Examining Long-Term Trends in Company Fundamentals Data

Michael Dickens

MOTIVATION

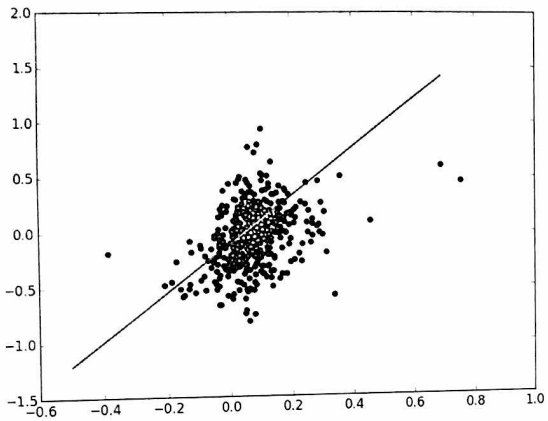
The equities market is generally considered to be efficient, but there are a few indicators that are known to have some predictive power over future price changes. This suggests that the market has some room for identifying inefficiencies. Much work is done on applying machine learning to short-term trading, but there exists little research on using machine learning to identify long-term inefficiencies; almost all mutual funds and hedge funds rely on the judgment of humans to make long-term bets about the market. Therefore, we have reason to believe \textit{a priori} that there exist long-term market inefficiencies which can be found with machine learning.

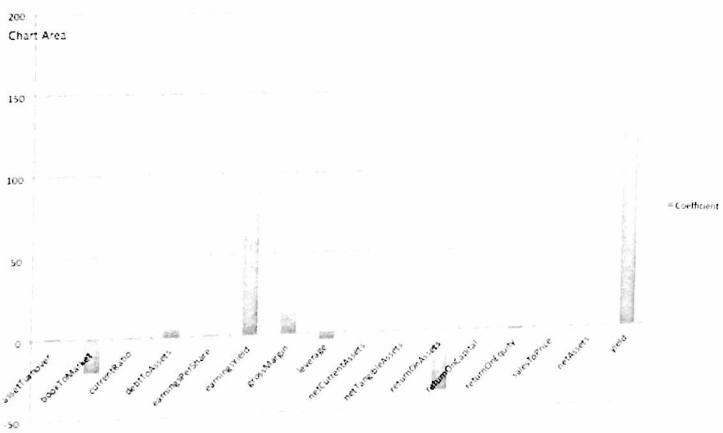
FEATURES

To identify useful derived features, I took a set of basic features and computed all possible ratios of these features. Ideally I would like to compute ratios of sums of features as well as other more complex combinations of basic features, but this becomes prohibitively expensive as the number of possible derived features expands rapidly.

To test the predictive power of derived features, I used a scoring function that divides the stock market into deciles according to a given feature, then returns the absolute value difference in average risk-adjusted returns between the first and last deciles. I chose this scoring function because it most accurately illuminates exploitable market inefficiencies.

I was not able to find any features that had stronger predictive power than well-known indicators. Earnings yield performed by far the best, followed by other metrics that are very similar to earnings yield. A few unpopular features did have reasonably strong predictive power, but none did as well as well-known features such as earnings yield and return on capital.



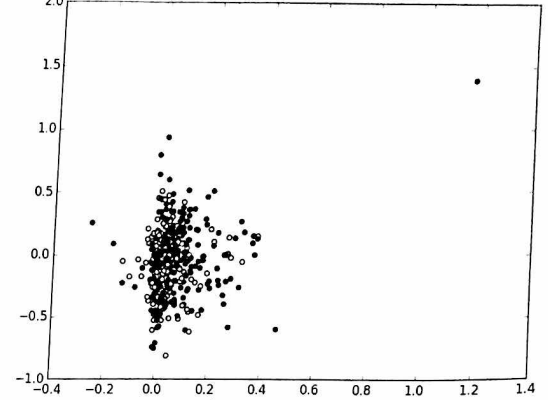


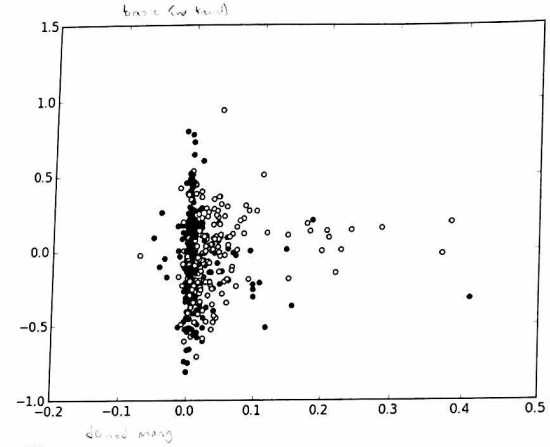
LINEAR REGRESSION

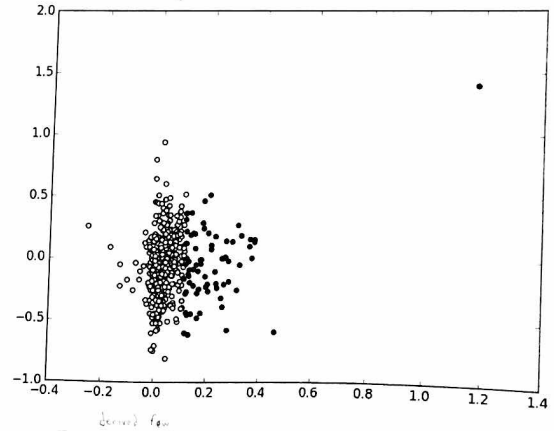
Linear regression weakly outperforms both earnings yield and dividend yield, but does not have better risk-adjusted return. I attempted some modifications to the regression but was not able to produce substantially better results.

Unlike with SVM, I found that adding more indicators as features to the regression did not reduce accuracy. The regression algorithm assigned weight 0 or near-0 to most indicators.

Name	Adjust.	Return	Stdev
Earnings yield	0.65	16.3	25.0
Dividend yield	0.59	17.9	30.5
Linear regression	0.56	18.1	32.2
SVM	0.91	27.4	30.1







SVM

Predicting returns more naturally fits with regression than with classification. However, we want to solve a classification problem in some sense: we want to classify stocks as "buy" or "don't buy." This model does in fact produce some useful results. We consider a stock to be a positive example if it gets an annual return above some threshold (I found the best results when I used 20%), and a negative example otherwise. Then we run the SVM algorithm to separate the sets of positive and negative examples.

As with linear regression, I used well-known indicators as features. I also tried using basic fundamentals from companies' income statements and balance sheets, and then applying kernel methods to produce derived indicators. I found that SVM had a major problem with overfitting. Using many features, or using kernel methods to combine features, created results that performed well on the training set but did not generalize to the test set. But using a small number of well-known features known to have predictive power (earnings yield, return on capital, dividend yield) produced a classification that has better predictive power than earnings yield alone.