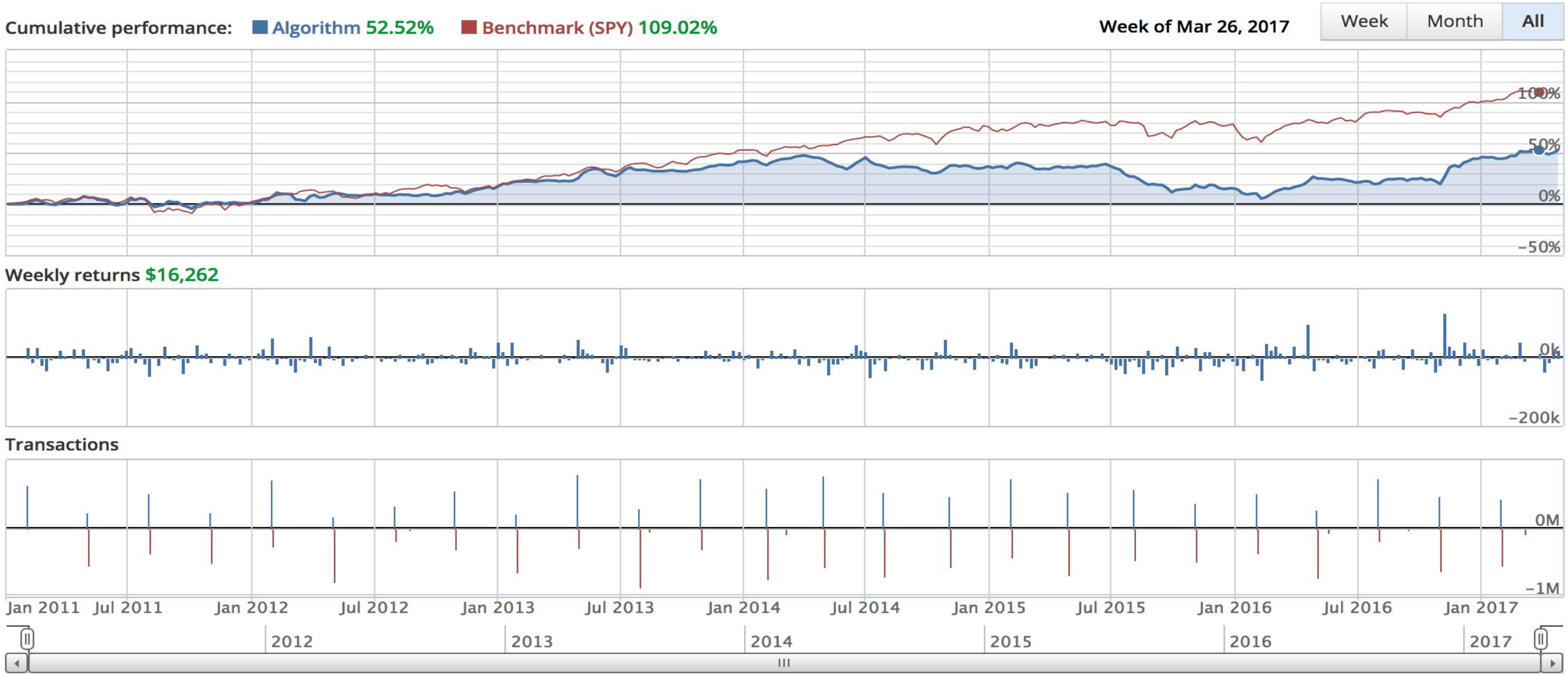
1. **Obtain Earnings Call Transcripts**
2. Apply Language Processing algorithm
3. Run regressions to weight keywords
4. Develop scoring algorithm to determine when to buy/sell
5. Run backtest evaluation of algorithm

Stock Criteria

- **Small**
  - Between the 25th and 75th percentile based on their market capitalization
- **Cheap**
  - Below 50th percentile EV to Ebitda ratio
- **Highly Leveraged**
  - Above mean LT Debt/EV

Benchmark and Relative Performance

Total Returns	Benchmark Returns	Alpha	Beta	Sharpe	Sortino	Volatility	Max Drawdown
52.4%	111.5%	0.01	0.53	0.57	0.85	0.13	-30.4%



Data Collection

- Initial Approach:
  - Process and sort 180,000 html files by file name to group all transcripts for one company together.
  - Merge files together to have one single file for every company.
- Importance of computing power.

```
import os
with open('Companies.txt') as f:
    for line in f:
        line = line.strip()
        str = "*" + line + "*.*txt"
        command = "find . -name \"" + str + "\" -exec mv {} ~/Documents/Stanford/Year_2/MSE_448/Companies/" +
            line + "/ \;"
        os.system(command)
f.close()
```

- Decided to process all files at once and generate one giant CSV.

Data Collection

Apple (AAPL) Q4 2016 Results - Earnings Call Transcript
Oct.25.16 | About: Apple Inc. (AAPL)

Apple, Inc. (NASDAQ:AAPL)
Q4 2016 Earnings Call
October 25, 2016 5:00 pm ET
Executives
Nancy Paxton - Apple, Inc.
Timothy Donald Cook - Apple, Inc.
Luca Maestri - Apple, Inc.
Analysts
Eugene Charles Munster - Piper Jaffray & Co.
Kathryn Lynn Huberty - Morgan Stanley & Co. LLC
Shannon S. Cross - Cross Research LLC
Antonio M. Sacconaghi - Sanford C. Bernstein & Co. LLC
Simona K. Jankowski - Goldman Sachs & Co.
Steven M. Milunovich - UBS Securities LLC
Wamsi Mohan - Bank of America Merrill Lynch
James D. Suva - Citigroup Global Markets, Inc. (Broker)
Rod B. Hall - JPMorgan Securities LLC
Operator
Good day everyone and welcome to this Apple, Incorporated Fourth Quarter Fiscal Year 2016 Earnings Release Conference Call. Today's call is being recorded. At this time, for opening remarks and introductions, I would like to turn the call over to Nancy Paxton, Senior Director of Investor Relations. Please go ahead, ma'am.
Nancy Paxton - Apple, Inc.
Thank you. Good afternoon and thanks to everyone for joining us. Speaking first today is Apple CEO Tim Cook, and he will be followed by CFO Luca Maestri. And after that, we'll open the call to questions from analysts.
Please note that some of the information you'll hear during our discussion today will consist of forward-looking statements, including without limitation those regarding revenue, gross margin, operating expenses, other income and expense, taxes, and future business outlook. Actual results or trends could differ materially from our forecast. For more information, please refer to the risk factors discussed in Apple's Form 10-K for 2015, the forms 10-Q for the first three quarters of fiscal 2016, and the Form 8-K filed with the SEC today along with the associated press release. Apple assumes no obligation to update any forward-looking statements or information which speak as of their respective dates.

Roadmap

1. Obtain Earnings Call Transcripts
2. **Apply Natural Language Processing algorithm**
3. Run regressions to weight keywords
4. Develop scoring algorithm to determine when to buy/sell
5. Run backtest evaluation of algorithm

Word Selection Criteria

- We started analyzing the frequencies of 23 keywords and 22 phrases of interest that could possibly indicate future growth for the company.
- We predict the presence of words and phrases like “cost-cutting,” “deleveraging,” and “debt reduction” in earnings call transcripts will lead these small, highly-leveraged, companies to produce higher returns than the benchmark.

```
WordsOfInterest = ["cost-cut","deleverage","deleveraging","deleveraged","debt-
    reduction","synergy","synergies","acquisition","acquisitions","acquire","acquiring","merger","mergers","paydown","pay-down"
    ,"buyback","buy-back","repurchase","repurchased","repurchasing","tender-offer","dividends","dividend"]

PhrasesOfInterest = ["improved margins","improve margins", "margin improvement", "margin expansion", "eliminate costs", "cost
    cutting", "cut cost", "reduced costs", "reducing costs", "pay down","debt reduction","reducing debt","reduced debt","reduce
    debt","buy back","bought back","tender offer", "revenue growth","debt restructuring","decrease leverage","decreasing
    leverage","pay off"]
```

Language Processing

Methodology:

1. Parse through .txt file to find date and ticker symbol.
2. Account for discrepancies in date format or exchange traded in.
3. Count the number of appearances of all keywords and phrases of interest.
4. Write csv files of organized data for regression analysis.

Data Generated from NLP

Screenshot table containing number of occurrences of all keywords for all transcripts processed:

Date	Ticker	cost-cut	deleverage	deleveraging	deleveraged	debt-reducti	synergy	synergies	acquisition	acquisitions	acquire	acquiring	merger	mergers	paydown	pay-down	buyback	buy-back
11/12/12	LMT	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11/12/12	MOBI	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0
10/15/08	WFC	0	0	0	0	0	0	0	4	5	0	0	6	0	0	0	0	0
11/12/12	GGAL	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11/12/12	NKTR	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10/15/08	HCCI	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0
10/15/08	KO	0	0	0	0	0	0	0	4	6	1	0	0	0	0	0	0	0
11/12/12	CYD	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10/15/08	CSX	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
10/15/08	LUFK	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
10/15/08	PJC	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0
11/12/12	SSRX	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10/15/08	DAL	0	0	0	0	0	1	5	2	0	1	0	11	0	0	0	0	0
10/15/08	JPM	0	0	0	0	0	0	0	2	0	0	0	3	0	0	0	2	0
11/12/12	CWCO	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10/15/08	RLI	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	2	0
10/15/08	ASUR	0	0	0	0	0	0	2	2	1	0	0	0	0	0	0	1	0
10/15/08	STT	0	0	0	0	0	1	0	3	0	0	0	1	0	0	0	0	0
10/15/08	DRH	0	0	0	0	0	0	0	1	0	1	0	0	1	0	0	0	0
11/12/12	AXAS	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
10/15/08	OZRK	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

```
for line in f:
    if "NYSE:" in line:
                i=0
                while i < len(line):
                    if line[i-5:i] == "NYSE:":
                        while i < len(line):
                            if line[i] == ")":
                                break
                            ticker += line[i]
                            i += 1
                        break
                    i += 1
    if "NASDAQ:" in line:
        i=0
        while i < len(line):
            if line[i-7:i] == "NASDAQ:":
                while i < len(line):
                    if line[i] == ")":
                        break
                    ticker += line[i]
                    i += 1
                break
            i += 1
    for month in listOfMonths:
        if month in line and finalDate == "":
            date = line[:9]

            date = str((datetime.strptime(date,'%b.%d.%y')))
            finalDate = date[:10]
    for word in line.split():
        word = word.lower()
        if word[len(word)-1] == ',' or word[len(word)-1] == '.':
            word = word[:len(word)-1]
```

Roadmap

1. Obtain Earnings Call Transcripts
2. Apply Natural Language Processing algorithm
3. **Run analysis to weigh keywords**
4. Develop scoring algorithm to determine when to buy/sell
5. Run backtest evaluation of algorithm

Regressions

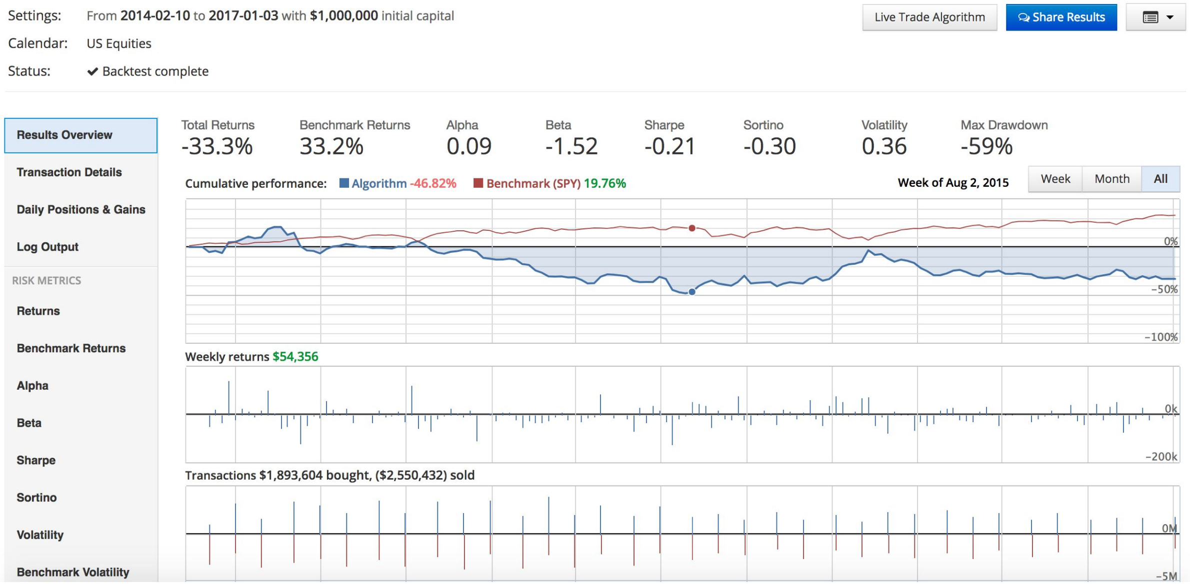
INTRINIO
FINTECH MARKETPLACE

- Used Intrinio Excel add-on to get 1 year forward returns for all stocks within our universe, accounting for dividends and splits.
- Normalize these returns in relation to the Russell-2000.
- Russell-2000 most closely resembles our target market cap

1 Yr. Stock Performance	1 Yr. Russel Performance	Normalized Retrun	Date	Ticker	cost-cut	deleverage	deleveraging	deleveraged	debt-reducti	s
58.96%	34.13%	24.83%	11/13/12	ESLT	0	0	0	0	0	
39.71%	34.13%	5.58%	11/13/12	ACM	0	0	0	0	0	
50.89%	38.60%	12.29%	11/14/12	MTOR	0	0	1	0	0	
27.42%	38.60%	-11.18%	11/14/12	KEM	0	0	0	0	0	
36.50%	38.60%	-2.10%	11/14/12	AMKR	0	0	0	0	0	
39.24%	38.55%	0.69%	11/15/12	KND	0	0	0	0	0	
80.88%	38.55%	42.33%	11/15/12	SB	0	4	3	0	0	
39.82%	38.55%	1.27%	11/15/12	DRYS	0	0	0	0	0	
39.54%	38.77%	0.77%	11/16/12	KND	0	0	0	0	0	
127.09%	43.75%	83.34%	11/20/12	NM	0	0	0	0	0	

First test

- Used a binary variable for the existence words to score the companies (0 if quarterly transcript did not contain keywords, 1 if it did)
- Poor results as expected, with a negative Beta and Sharpe



Regressions

- Ran analysis on individual words and groups of words:
  - Words
    - Revenue Growth
    - Repurchase
    - Margins
    - Dividends
    - Deleverage
    - M&A
    - Synergies
    - Cost-Reduction
  - Groups
    - Dividend/Repurchase
    - Expansion
    - Operation Improvement
    - Deleverage

Words vs returns above Russell 2000

Regression Statistics	
Multiple R	0.1015741
R Square	0.0103173
Adjusted R Square	0.0077475
Standard Error	0.4684713
Observations	3090

ANOVA

	df	SS	MS	F	Significance F
Regression	8	7.049008139	0.881126	4.014874	9.377E-05
Residual	3081	676.1728947	0.219465		
Total	3089	683.2219028			

	Coefficients	Standard Error	t Stat	P-value	Lower 95%	Upper 95%	Lower 95.0%	Upper 95.0%
revenue growth	0.0134749	0.004411086	3.05478	0.002272	0.0048259	0.0221239	0.00482593	0.0221239
repurchase	0.007663	0.003195305	2.398211	0.016535	0.0013979	0.0139282	0.00139787	0.0139282
margins	0.0064945	0.003996664	1.624982	0.104269	-0.001342	0.0143309	-0.0013419	0.0143309
dividends	0.0035383	0.002340868	1.51152	0.130759	-0.001052	0.0081281	-0.0010516	0.0081281
Intercept	0.014172	0.013036416	1.08711	0.277073	-0.011389	0.039733	-0.0113889	0.039733
deleverage	0.005644	0.005346326	1.055677	0.291199	-0.004839	0.0161267	-0.0048387	0.0161267
M&A	0.0009954	0.001114895	0.892788	0.372041	-0.001191	0.0031814	-0.0011906	0.0031814
Synergies	-0.000353	0.00327166	-0.10795	0.914042	-0.006768	0.0060617	-0.006768	0.0060617
cost reduction	-0.040857	0.020627294	-1.98074	0.047709	-0.081302	-0.000413	-0.081302	-0.0004127

Group vs returns above Russell 2000

SUMMARY OUTPUT

Regression Statistics	
Multiple R	0.075041
R Square	0.005631
Adjusted R Square	0.004342
Standard Error	0.469275
Observations	3090

ANOVA

	df	SS	MS	F	ignificance F
Regression	4	3.847342	0.961836	4.367639	0.001596
Residual	3085	679.3746	0.220219		
Total	3089	683.2219			

	Coefficients	andard Erro	t Stat	P-value	Lower 95%	Upper 95%	ower 95.0%	pper 95.0%
dividend/repurchase	0.005147	0.001742	2.954937	0.003151	0.001732	0.008562	0.001732	0.008562
expansion	0.001621	0.000902	1.79641	0.072527	-0.00015	0.00339	-0.00015	0.00339
Operation improvement	0.005953	0.003886	1.532152	0.125587	-0.00167	0.013572	-0.00167	0.013572
Intercept	0.016597	0.012737	1.303098	0.192638	-0.00838	0.04157	-0.00838	0.04157
deleverage	0.005642	0.005331	1.058302	0.290001	-0.00481	0.016094	-0.00481	0.016094

Roadmap

1. Obtain Earnings Call Transcripts
2. Apply Natural Language Processing algorithm
3. Run regressions to weight keywords
4. **Develop scoring algorithm to determine when to buy/sell**
5. Run backtest evaluation of algorithm

Scoring

1. Ranked each transcript by group
2. Weighted each ranking by statistical significance
3. Calculated Sum Product

1	Date	Ticker	divident/repurchase	DR rank	operational improvement	OI rank	Deleverage	D rank	score 1	score 2	score 3	total
80121	2/29/2016	FRO	7	76052	0	1	0	1	40525.34	0.276293	0.190844	40525.81
80122	2/29/2016	GLNG	9	81467	0	1	0	1	43410.8	0.276293	0.190844	43411.27
80123	2/29/2016	GMLP	13	87372	0	1	0	1	46557.36	0.276293	0.190844	46557.83
80124	2/29/2016	SFL	15	88807	0	1	0	1	47322.02	0.276293	0.190844	47322.49
80125	2/29/2016	VET	16	89337	0	1	0	1	47604.44	0.276293	0.190844	47604.9
80126	2/29/2016	RESI	30	91738	0	1	0	1	48883.84	0.276293	0.190844	48884.31
80127	2/29/2016	IPI	0	1	1	64753	0	1	0.532864	17890.79	0.190844	17891.52
80128	2/29/2016	AXGN	0	1	1	64753	0	1	0.532864	17890.79	0.190844	17891.52
80129	2/29/2016	NSTG	0	1	1	64753	0	1	0.532864	17890.79	0.190844	17891.52
80130	2/29/2016	FDML	2	47839	1	64753	0	1	25491.66	17890.79	0.190844	43382.64
80131	2/29/2016	ARES	5	67994	1	64753	0	1	36231.53	17890.79	0.190844	54122.51
80132	2/29/2016	FSS	6	72439	1	64753	0	1	38600.11	17890.79	0.190844	56491.09
80133	2/29/2016	OFIX	6	72439	1	64753	0	1	38600.11	17890.79	0.190844	56491.09
80134	2/29/2016	EBIX	7	76052	1	64753	0	1	40525.34	17890.79	0.190844	58416.33
80135	2/29/2016	FOXF	7	76052	1	64753	0	1	40525.34	17890.79	0.190844	58416.33
80136	2/29/2016	BCPC	0	1	4	87620	0	1	0.532864	24208.78	0.190844	24209.5
80137	2/29/2016	UNFI	0	1	0	1	1	71590	0.532864	0.276293	13662.49	13663.3
80138	2/29/2016	SUN	0	1	0	1	1	71590	0.532864	0.276293	13662.49	13663.3
80139	2/29/2016	TASR	4	62353	0	1	1	71590	33225.64	0.276293	13662.49	46888.41
80140	2/29/2016	STON	5	67994	0	1	1	71590	36231.53	0.276293	13662.49	49894.29
80141	2/29/2016	EVEP	6	72439	0	1	1	71590	38600.11	0.276293	13662.49	52262.87
80142	2/29/2016	ORC	6	72439	0	1	1	71590	38600.11	0.276293	13662.49	52262.87

Algorithm

Insight: By using the correlations found in the Earnings Call Transcript analysis, we can score companies by quarter as new transcripts are released.

Methodology:

- When high frequencies of key terms-groups occur (operational improvement, deleveraging, dividends and repurchases) increment score.
- Buy companies with highest scores.

Roadmap

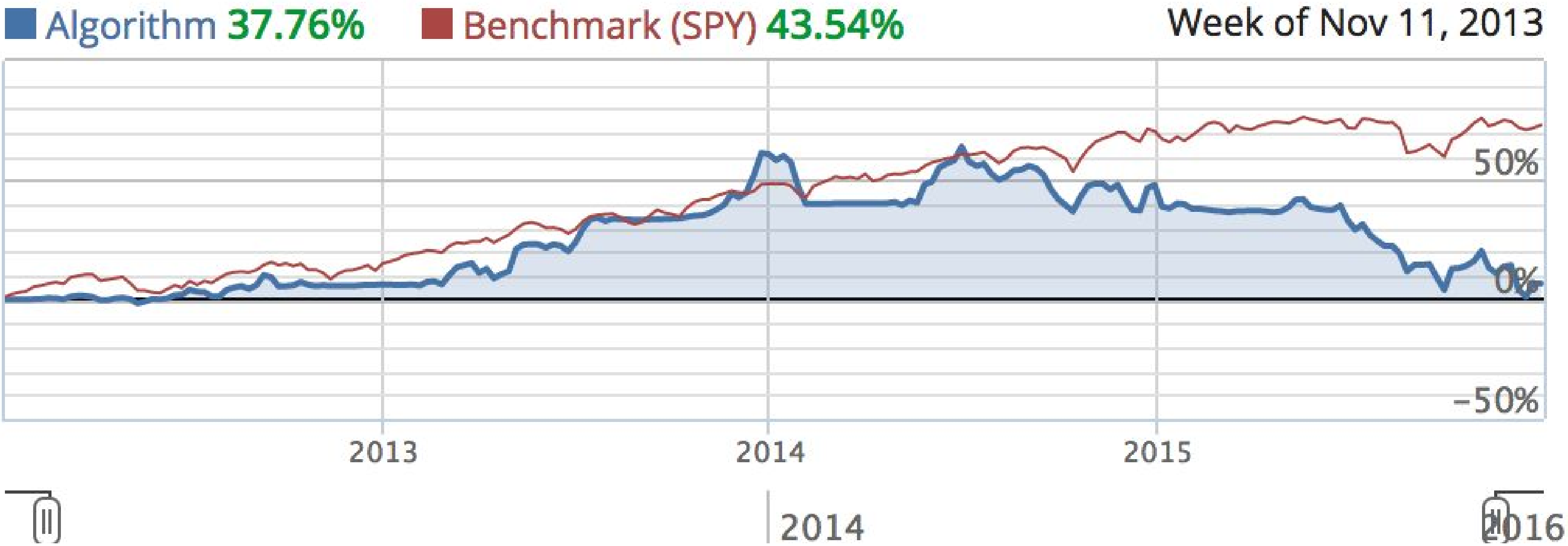
1. Obtain Earnings Call Transcripts
2. Apply Natural Language Processing algorithm
3. Run regressions to weight keywords
4. Develop scoring algorithm to determine when to buy/sell
5. **Run backtest evaluation of algorithm**

1 Year Holding Period

```
for stock in context.security_list:
    if data.can_trade(stock):
        if data.current(stock, 'indicator') > 0:

            last_date_str = str(data.current(stock, 'indicatordate'))
            if(last_date_str != 'nan'):

                last_date = datetime.strptime(last_date_str, "%m/%d/%y")
                today = str(get_datetime(None))
                today_date = datetime.strptime(today[0:10], "%Y-%m-%d")
                difference = (today_date - last_date).days

                score = data.current(stock, 'indicator')
                if score == 4 and difference < 295:
                    order_target_percent(stock, .05)
                elif score == 3 and difference < 190:
                    order_target_percent(stock, .05)
                elif score == 2 and difference < 100:
                    order_target_percent(stock, .05)
                elif score == 1 and difference < 10:
                    order_target_percent(stock, .05)
```

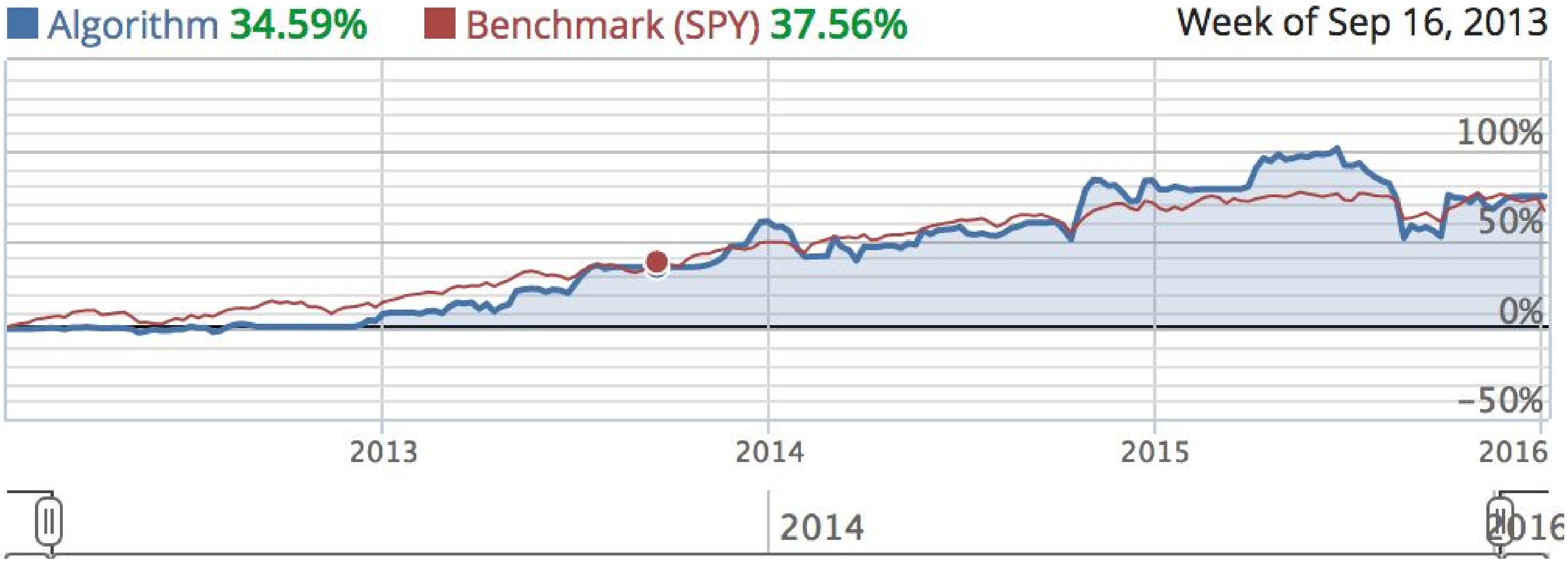
Book Size Too Large

RETURNS	ALPHA	BETA	SHARPE	DRAWDOWN
5.7%	-0.11	1.00	0.17	-40.41%



Stock Performance: 1 Year Holding

RETURNS	ALPHA	BETA	SHARPE	DRAWDOWN
74.1%	0.03	1.00	0.77	-30.8%



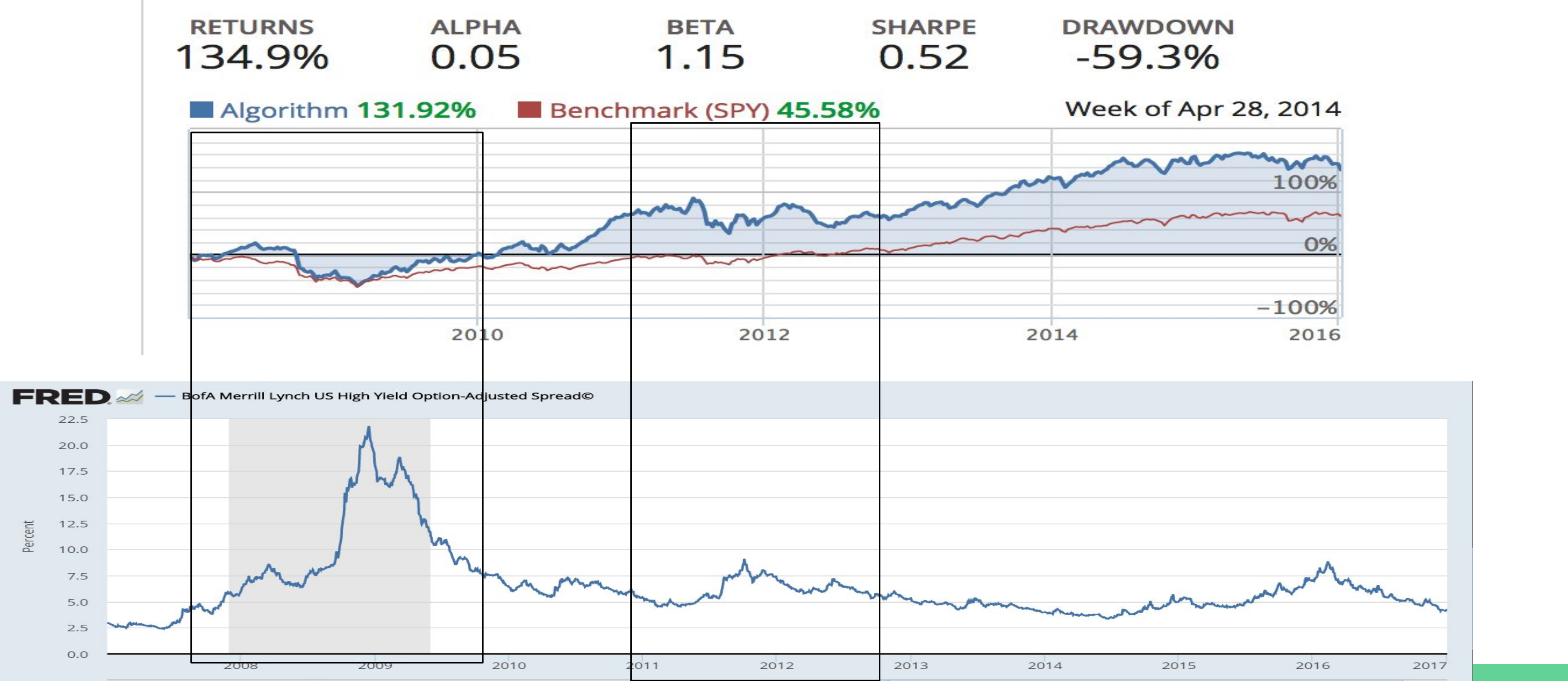
Reactionary

```
        today = str(get_datetime(None))
        today_date = datetime.strptime(today[0:10], "%Y-%m-%d")
        for stock in context.security_list:
            if data.can_trade(stock):
                last_date_str = str(data.current(stock, 'indicatordate'))
                if(last_date_str != 'nan'):

                    last_date = datetime.strptime(last_date_str, "%m/%d/%y")
                    #check if new transcript came out today
                    if (today_date-last_date).days == 0:
                        new_score = data.current(stock, 'indicator')
                        if new_score < data.current(context.min_stock,
'indicator'):
                            order_target_percent(context.min_stock, 0)
                            context.stock_list.remove(context.min_stock)
                            order_target_percent(stock, 1.0/30)
                            context.stock_list.append(stock)
                            updateMin(context, data)
```

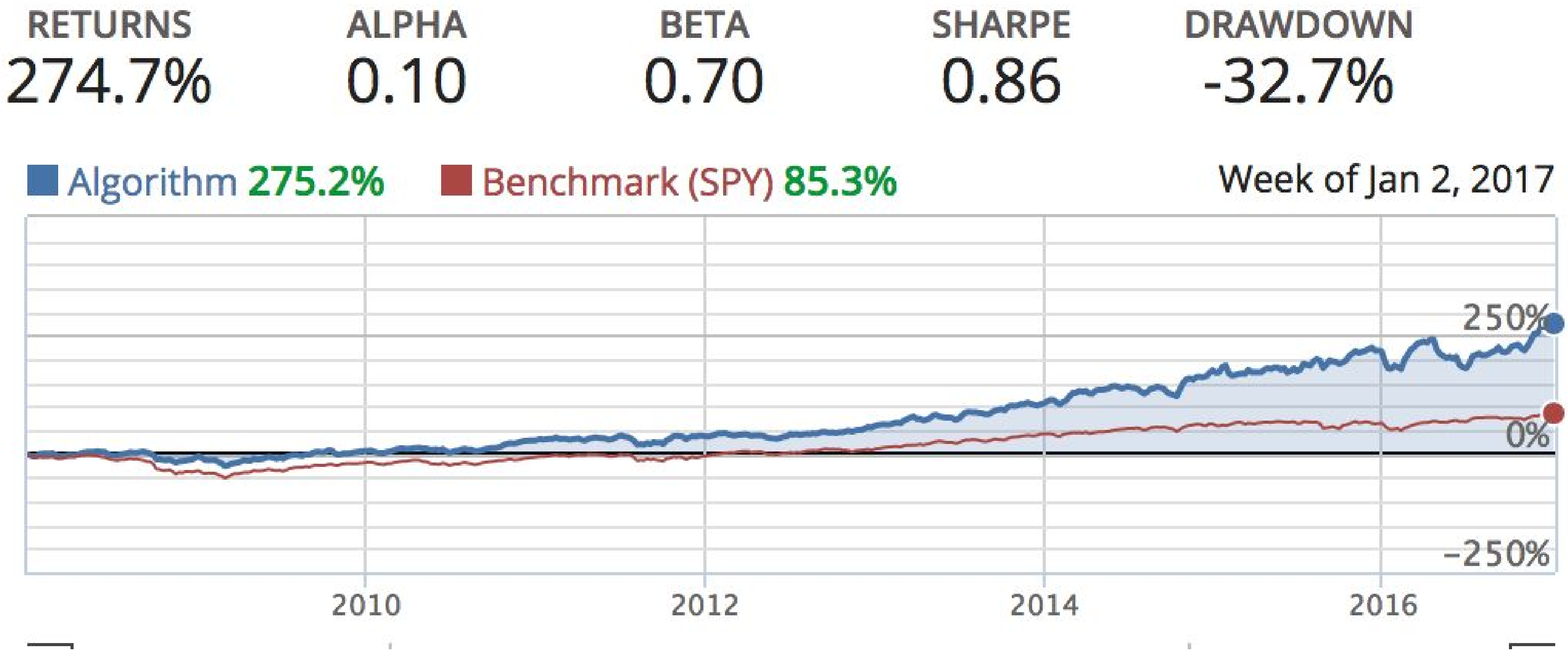
Stock Performance: Reactionary

RETURNS	ALPHA	BETA	SHARPE	DRAWDOWN
141.6%	0.04	0.92	1.19	-15%

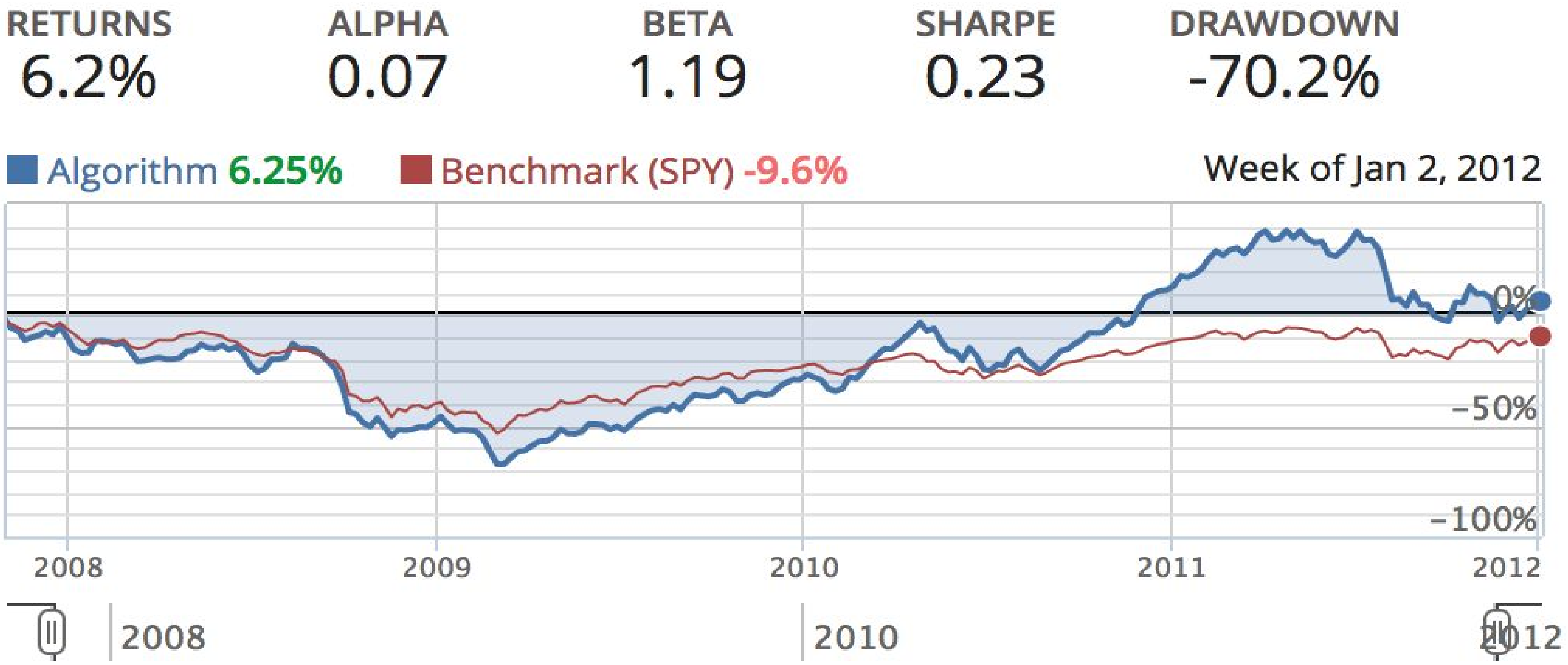
Top 30 Basket, Daily Shuffle



Top 10 Basket, Daily Shuffle

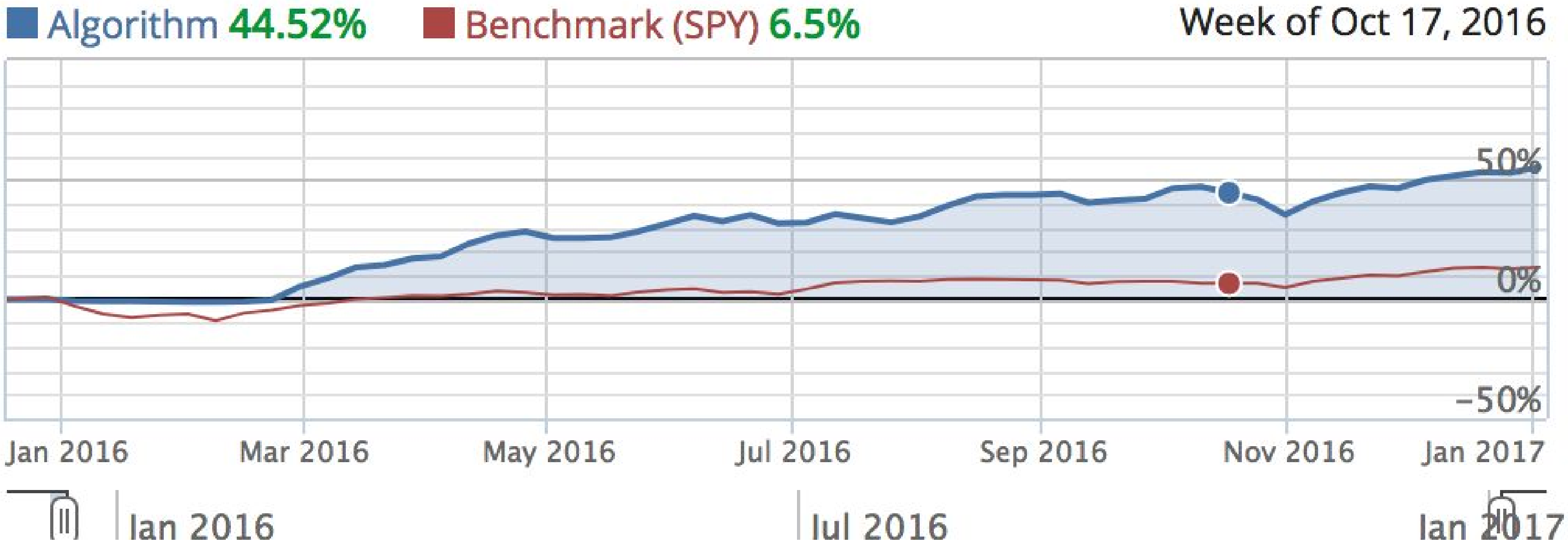


Top 10 Basket, Daily Shuffle, Out of Sample Period



Top 10 Basket, Daily Shuffle, Out of Sample Period

RETURNS	ALPHA	BETA	SHARPE	DRAWDOWN
56.1%	0.34	0.82	2.14	-9.5%



Issues we encountered

- Getting the data
- Filling holes in the data
- Starting with the incorrect universe
- Data is not standardized and is difficult to work with
- Working with Quantopian API, fetching csv, ensuring that the data we are trading on was available,

Further Improvements

- Decay scores based upon the time from the announcement
- Stop loss orders to minimize max drawdown
- Potential implement shorting strategy to minimize market correlation
- Experiment with different basket sizes
- Consider ways of expanding universe to make larger book sizes possible
- Update scores based on difference from previous score (0004)

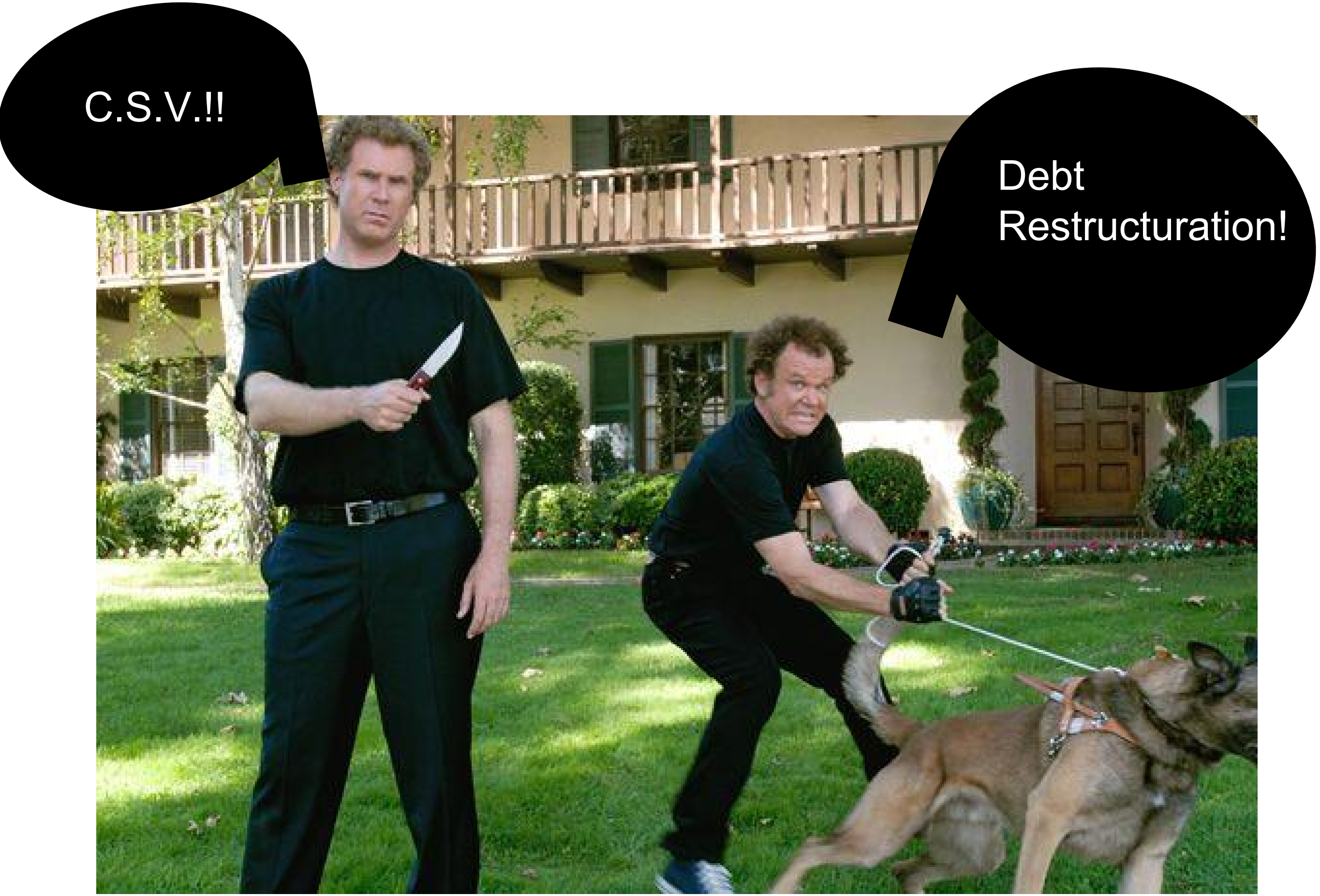
What We Learned

- There's potential to use earnings call transcripts for indicators for a trading strategy
- Need more data for backtesting
- Murphy's Law
    - Cleaning Data

Thank You!

Appendix 1:

Jonny and Juanpa working on our NLP algorithm last night!


C.S.V.!!
Debt Restructuration!