PRECISION ALPHA

White Paper

Abstract

Precision Alpha is an alternative data company providing predictive analytics for financial markets. Precision Alpha's algorithm forecasts daily moves for all stocks listed on multiple exchanges globally.

The following paper analyses Precision Alpha's dataset, from exploratory data analysis, to machine learning prediction and designing a trading strategy. Finally, the robustness of the trading strategies is assessed, and we benchmark their performance versus various indices.

Paper authored by Prince Analytics LLC

info@princeanalytics.com

SUMMARY

In this paper, we assessed the application of Precision Alpha's dataset to trading daily the 150 most liquid stocks on the NASDAQ, between November 2016 and August 2019.

In the Exploratory Data Analysis section, we analyzed the distribution and links between variables. Some variables are functionally related, such as proba_up and emotion, power. Most variables are non-normally distributed.

In the predictive modeling section, we found that the dominant probability ("proba_up"), had predictive power over future returns. Then in the Machine Learning section, we confirmed that proba_up had indeed the best accuracy (53%) as it relates to predicting the sign of future returns, closely followed by emotion. Other metrics such as power, resistance and noise had less significant results. As we tested other derivative features, we found that 1-week change in probabilities also had significant predictive power.

In the trading strategy section, we backtested directional (time-series) and market-neutral (cross-sectional) strategies using proba_up as the signal. Trading Strategies are profitable with Sharpe Ratios of 2.4 and 3.1 respectively, significantly outperforming the market. We found that the trading strategies were robust to various assumptions about trading costs, execution prices, portfolio construction.

Finally, we showed that the trading strategies generated substantial alpha over their respective markets, as well as outperformed major asset class and hedge fund indices for the considered period.

Table of Contents

1. Data description

a. Inputs

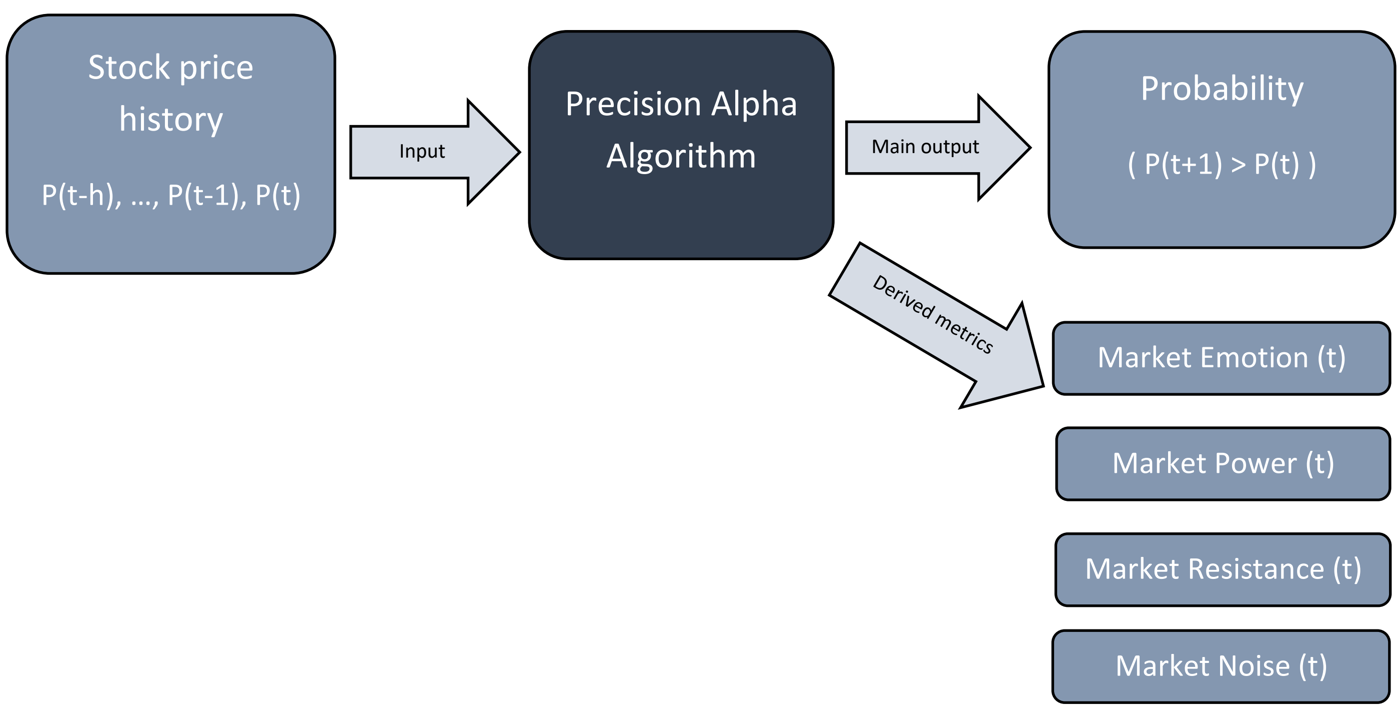
The inputs of the model are historical daily close market prices for securities.

b. Process and output

Precision Alpha uses a proprietary, physics-inspired algorithm using Machine Learning to predict probabilities of a stock price ***P(t)*** going up or down on the next day, based on the past h days of data. Additionally, it derives four metrics defined as:

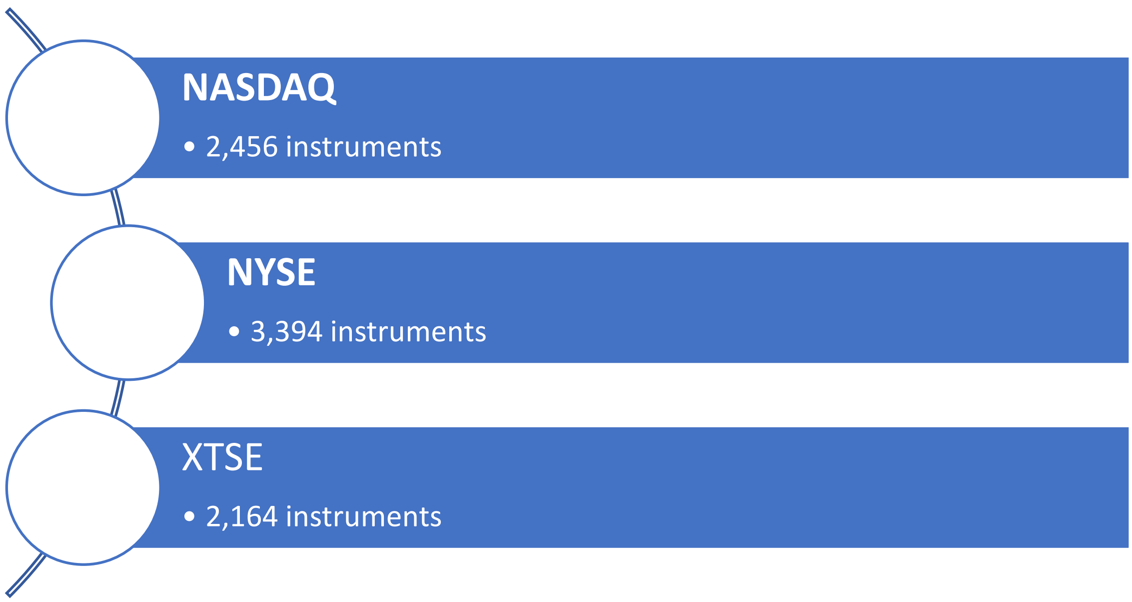
- **Market Emotion:** Behavioral Energy measured from the equilibrium energy. *Measured in Millivolts.*
- **Market Resistance:** Market resistance to changing price. Same as market viscosity. *Measured in ohms.*
- **Market Power:** Power is the rate (energy amount per time period) at which work is done or energy converted to price movement. Market power combines Emotion and Resistance. Power is equal to Emotion squared divided by R, that is, V^2/R). Power is zero at equilibrium. *Measured in microWatts*.
- **Market Noise: (**Nyquist) noise that dissipates Behavioral Energy so that it is not used to generate price movement. *Measured in dBs.*

Those four metrics might contain additional information about future market price moves.



c. Coverage

The markets covered so far are all US instruments listed on the NASDAQ, New York Stock Exchange (NYSE), and Canadian stock on the Toronto Stock Exchange (XTSE). By instruments, it includes stocks, depository receipts, funds, preferred shares and structured products.



In this report, **all analytics will be done using a data sample for stock listed on the NASDAQ**.

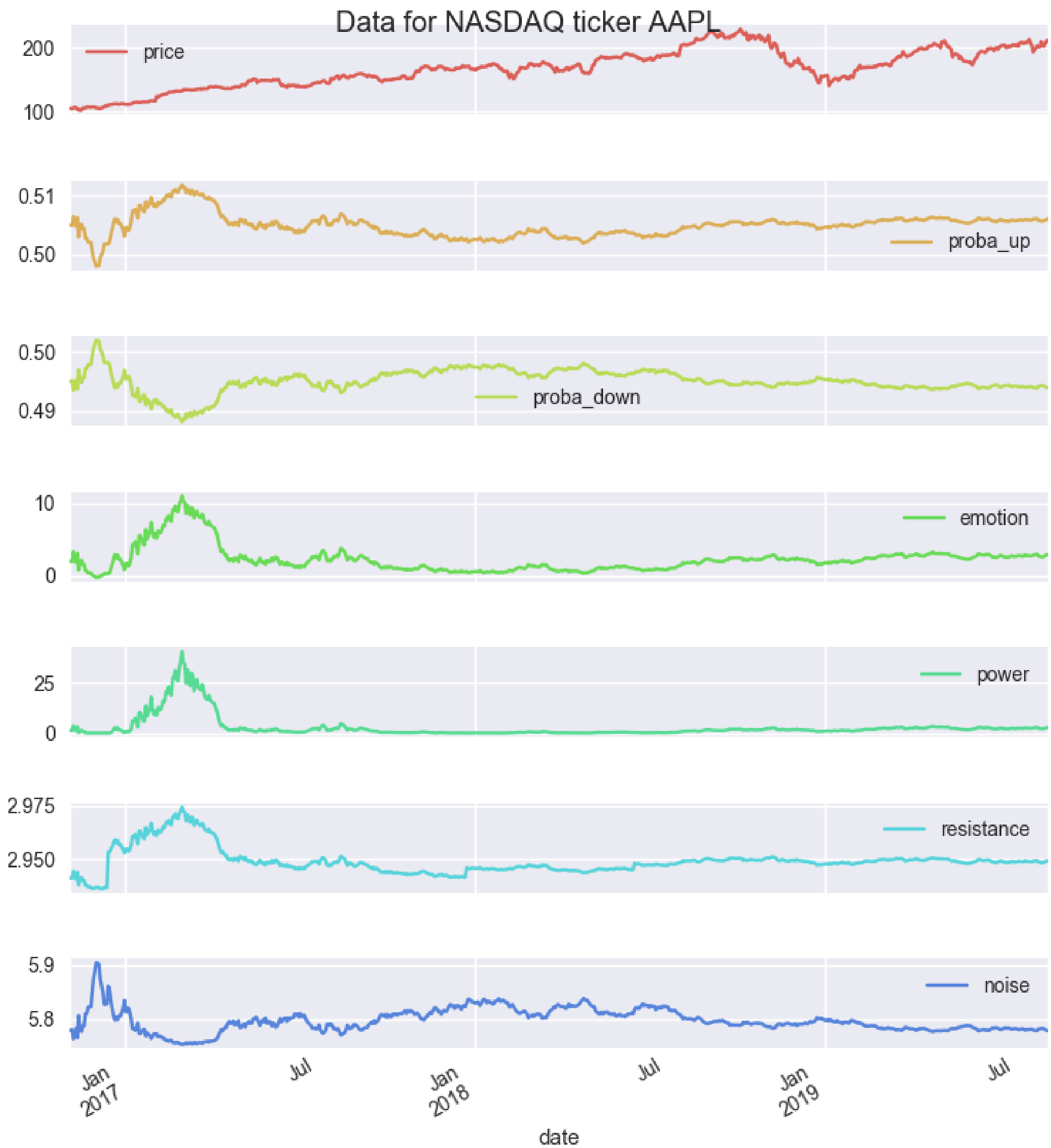
2. Exploratory data analysis

a. Example data for one stock: Apple

Below we plot the daily closing share price (in $US) for NASDAQ's stock Apple (Bloomberg ticker: "AAPL"), alongside all output from Precision Alpha's algorithm. Note that ***proba_down*** = 1- ***proba_up***, where ***proba_up*** is the probability that the share price goes up on the next day.

We can see that the stock has been trending up over time, and as such, the fact that the system predicts up moves (***proba_up*** > ***proba_down***) most of the sample makes sense. We can see how ***emotion*** can be positive or negative, while the other 3 metrics (***resistance***, ***power*** and ***noise***) are all always 0 or positive. We also note that ***power*** seem to have sharp rises and falls, while ***resistance*** has some discontinuities.

We will in the next sections analyze more in detail the distribution of each of those variables.



b. Overall dataset

The overall dataset we will use for this paper covers the NASDAQ.. Out of the thousands of stocks listed on that exchange, we have selected *the 150 most liquid stocks* (liquidity is defined as average daily volume over the past 30 days as of August 2019). We have daily data from January 2017 to August 2019, that is about 750 datapoints. We will load and analyze 7 variables: price, ***proba_up***, ***proba_down***, ***emotion***, ***resistance***, ***power***, ***noise***. As such, the datasets contain around 800,000 datapoints.

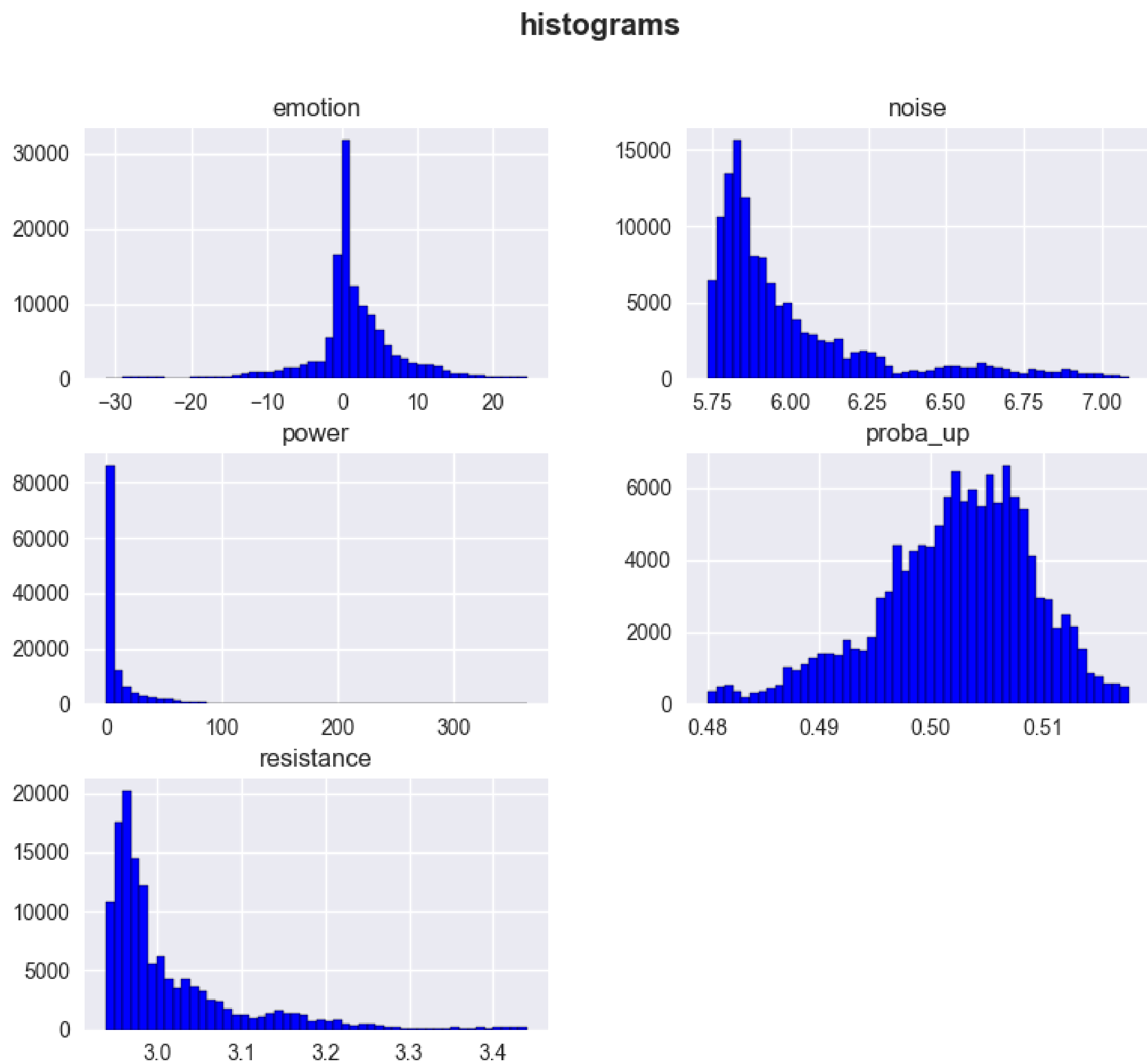
c. Panel distributions

By pooling all stock times series into a panel (i.e. just concatenating the history all each stock into a single dataframe), we can observe the historical distribution of the metrics. We output the total number of datapoints ("count"), mean and standard deviation, as well as minimum, maximum values and 25^{th}, 50^{th} (median) and 75^{th} percentiles.

Probabilities (***proba_up*** and ***proba_down***) have a median value of 50%, which makes sense as over time the system should be balanced and predict roughly as many up moves as down moves. We can also observe the median order of magnitude for the 4 metrics: 0.7, 1.7, 3 and 5.9 for ***emotion***, ***power***, ***resistance*** and ***noise*** respectively. We can also see that emotion is the only variable that can be positive or negative. ***Price*** is just the daily stock prices at the close.

	price	proba_up	proba_down	emotion	power	resistance	noise
count	133,751	133,751	133,751	133,751	133,751	133,751	133,751
mean	55.8	50%	50%	1.2	24.5	3.0	6.0
std	137.1	1%	1%	8.6	105.8	0.1	0.3
min	0.1	46%	47%	-122.3	0.0	2.9	5.7
25%	11.6	50%	49%	-0.3	0.1	3.0	5.8
50%	27.7	50%	50%	0.7	1.7	3.0	5.9
75%	59.5	51%	50%	4.0	12.2	3.0	6.1
max	2220.0	53%	54%	65.2	4565.1	3.8	8.3

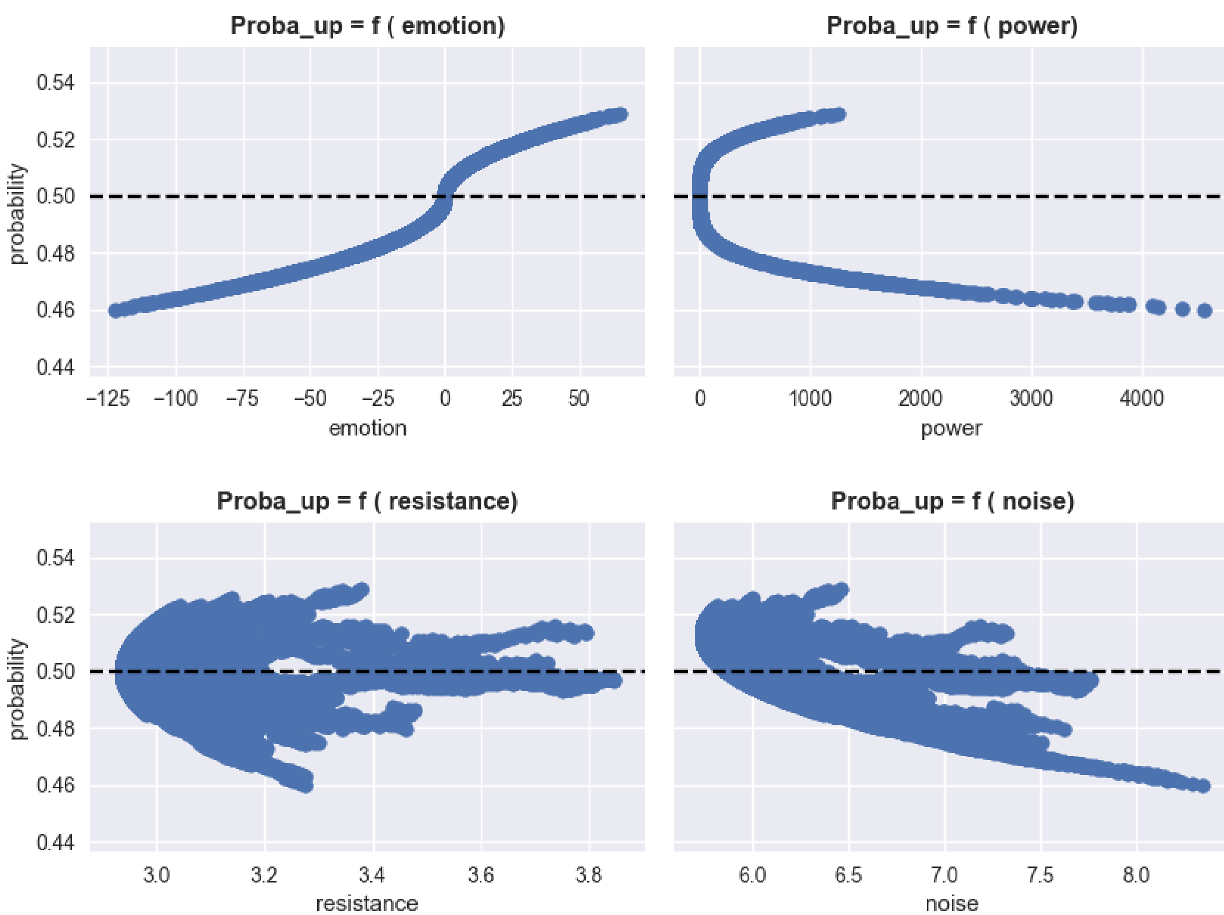
Another way to visualize the distribution is to plot the histograms of variables. We can see that ***emotion*** has a fairly symmetric distribution, although with much slimmer "belly" and fatter "tail" than a normal distribution. ***Power*** has more of a Chi-squared type distribution, with most values being equal to 0. ***Noise*** and resistance seem to have a more Fréchet distribution profile. More work would be required both to fit an appropriate distribution to those variables, and also understand why they follow those distributions. In any case, the main point we are making here is that, apart from ***proba_up***, **these metrics are not normally distributed**, as is often assumed in quant finance.



d. Proba_up as a function of all 4 metrics

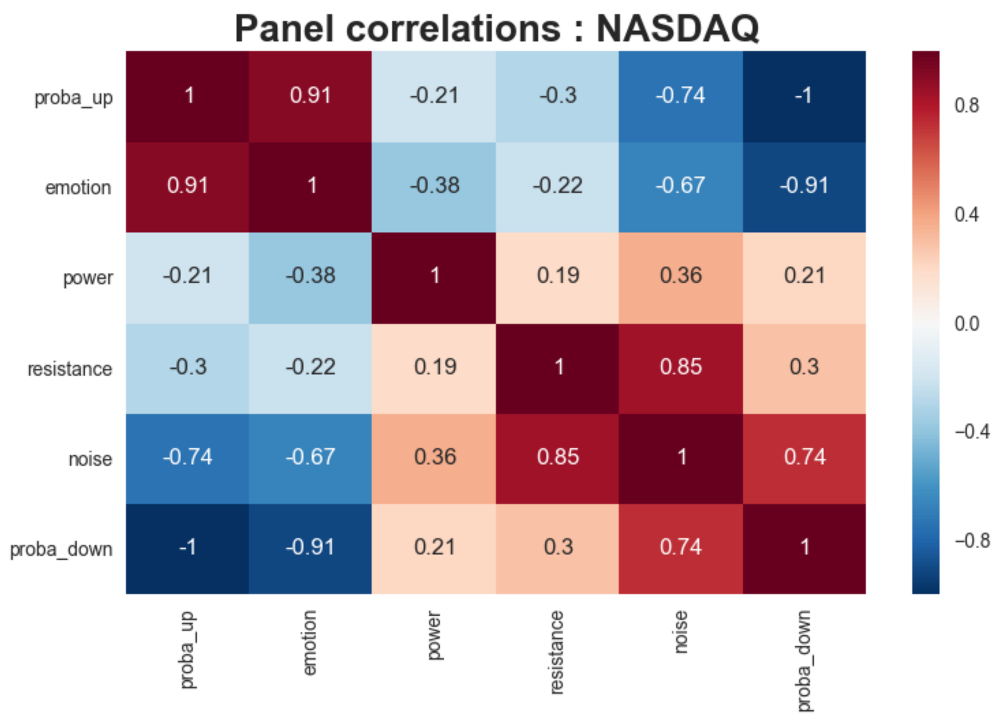
If we plot all 134,000 datapoints for ***proba_up*** versus all 4 metrics: ***emotion***, ***power***, ***resistance*** and ***noise***, we can see that it has a direct functional relationship to ***emotion*** and ***power***. As ***emotion*** is negative, ***proba_up*** will be below 50% (and ***proba_down*** will thus be above 50%), and as ***emotion*** is positive, ***proba_up*** will be above 50%. Thus, the sign of ***emotion*** directly represents whether the model believes stock will go up (***emotion***>0) or down (***emotion***<0) over the next day. ***Power***, on the other hand, is symmetric. High ***power*** will mean higher probability of a move, either up or down. ***Resistance*** and ***noise***, on the other hand, seem to have little functional relationship to ***proba_up***.

We have established that 2 of the 4 metrics have a direct functional relationship to proba_up, although those relationships are highly non-linear. This will be relevant to put the upcoming Machine Learning modeling into perspective.



e. Panel correlations

Next, we can also run the panel correlation for all stock in the Nasdaq. The correlation between ***proba_up*** and ***proba_down*** is -100%, which makes sense. Of note, ***emotion*** and ***proba_up*** are highly correlated, which is in line with the findings of the previous section. ***Power***, as seens in section d., has a square relationship to emotion (as ***power*** = ***emotion^2***/***resistance***), but has most of its observations for ***proba_up*** being below 50% (and thus ***emotion*** being negative), thus showing a negative correlation with ***proba_up***. ***Noise*** and ***resistance*** have a high correlation (85%), and tend to have low to negative correlation with ***proba_up***, and should thus be seen as more contrarian indicators.


Panel correlations : NASDAQ
proba_up
emotion
power
resistance
noise
proba_down
1
0.91
-0.21
-0.3
-0.74
-1
0.91
1
-0.38
-0.22
-0.67
-0.91
-0.21
-0.38
1
0.19
0.36
0.21
-0.3
-0.22
0.19
1
0.85
0.3
-0.74
-0.67
0.36
0.85
1
0.74
-1
-0.91
0.21
0.3
0.74
1
0.8
0.4
0.0
−0.4
−0.8

3. Predictive modeling

In this section of the paper, we focus on assessing the predictive power of ***proba_up*** (the dominant probability) and other metrics from the dataset, over future stock price returns.

a. Predictive power for each feature

We start by computing:

$$\rho(\text{variable}(t), \text{return}(t \rightarrow t+1))$$

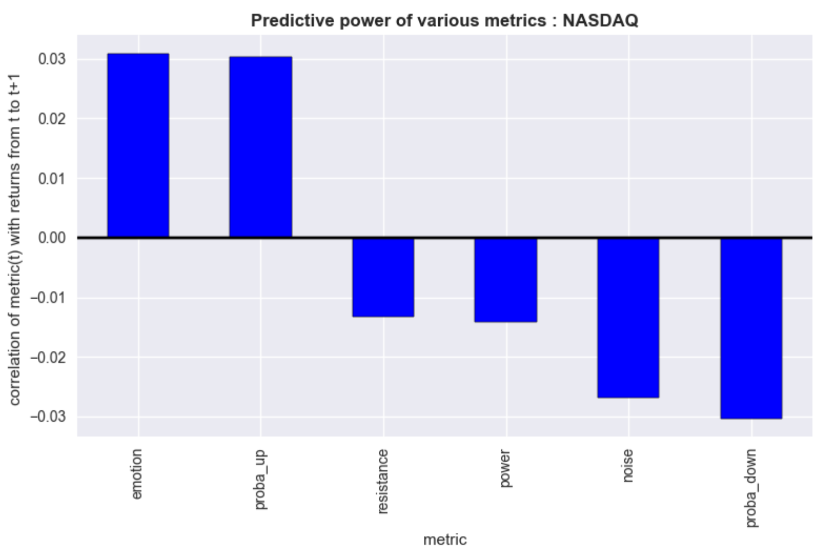
For variable(t) = proba_up(t), proba_down(t), emotion (t), power(t), resistance (t), noise (t)

Where:

$$\text{return}(t \rightarrow t+1) = \log\left(\frac{\text{price}(t+1)}{\text{price}(t)}\right)$$

Price(t) is the closing price of each stock at date t. The system outputs probabilities and other metrics at each close of market t. We assume that we can enter trades as of close of market t as well.

Proba_up, emotion have the highest predictive power, as expected. Indeed, as we saw in the Exploratory Data Analysis, those two measures are highly correlated as they are function of each other. ***Power*** has lower predictive power, which makes sense as it is a derived metric of ***emotion*** and ***resistance***. ***Noise*** has nearly as much predictive power on the downside as ***proba_down***.



b. Panel regression using proba_up: lead/lag

We compute:

$$\rho\left(\text{proba}_{\text{up}}(t), \text{returns}(t+i \rightarrow t+i+1)\right)$$

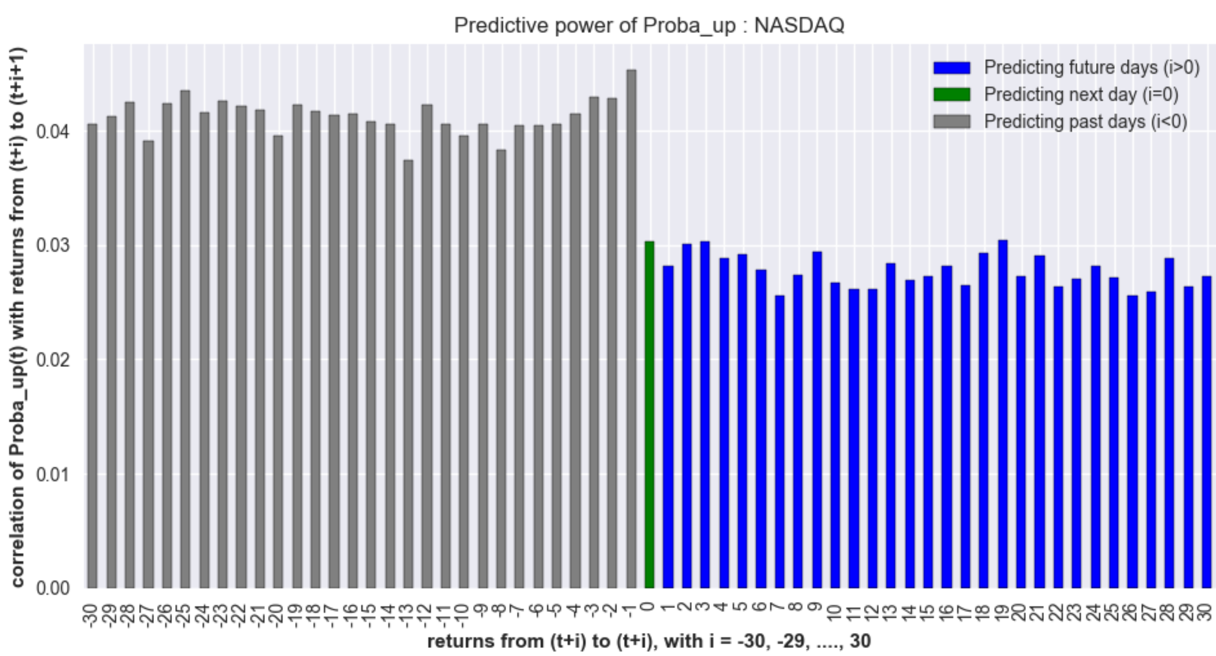
For i = -30, -29, -28,…, 29, 30 business days

Where

$$\text{returns}(t+i \rightarrow t+i+1) = \log\left(\frac{\text{price}(t+i+1)}{\text{price}(t+i)}\right)$$

i.e. we want to assess the lead/lag correlation between the probability of a future stock return being positive and its actual future return. We also consider negative value of i to check how much ***proba_up*** is dependent on past returns (i.e. is it a momentum, or more mean-reversion type of signal).

As we can see from the result below, the ***proba_up*** metric has positive predictive power over future returns. **The predictive power stays constant even when forecasting returns several days ahead (i>>0)**. ***proba_up*** **also has positive correlation with past returns, which implies that it reflects the price momentum inherent to the security.**



4. Machine Learning modeling

a. Introduction

In the following section, we will use various machine learning techniques to assess the predictive power of Precision Alpha's dataset over future returns.

We define the *target variable* ***Y(t)*** as the sign of the next day return of the stock:

$$Y(t) = \mathrm{SIGN}\left(\log\frac{\mathrm{price}(t+1)}{\mathrm{price}(t)}\right)$$

We choose to use the sign function to remove the magnitude of returns, and instead focus on their direction. That way, we remove biases potentially arising from outlier returns, and we avoid giving more emphasis to stock with higher volatility and thus higher returns. **The prediction problem becomes a classification problem**.

The **features** $X_i(t)$ will be derived from ***proba_up***, ***emotion***, ***resistance***, ***power***, ***noise***.

b. Machine learning metrics using feature $\mathbf{X_{probaup}}$

We define $X_{probaup}$ as:

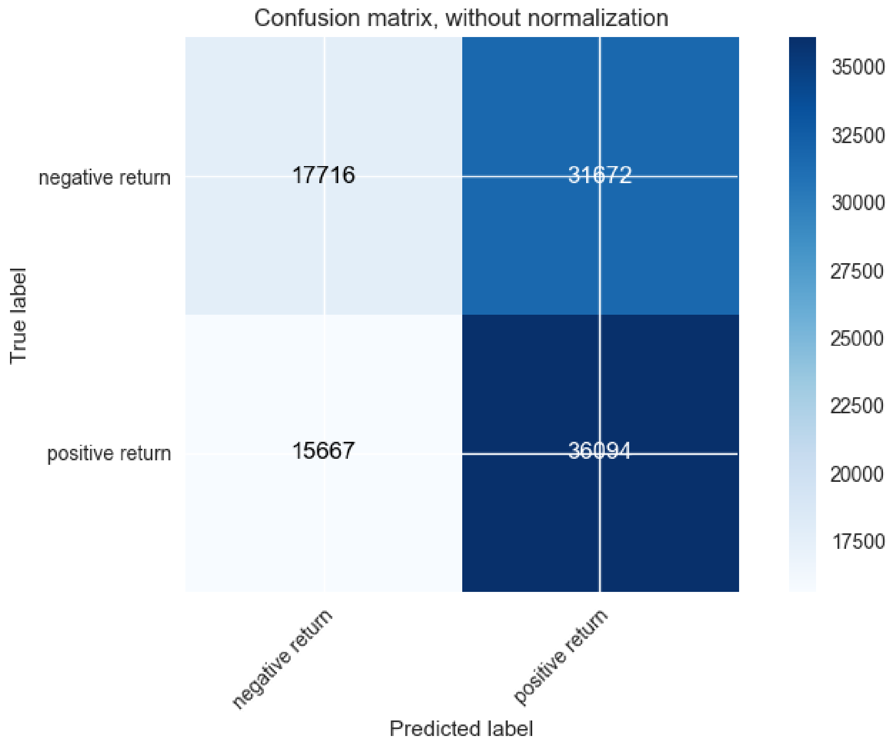
$$X_{probaup}(t) = \mathrm{SIGN}\left(\mathrm{Proba_{up}}(t) - 0.5\right)$$

And in this case we are simply assuming that $X_{probaup}(t)$ predicts the next day return, our target variable, i.e. :

$$Y(t) = X_{probaup}(t) \Leftrightarrow \mathrm{SIGN}\left(\log\frac{\mathrm{price}(t+1)}{\mathrm{price}(t)}\right) = \mathrm{SIGN}\left(\mathrm{Proba_{up}}(t) - 0.5\right)$$

Confusion matrix

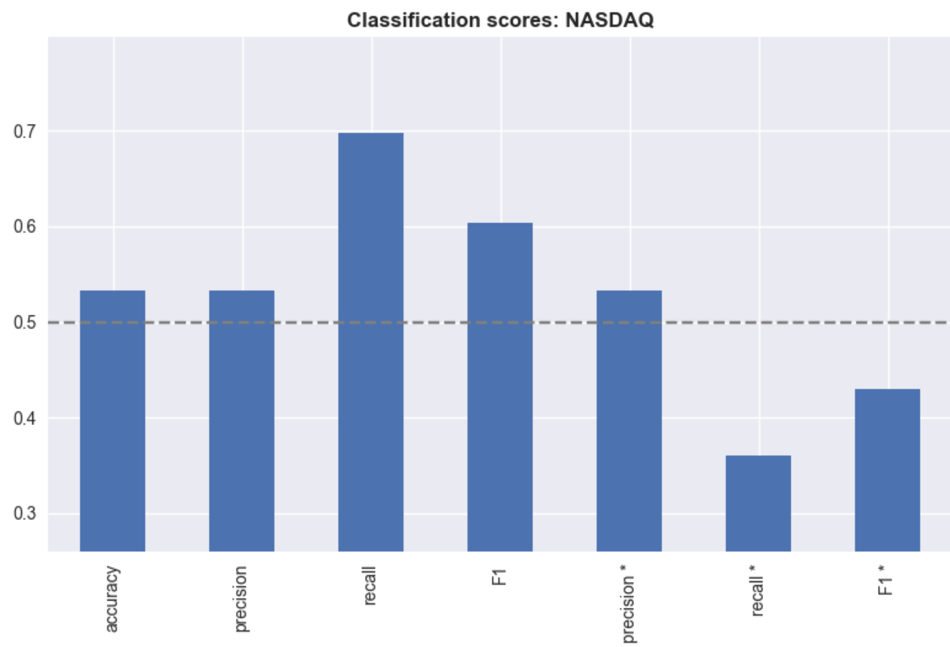
We compute the confusion matrix for the full NASDAQ sample (101,000 business day datapoints). We can see that the ***proba_up*** metric from Precision Alpha's algorithm predicts positive returns most of the time, although this might be dependent on the particular period (2017 to 2019) we consider. Next, we are going to breakdown the confusion matrix into various standard classification metrics, such as accuracy, precision and recall.



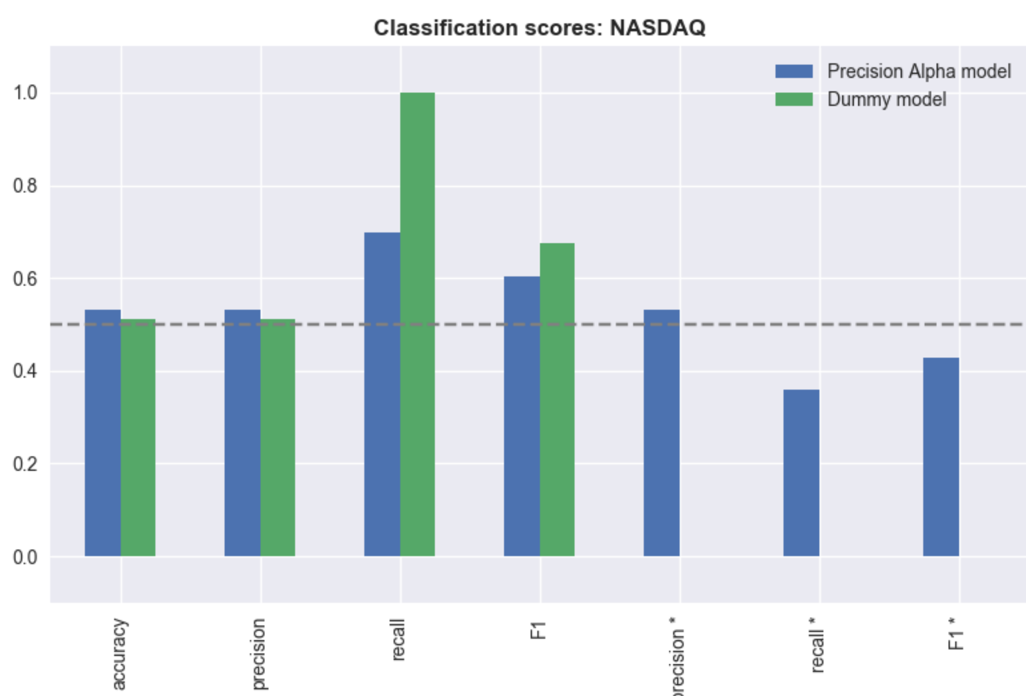
Classification scores: Accuracy, Precision, Recall, F1

The accuracy of the entire system is 53%. However, we can break down the precision, recall, F1 scores for when predicting positive returns and negative returns (**noted with a "*").**

The **precision** (True positives / Total predicted positives) is good (53%) when predicting positive and negative returns. However, we can see that the **recall** (True positives / Actual positives) is actually much worse when predicting negative returns (36% versus 70%). This is not a big problem when trading, as recall really represent a "regret" of not having captured a trade that could have been profitable. **In our case, it will mean that a trading system based on Precision Alpha's algorithm might not capture all the shorting opportunities available in the market (low recall), but out of all the opportunity it actually trades, it will have a good hit ratio (i.e. good precision).**



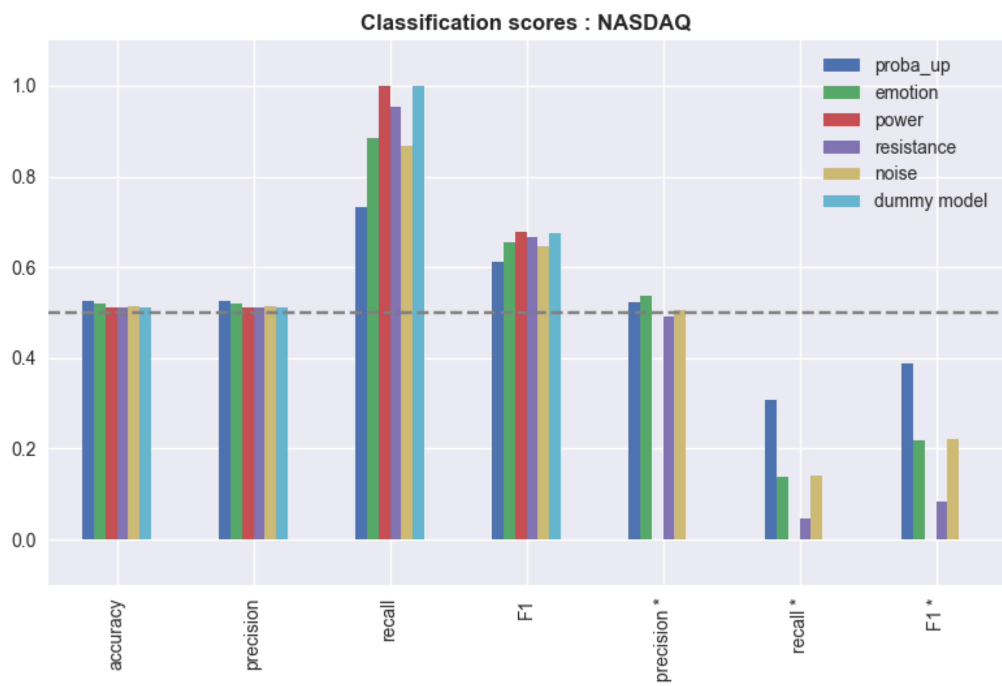
We can then put those numbers in perspective, by comparing them to a "dummy" model, that simply forecasts the target variable to be always positive, as $\mathrm{Y(t)} = 1$ (positive returns) is the dominant class from the sample. **It is good to note that the accuracy of Precision Alpha's Model is higher than the 50% accuracy of the dummy model**. As the dummy model always predicts positive returns, it will capture all positive returns, and thus the recall is 100%. However, as it never predicts negative returns, all scores for the negative return class (*) are 0%. *Note that the calibration of the dummy model is very sample dependent, and we have here a fairly short 3-year data sample.*



c. Each metric as a separate model

Rather than using X_{probaup} as a predictor of the sign of future returns, we can also develop 4 separate models that use each of the 4 metrics (***emotion***, ***power***, ***resistance***, ***noise***) to try to predict $Y(t)$.

We fit a **logistic regression model** for each of those 4 metrics, and observe the various classification scores.



Focusing on the **precision** for both positive and negative (*) returns, as well as **accuracy**, we can see that only ***proba_up*** and ***emotion*** seem to have a significantly higher predictive power than using a dummy model.

d. Combining all metrics into a single model

We can now try to combine the 5 metrics as features in a multivariate model:

$$X_1(t) = \text{proba\_up}(t)$$

$$X_2(t) = \text{emotion}(t)$$

$$X_3(t) = \text{power}(t)$$

$$X_4(t) = \text{resistance}(t)$$

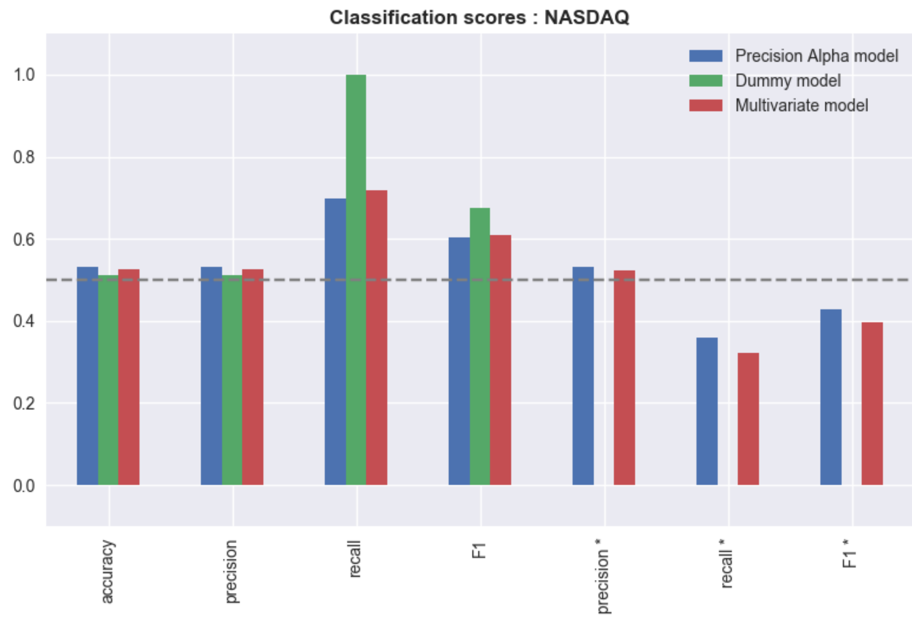
$$X_5(t) = \text{noise}(t)$$

Note that as variables have various units, we should normalize all those 5 features to a Z-score, i.e. 0 mean 1 variance variable, by doing:

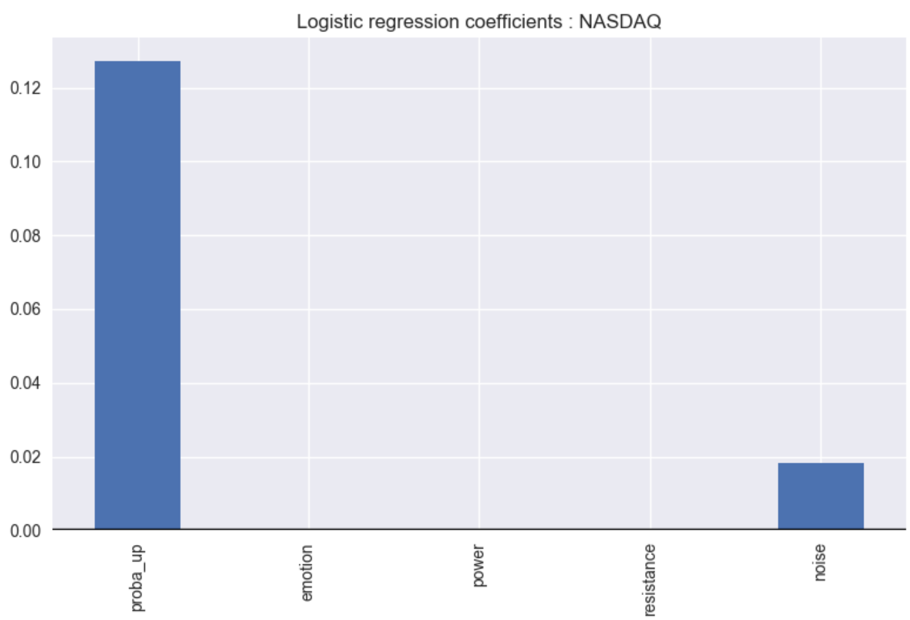
$$Z(X_i(t)) = \frac{X_i(t) - \text{mean}(X_i)}{\text{std}(X_i)}$$

However, as many metrics are highly skewed and non-normal, we use the so-called **robust scaling method**, where the mean is replaced by the median, and the standard deviation by the 75^{th} -25^{th} quantile range.

We then fit a logistic regression model on the full dataset, and compare the results of this multivariate model with simply using ***proba_up*** (i.e. the "Precision Alpha" model). We can see that **the overall accuracy and other scores do not meaningfully improve.**



We also output the regression coefficients: we observe that ***proba_up*** is the largest contributor to prediction. Thus, it confirms that ***proba_up*** is the best predictor of returns, even in a multivariate case.



e. Feature engineering

So far, we have simply used the level of each metric as they were provided in the dataset. We will now add 5 additional features into the multivariate model, where those features are the 5-day change in those variables:

$$X_6(t) = \Delta_{5day}(\text{proba\_up}(t))$$

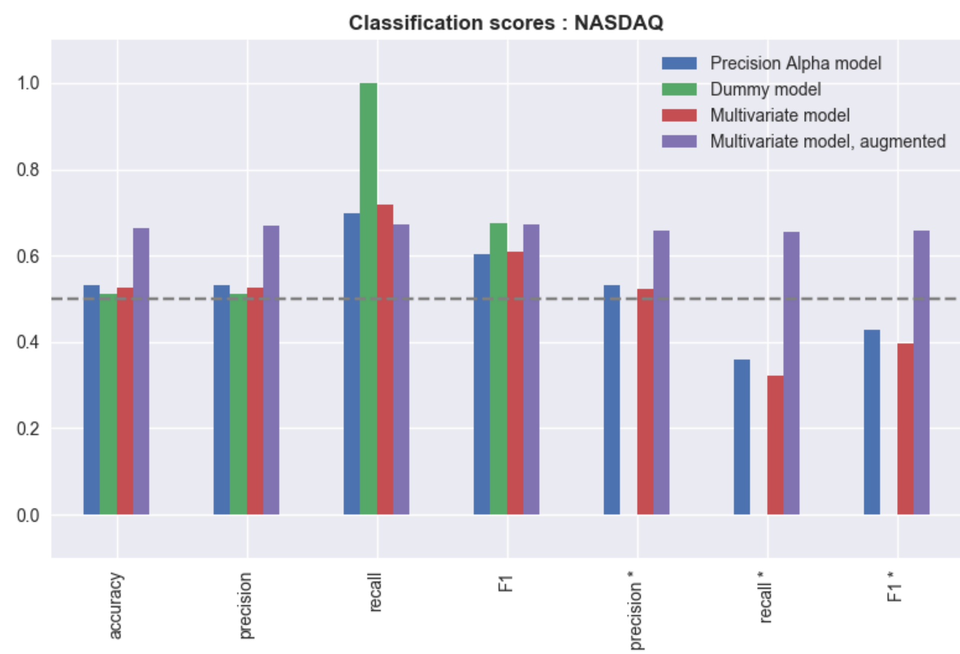
$$X_7(t) = \Delta_{5day}(\text{emotion}(t))$$

$$X_8(t) = \Delta_{5day}(\text{power}(t))$$

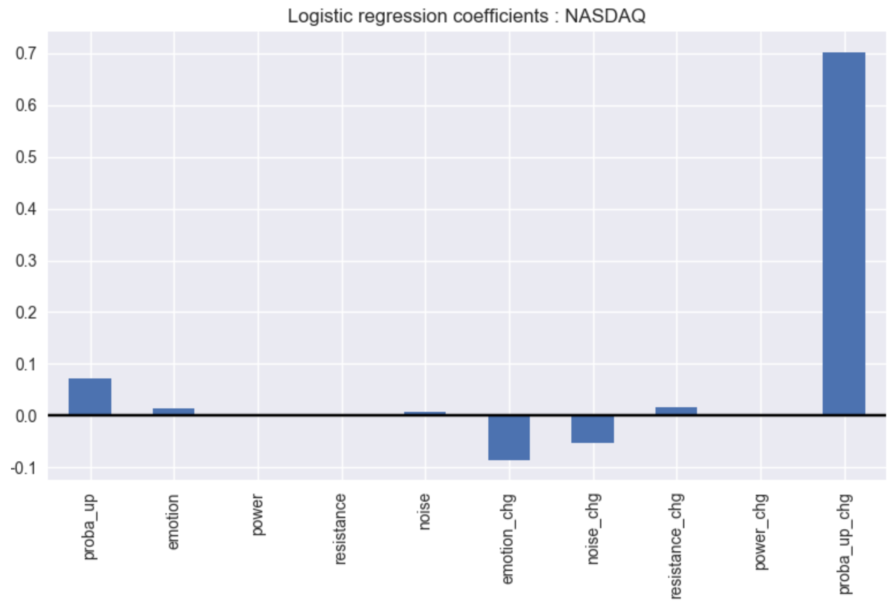
$$X_9(t) = \Delta_{5day}(\text{resistance}(t))$$

$$X_{10}(t) = \Delta_{5day}(\text{noise}(t))$$

We can now compare the classification score of this augmented model with 10 features, with the previous multivariate model that had only 5. We can see in this case that **adding those extra variables does improve results on nearly all scores**.



Looking at the regression coefficients, it appears that the 21-day change in proba_up (***proba_up_chg***) has the largest contribution to prediction, thus **suggesting that looking at changes in probability might give a better predictive power and thus trading profitability than the probability level.**



There is obviously an infinite amount of transformation of variables that we can fit into a model. In the interest of time, and to keep the paper short, we will let the readers investigate other transformations in their own time.

5. Trading strategies

We are going to use Precison Alpha's dataset to develop trading strategies for NASDAQ stocks. Those trading strategies will then be backtested over the full sample period.

a. Methodology

To backtest the strategies, we follow the following methodology for each stock i:

$$\mathrm{backtest_i(t) = signal_i(t) \times return_i(t \rightarrow t+1)}$$

Where:

$$\mathrm{signal_i(t) = Z(X_{probaup,i}(t))}$$

$$\mathrm{returns(t \rightarrow t+1) = \log\left(\frac{price(t+1)}{price(t)}\right)}$$

As we get the signals at the close of market at t, we suppose that we can trade based on this information at the close at date t, until the close at date t+1. We will later on adjust this by supposing we wait to trade at the market open of t+1.

For now, we do not consider trading costs, or risk-sizing of positions. Those will be considered in the next "robustness" section. Also note that we only consider price returns, thus excluding dividend payments.

Z-scoring method

In the previous section, we have used a robust z-scoring method for all metrics, as they were highly non-normal. Here, we will take a different approach, as we have observed two things in the exploratory data analysis: first, ***proba_up*** is a fairly normally distributed random variable, and second, its average value is empirically and should be by design 50%. As such, we will normalize all ***proba_up*** stock data $\mathrm{X_i(t)}$ using the following formula:

$$\mathrm{Z(X_i(t)) = \frac{X_i(t) - 50\%}{std(X)}}$$

Where $\mathrm{std(X)}$ is the panel standard deviation of X, equal to 0.50%.

By normalizing that way, we ensure that a ***proba_up*** above 50% will always lead to a positive signal (and thus a long position in the stock), and conversely ***proba_up*** below 50% will always lead to a negative signal.

Example: z-scoring Apple's (AAPL) proba_up signal

For illustration, we plot below the ***proba_up*** data, and its corresponding z-scored signal.

b. Time-Series results

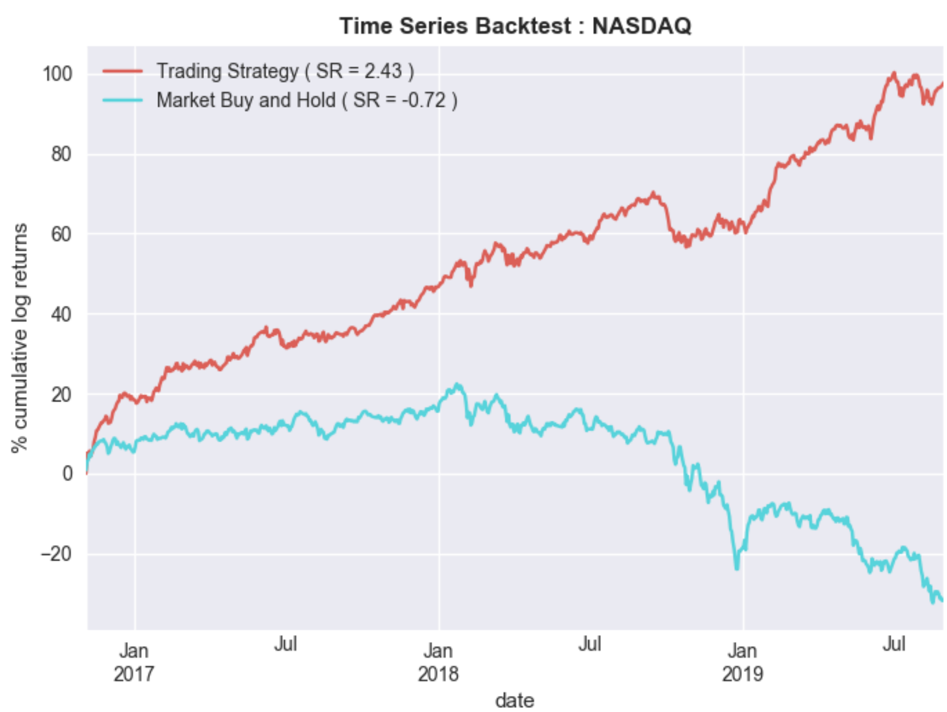
We can plot the cumulative log return of the entire trading strategy, which is equally weighting the backtest for each stock:

$$\text{Trading Strategy}(t) = \frac{1}{N}\sum_{i=1}^{N} \text{backtest}_i(t)$$

And compare it to the market, which is equally weighting all stock returns:

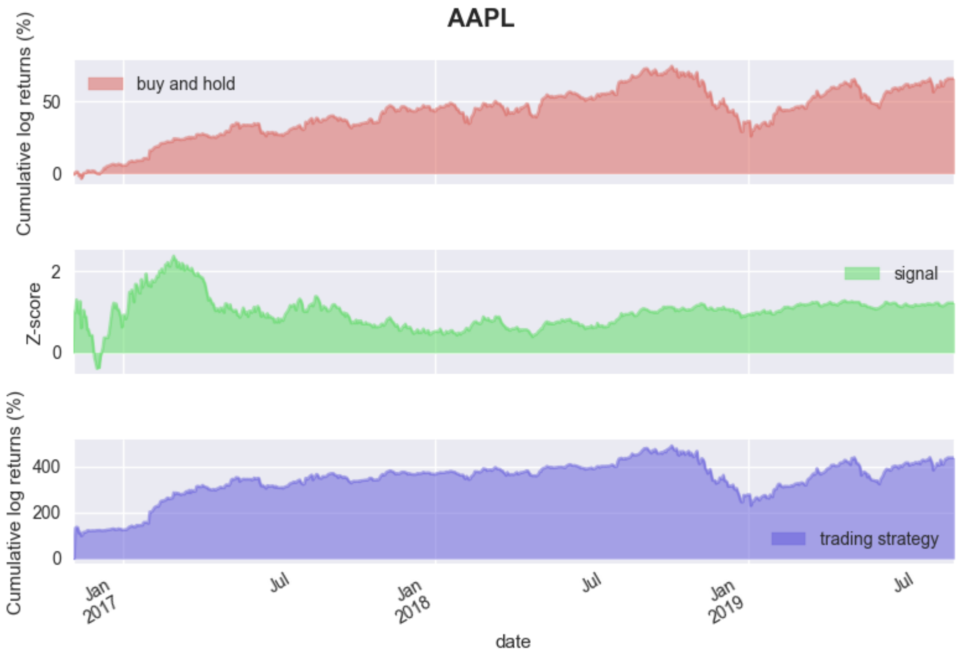
$$\text{Market Buy and Hold}(t) = \frac{1}{N}\sum_{i=1}^{N} \text{return}_i(t)$$

We see that **the trading strategy has a Sharpe Ratio of 2.43, which is significantly superior to the market buy and hold at -0.72**.



Example: Trading Apple's (AAPL) stock using the system

The example below shows the buy and hold return for Apple's stock, and compare it with the return from trading the stock proportionally to the signal extracted from Precision Alpha's system. Trading using Precision Alpha generate a return of 400% for the considered period, outperforming the 60% return from simply buying and holding the stock.



c. Cross-Sectional results

While the previous strategy considered trading stocks in the time series, i.e. going long or short based on whether their probability of going up is below or above 50%, we can also trade stock in the cross-section, i.e. at every date t, ranking them from most bullish (highest ***proba_up***) to most bearish (lowest ***proba_up***).

We thus redefine the signals for each stock as:

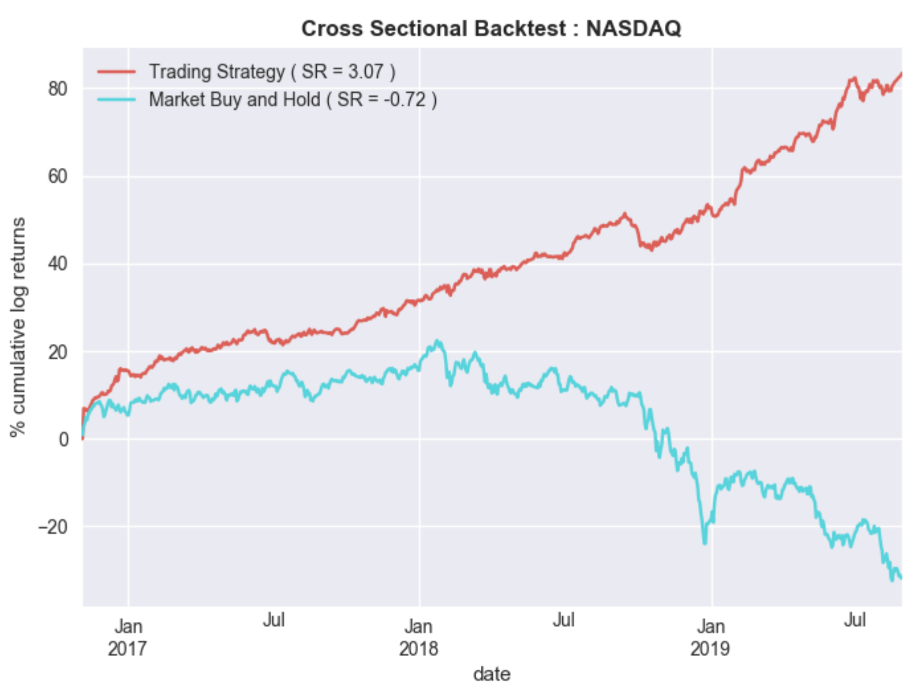
$$\mathrm{signal_i(t) = Z_{cross-sectional}(X_{probaup,i}(t))}$$

Where:

$$\mathrm{Z(X_i(t)) = \frac{X_i(t) - mean(X(t))}{std(X(t))}}$$

And $\mathrm{mean}\big(\mathrm{X(t)}\big), \mathrm{std(X(t))}$ are the cross-sectional mean and standard deviation of $\mathrm{X_i(t)}$, at every date t. We should note that using this method, we might end up shorting a stock that has ***proba_up***>50%, and vice versa for longs, as whether we go long or short will depend on where $\mathrm{mean}\big(\mathrm{X(t)}\big)$ is.

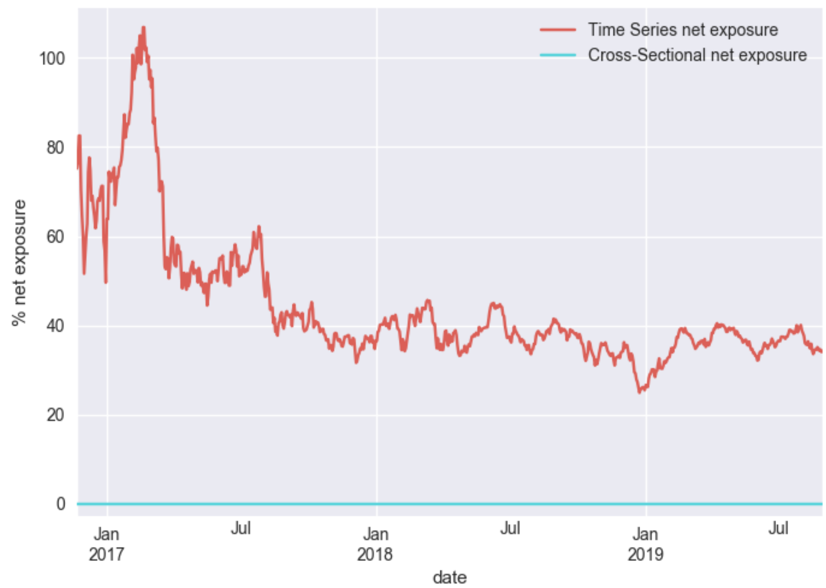
We follow the similar backesting methodology as before. Results are plotted below. Once again, the trading strategy outperforms significantly the market buy and hold.



d. Contrasting Time Series versus Cross-Sectional results

The Cross-Sectional results are encouraging as they show that the strategy is probably generating true stock selection alpha, rather than just timing the equity market. Indeed, time-series strategies, as they are built bottom-up, might end up having a large market bias over time, whereas, as cross-sectional strategies have a long for every short, they should have minimal market exposure.

We can check this by plotting the net exposure of both styles of strategies, and see that the time series net exposure to the market was always positive, and peaked at 107% net long in February 2017 for example, while the cross-sectional stayed market-neutral from a dollar exposure perspective.



6. Robustness tests

In this section, we will test the robustness of the results from the previous sections, by computing results for various assumptions and computation methods.

a. Backtest analytics

We compute several analytics to analyze the backtest. They enable to assess the robustness of the trading strategy beyond the traditional Sharpe Ratio. Most analytics show the trading strategy is pretty robust.

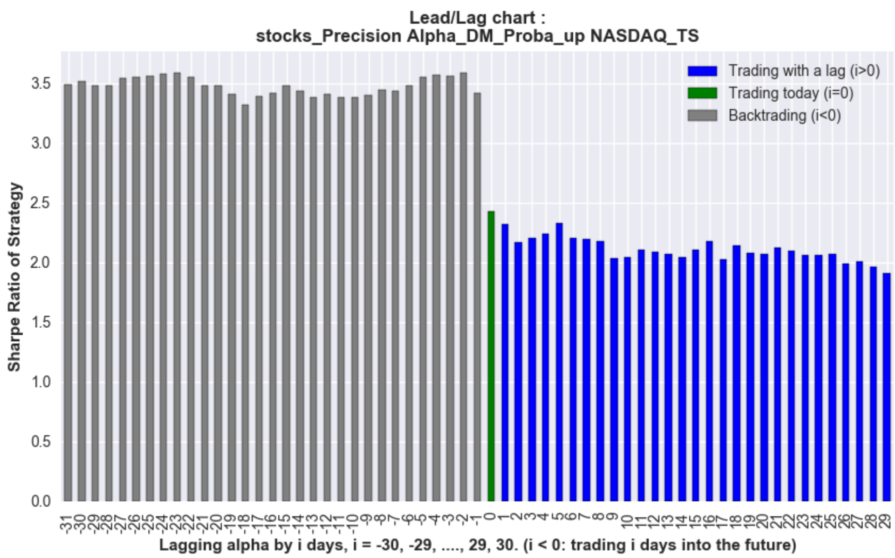
We not that the turnover for the cross-sectional strategy is only slightly lower than the time-series based. Also, the mean and median trading lengths are very dissimilar, suggesting a right-skewed distribution of trade lengths. In other words, 50% of trades will have a length of less than 2/4 days for TS/CS respectively, which could be difficult to monetize, while there will be some trades that will last weeks if not months.

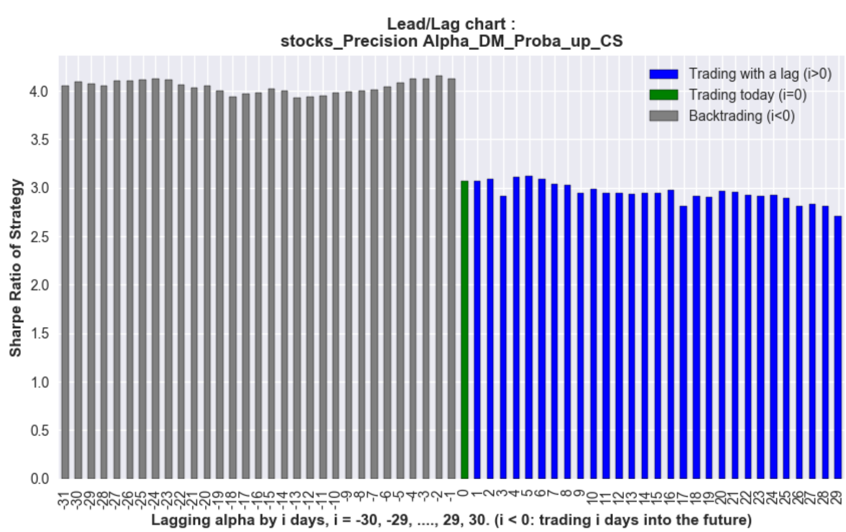
	TS	CS	Buy and Hold	*description*	ideal value
Annualized return	34.6%	29.5%	-12.7%	*Mean daily return x 260*	*high*
Annualized volatility	14.2%	9.6%	15.5%	*Daily standard deviation x sqrt(260)*	*high*
Max drawdown	13.8%	8.5%	54.8%	*Maximum drawdown*	*low*
SR	2.4	3.1	-0.8	*Sharpe Ratio*	*high*
PnL_curve	98%	97%	-74%	*Correlation between the cumulative return PnL and time*	*high*
SR_abs_diff	0.9	0.2	2.2	*Absolute difference between the SR in the first half versus second half of the sample*	*low*
median trading length (days)	2.0	4.0	732.0	*Median length of a trade (trade = being long or being short a stock)*	*depends*
mean trading length (days)	65.7	80.6	723.4	*Mean length of a trade (trade = being long or being short a stock)*	*depends*
monthly turnover	98.2%	57.1%	0.0%	*Average sum of daily changes in absolute weights per month*	*depends*

b. Lead/Lag impact

We want to assess the effect on the Sharpe Ratio of the strategy, from lagging the signals by 1,2,...30 days. I.e. if we have a buy signal on Monday close of market, rather than assuming that we enter the trade at Monday close until Tuesday close (lag = 1 days), we will enter the trade on Tuesday close (lag = 2 days).

Running those results, we get similar results for the Time Series (TS) and Cross-Sectional (CS) strategies: **i.e. no significant drop of alpha when lagging the signal by 1,2,... 30 days.** This is in line with the finding in the predictive modeling section. This is however odd, as we would expect the profitability of the strategy to drop to close to 0 after lagging the signal by several weeks. Inversely, we clearly see that the SR jumps significantly when backtrading, i.e. using future information to trade, which makes sense as we then foresee the market momentum in advance.





c. Trading at open(t+1) rather than close (t)

When we defined the backtesting framework in Section 5. a., we assumed we could trade at close of market at date t, using the signals as of close of date t. However, in practice, it is more likely that users will have to wait for markets to open the next day to execute their trades. As such, we are going to re-compute backtests now assuming:

$$\mathrm{backtest_i(t) = signal_i(t) \times return_i(t+1 \rightarrow t+2)}$$

Where:

$$\mathrm{signal_i(t) = Z(X_{probaup,i}(t))}$$

$$\mathrm{returns(t+1 \rightarrow t+2) = \log\left(\frac{price\_open(t+2)}{price\_open(t+1)}\right)}$$

Remember that as we get the signals after the close of market at (t), we will trade based on this information at the open at date t+1, until the open at date t+2.

Comparing the results, we can see that trading at the next open does not detracts much from the performance of the strategy.

d. Trading change in probabilities rather than level

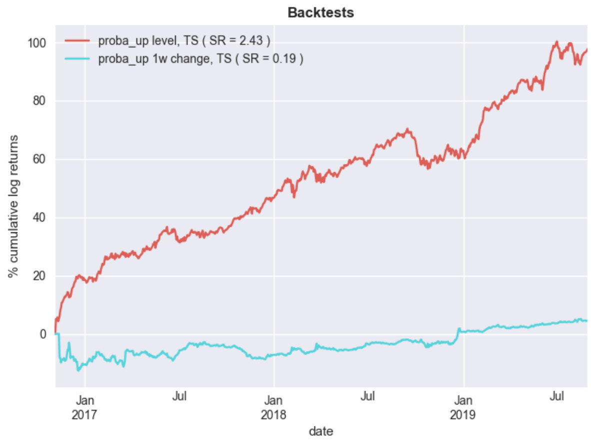
As we have seen in the machine learning section, it seems that looking at changes in probabilities leads to better predictive power than looking at its level. As such, rather than trading using the following signal:

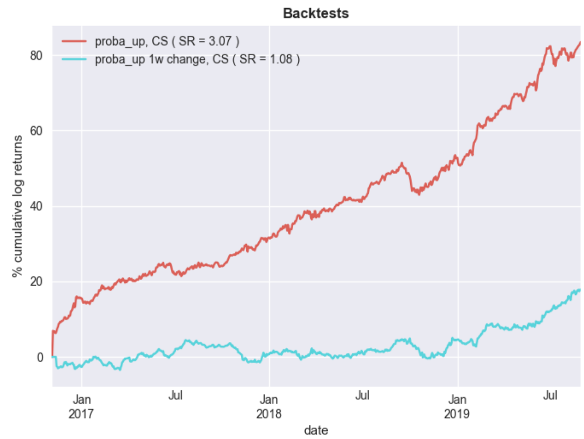
$$\mathrm{signal_i(t) = Z(X_{probaup,i}(t))}$$

We also backtest the 5-day change in probability:

$$\mathrm{signal_i(t) = Z(\Delta_{5day} Proba_{up}(t))}$$

A shown below, and contrary with our finding in the machine learning section, it seems that trading changes in probabilities is nowhere near profitable as trading the level.

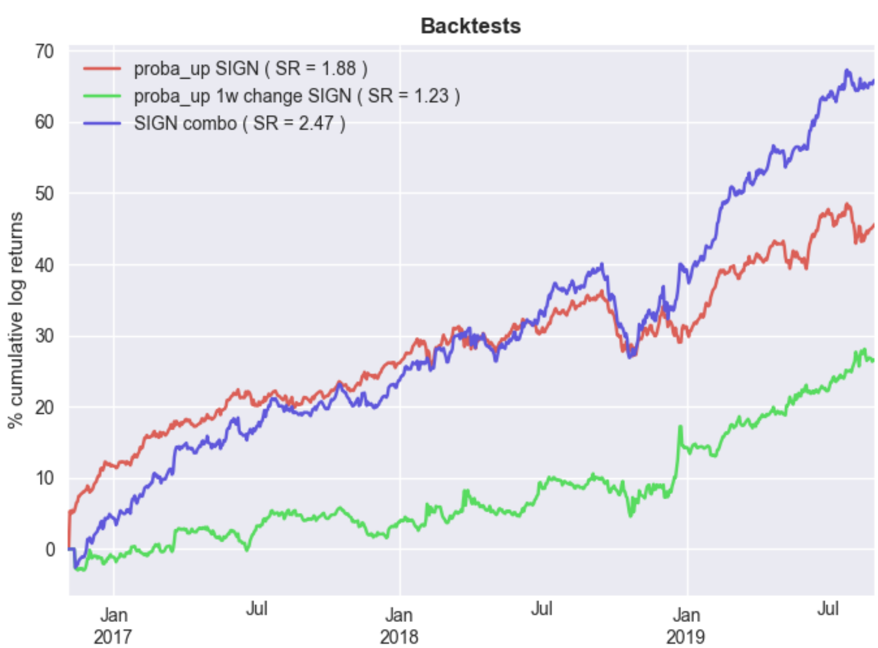




However, it is important to note that during the machine learning section, the goal was to predict the sign of the next day return, whereas here the strategy will be most profitable if we can predicit well its magnitude too. A good way to illustrate this, is to simplify both level and 5-day change strategies, so that we trade based on the sign of the varaible, rather than their z-score, i.e.:

$$\begin{cases} Z(\mathrm{Proba_{up}}) = \mathrm{SIGN}(\mathrm{Proba_{up}}(t) - 0.5) \in [[-1;+1]] \\ Z(\Delta_{1w}\mathrm{Proba_{up}}) = \mathrm{SIGN}\left(\Delta_{1w}\mathrm{Proba_{up}}(t)\right) \in [[-1;+1]] \\ Z(\mathrm{Combo}) = Z(\mathrm{Proba_{up}}) + Z(\Delta_{1w}\mathrm{Proba_{up}}) \in [[-2;-1;0;1;2]] \end{cases}$$

Both these strategies are profitable, and in this case we can see that trading the change in probabilities has a comparable Sharpe Ratio of 1.2. We find that $Z(\mathrm{Proba_{up}})$ and $Z(\Delta_{1w}\mathrm{Proba_{up}})$ have a correaltion of -20%, and thus combining those two strategies into a “combo” leads to a higher SR of 2.5.



e. Trading cost impact

So far in our analysis, we have not considered the impact of trading costs on the trading strategy profitability. While transaction cost modeling can be a very complex exercise, we will here take a simple approach, whereby we model transaction costs as a constant bid-offer cost as basis point (bps = 0.01%) of the trade notional.

For example, if the % exposure to a stock i at dates t, t+1, t+2 are:

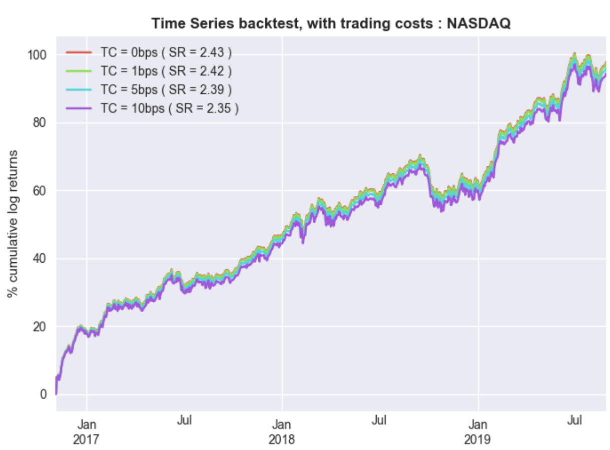
$$\begin{cases} w_i(t) = 0\% \\ w_i(t+1) = 25\% \\ w_i(t+2) = 0\% \end{cases}$$

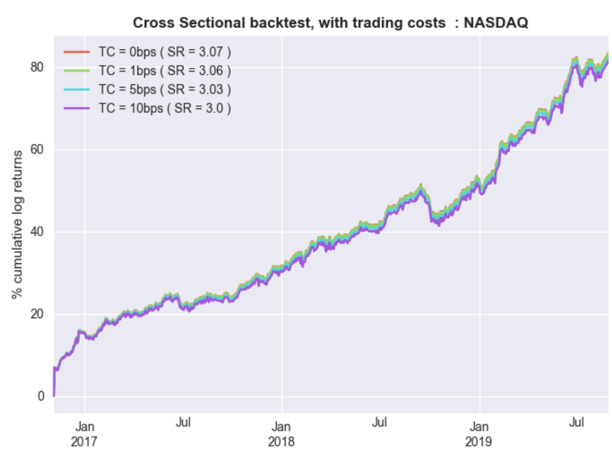
As a position was established and then closed, we crossed the bid-offer once. If the bid-offer was, say 10bps, the transaction cost at portfolio level of that trade is 25%*10bps = 2.5bps. Seen another way, we could break down the two-way trading cost of bid-offer as two, one-way trading costs of (bid-mid) and (mid-offer) of 5bps each, and calculate the transaction cost as |25% -0%|*5bps + |0%-25%|*5bps = 2.5bps. Mathematically, we can model the transaction cost at any date t as:

$$\text{transactioncost}_i(t) = |w_i(t) - w_{i-1}(t)| \times \underbrace{\text{one\_way\_trading\_cost}}_{=(\text{bid}-\text{offer})/2}$$

Rather than imposing assumption of trading costs, we will show the impact of an array of bid-offers, of 1bps, 5bps and 10bps. Given we are trading the 150 most liquid stocks on a large exchange, we think that an all-inclusive cost in the single digit basis points make sense.

As we can see in the chart below, the impact of trading costs on the strategies' profitability is minimal: the Sharpe Ratio only decreases by a few decimals.





f. Signal conviction and expected returns

Finally, we can see how our initial formula to size trades proportionally to their signal(t) makes sense, by computing the conditional Sharpe Ratio based on the magnitude of the signal (t). **We can see that the larger the signal in absolute value, the larger the expected risk-adjusted return.** This result validates our backtesting approach whereby we size position proportionally to the signal.

7. Benchmarking

In this final section, we compare the performance of the strategy versus various market benchmarks and indices.

a. Alpha versus equities

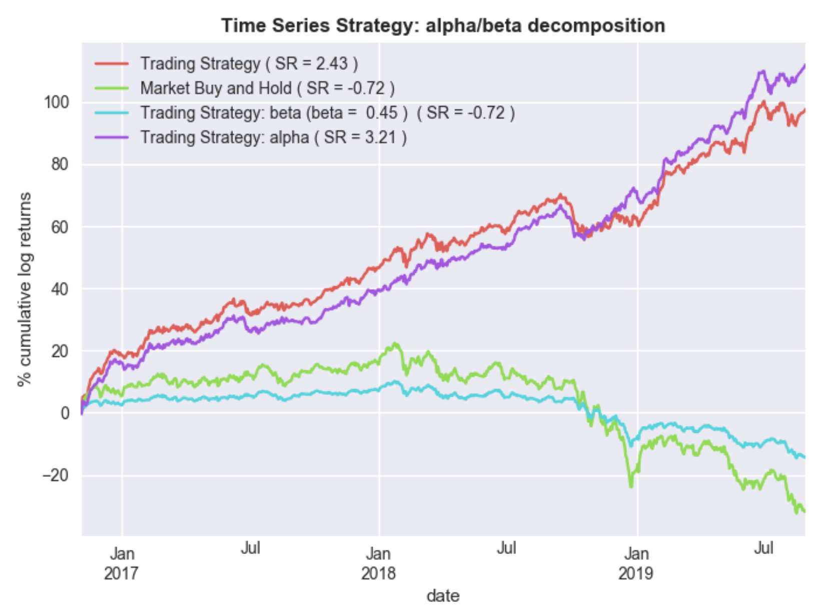
So far in this paper, we have assessed the absolute return of the strategy. But as the strategies have been built bottom-up, there could be some beta bias over time (i.e. the strategy being constantly long or short), that might generate some of the returns. Here we strive to extract the pure alpha from the strategy.

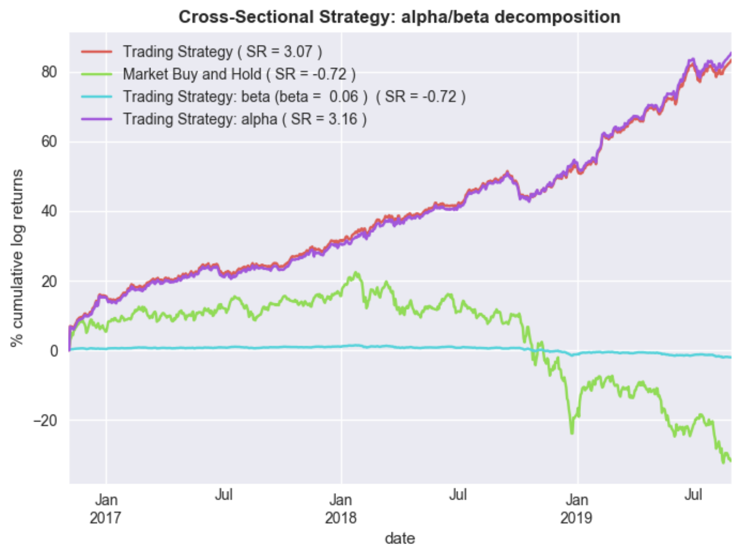
We run the following linear regression:

$$\text{Trading Strategy}(t) = \underbrace{\beta \times \text{Market}(t)}_{\text{beta returns}} + \underbrace{\alpha(t)}_{\text{alpha returns}}$$

Where β is obtained from a linear regression of the trading strategy returns versus the market (i.e. the simple equally weighted buy and hold of all stocks). The residual of the regression is $\alpha(t)$.

Running those analytics on the Time Series and Cross-Sectional strategies for the NASDAQ, we get:





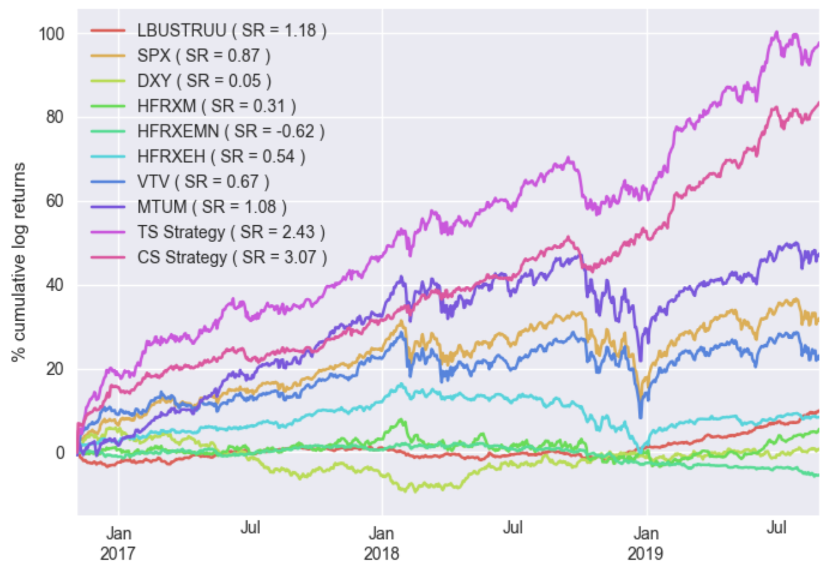
It shows that **both TS and CS strategies have the overwhelming majority of their returns from alpha**. Of note, the TS strategy has a long beta bias, whereas the CS strategy is pretty much market neutral, with a near zero beta. Obviously, given the short duration of our sample, those results might vary on longer horizons.

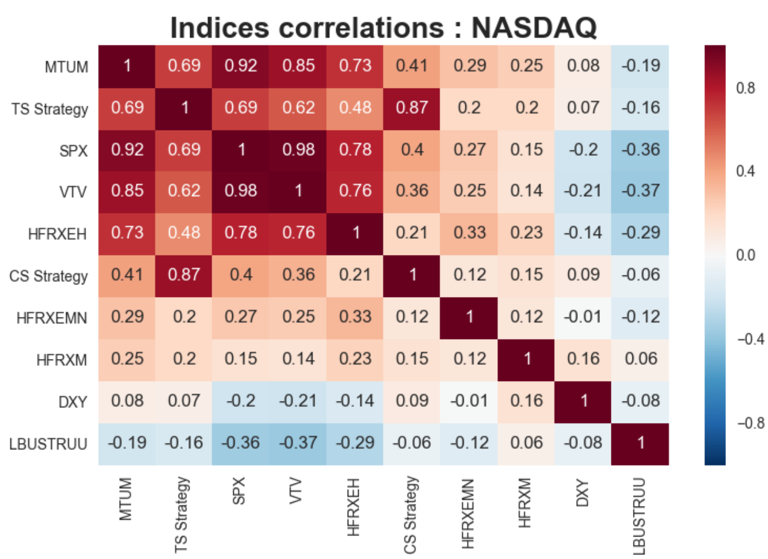
b. Performance versus various indices

We then compare the performance of the trading strategies versus various market indices, listed below.

Bloomberg ticker	Index name
LBUSTRUU Index	U.S. Aggregate Bond Index
SPX Index	S&P 500 INDEX
DXY Curncy	DOLLAR INDEX SPOT
HFRXM Index	Hedge Fund Research HFRX Macro HF
HFRXEMN Index	Hedge Fund Research HFRX Equity Market Neutral HF
HFRXEH Index	Hedge Fund Research HFRX Equity HF
VTV US Equity	VANGUARD VALUE ETF
MTUM US Equity	ISHARES EDGE MSCI USA MOMENTUM

On a risk-adjusted basis, the trading strategies outperform all indices. We can see that the time series strategy has quite a large correlation with equity indices (resulting from its positive market beta), while the Cross-Sectional, being market neutral, has lower correlations.





8. Conclusions

In this paper, we assessed the application of Precision Alpha's dataset to trading daily the 150 most liquid stocks on the NASDAQ, between November 2016 and August 2019.

In the Exploratory Data Analysis section, we analyzed the distribution and links between variables. Some variables are functionally related, such as proba_up and emotion, power. Most variables are non-normally distributed.

In the predictive modeling section, we found that the dominant probability ("proba_up"), had predictive power over future returns. Then in the Machine Learning section, we confirmed that proba_up had indeed the best accuracy (53%) as it relates to predicting the sign of future returns, closely followed by emotion. Other metrics such as power, resistance and noise had less significant results. As we tested other derivative features, we found that 1-week change in probabilities also had significant predictive power.

In the trading strategy section, we backtested directional (time-series) and market-neutral (cross-sectional) strategies using proba_up as the signal. Trading Strategies are profitable with Sharpe Ratios of 2.4 and 3.1 respectively, significantly outperforming the market. We found that the trading strategies were robust to various assumptions about trading costs, execution prices, portfolio construction.

Finally, we showed that the trading strategies generated substantial alpha over their respective markets, as well as outperformed major asset class and hedge fund indices for the considered period.

9. Appendix: Disclaimer

This paper has been prepared by Prince Analytics LLC, a data science and quantitative research consultancy based in Stamford, CT.

This paper is intended for information purpose only, and does not constitute investment advice. Past returns are no guide to future returns.

This paper and its analytics were generated from a dataset provided by Precision Alpha to Prince Analytics in August 2019. Prince Analytics does not take any responsibilities from computational errors that might have arisen throughout this report, or from erroneous or misleading data given as input.