Dear Thomas, PJ,

Following our last meeting I would like to present to you some further experiment results accompanied by my own understandings on the following two topics: feature selection and forecasting from 1d to 30d (d as in days).

1. Feature Selection

Background

There are generally two classes of feature selection techniques: Filter methods and Wrapper methods.

In the feature evolution process, if we extract as much information as possible from a given data set while using the smallest number of features, we can not only save much computational cost, but also build a simplified model with better generalization. Generally, filter methods such as principal component analysis (PCA) have been more performant and scales well with high dimensionality. However, in this study, the PCA is not appropriate because our goal is not only to reduce the data dimensionality, but also to obtain highly accurate predictive models. A common disadvantage of filter methods is that they ignore the interaction with the classifier (the search in the feature subset space is separated from the search in the hypothesis space, with *hypothesis* being the unseen and unknown relationship between the input space and the output), and that most proposed techniques are univariate. This means that each feature is considered separately, thereby ignoring feature dependencies and the combined impacts of features, which may lead to worse classification performance when compared to other types of feature selection techniques.

On the contrary, the Genetic Algorithm which belongs to the *randomised* class of Wrapper methods (the other class being *deterministic*) has proven to have superior performance to other algorithms for data set with high dimensionality. The word *Wrapper* refers to the fact that these methods embed the model hypothesis search within the feature subset search. In this setup, a search procedure in the space of possible feature subsets is defined, and various subsets of features are generated and evaluated. The evaluation of a specific subset of features is obtained by ***training and testing a specific classification/regression model, rendering this approach tailored to a specific model algorithm***. This statement is important because it means that whatever optimal features we've come up with, they are only optimal when working with the model 'wrapped' in the search process. They should not be considered optimal in the most general sense.

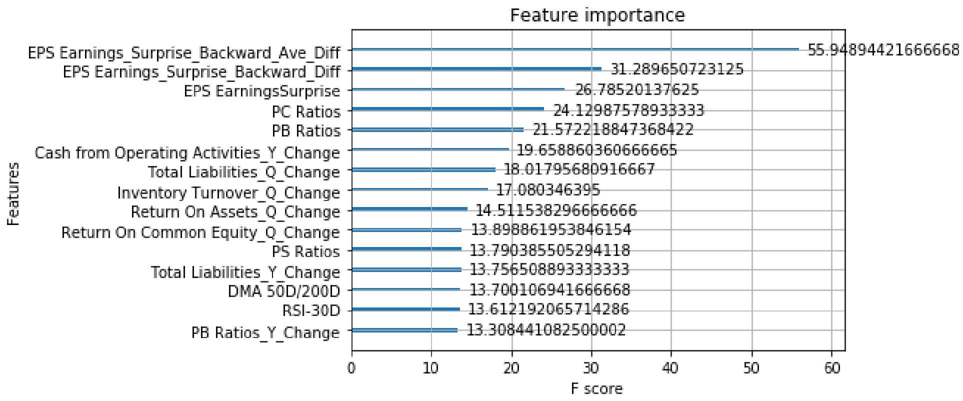
XGBoost Feature Plot

Independent of an external feature selection process, when the XGBoost model is trained it internally provides a number of statistics with regard to the *importance* of individual features. The most relevant statistic which is plotted below implies the relative contribution of the corresponding feature to the model calculated by taking each feature's contribution for each tree in the model.

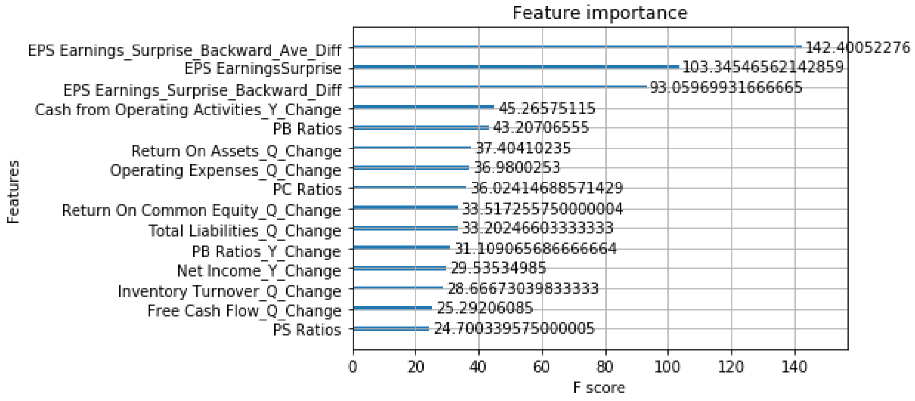
I am providing one plot of the 15 most significant features for each of two example runs predicting results for stocks from Q3 2018. The same is provided for two example runs for stocks from Q4 2018.

Predicting stocks from Q3 2048

Run 1

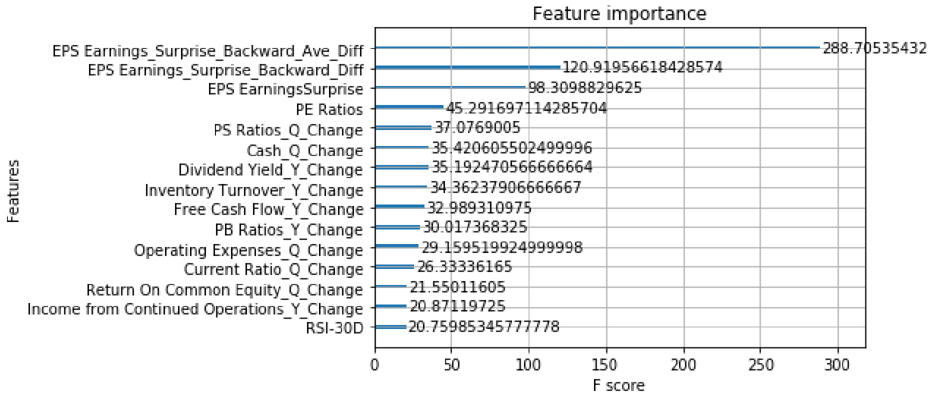


Run 2

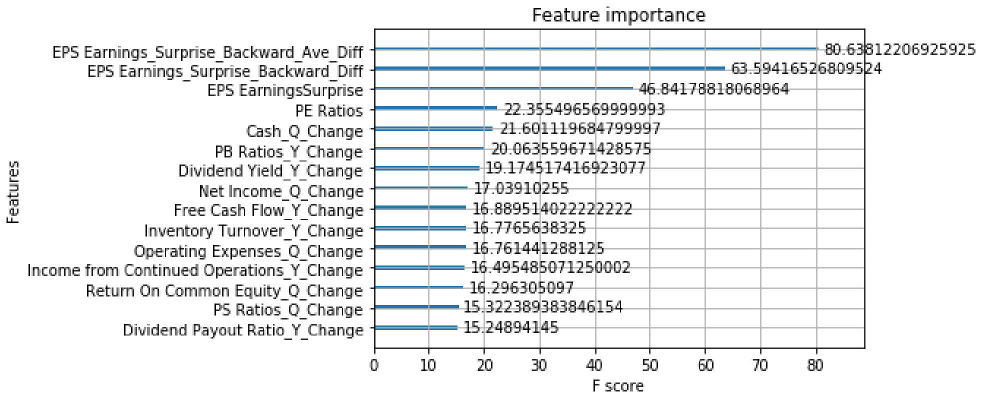


Predicting stocks from Q4 2048

Run 1



Run 2



We can draw two conclusions here:

1. The three features with regard to EPS (Earnings Per Share) have the more significant impacts on the model output. This is very much in line with reality because EPS numbers are closely monitored for listed companies by stock analysts and investors.

2. The list of *significant* features is not fixed and changes all the time.

3. The most significant feature can when the training set and test set changes (in this case changing from testing for Q3 2018 stocks to testing for Q4 2018 stocks).

Feature Selection

A Genetic algorithm feature selection program was one of the first programs I built in my system. I didn't mention this piece of work nor its results in my current paper due to the following two reasons:

1. The algorithm-chosen 'optimal' feature set is not fixed and changes slightly every time I run the program;
2. Reducing the original input feature space down to the 'optimal' set does not improve the eventual performance of the model;

I didn't understand fully why I wasn't able to benefit from feature selection and hence didn't include its results.

I have used the last few weeks to experiment more with this program. One of the simple tests I ran is this:

- I start the test by randomly selecting to use only one of the inputs. This is followed by model tuning and then classification forecast. This test is repeated a few times and the result is expectedly always very bad;
- I increase the inputs to two. Repeat the same experiment a few times and results are all bad;
- Then, I start using more input features randomly chosen. Results are improving but not stable.

  Up to now I have not performed feature selection.

- Lastly, when I use most or all inputs and use the feature selection program to pick the 'best' set out of them, and then perform model tuning and then forecasting based on the these algorithm-selected features, I more or less achieve the same results as when I use the full set of inputs but without performing feature selection.

My explanation to "*The algorithm-chosen 'optimal' feature set is not fixed and changes slightly every time I run the program*":

When I include enough input features, although no individual of them carries a distinctly stronger signal except for the three EPS features, the combination of some or all of them correlate with the output in one way or another, but not in a very strong sense of way. XGBoost is a very powerful model and it's able to identify different combinations of features that drive the output. This is why I am seeing different 'optimal' feature sets that produce more or less the same 'optimal' result.

My explanation to "*Reducing the original input feature space down to the 'optimal' set does not improve the eventual performance of the model*":

Every time we use a different (but sufficiently large) feature set, the XGBoost model is re-tuned again with respect to the new input feature set. I believe this GA-based model tuning is so effective and XGBoost is so adaptable, that whether I use the full set of input features or a sufficiently large sub set of it that carries similar signals, the model can tune its hyperparameters differently to work best with the current input feature space, and hence I am obtaining very similar results in these cases.

2. Forecasting 1d-to-30d Risk Adjusted Returns

What we know so far

First of all I would like to concede that, not being able to action on the forecasted 0d-to-30d stock returns/movement direction is a big problem in my previous experiments. Thanks for making me realise that. My work so far have been presented with a view to showcase certain observations with the forecast results, especially the interesting observation of how stocks could be grouped together with diminishing portfolio returns once stocks have been ranked by their predicted results. Such observation however can not be actioned upon as stock prices move away upon data release. It also makes an underlying assumption that prediction could be carried out as soon as data has come out.

It is important to recognise however that my input feature space is tied to the 0d-to-30d (or 0-day-to-n-day) stock returns following earnings release and that is the definition of Post Earnings Announcement Drift. All the economic paper which study the phenomena of PEAD and its drivers define PEAD as the 0-day-to-n-day returns.

That being the case, as encouraged by you (much appreciated), I would like not to be constrained by such definition and would like to relate my results more to the actionable 1d-to-30d stock returns.

1d-to-30d stock returns

Regarding the 1d-to-30d stock returns and directions, below is a summary of conclusions I've drawn so far through various experiments:

1. It is not possible to directly forecast the 1d-to-30d using the current input features and model.

   Verdict: This is expected.
   Reason: The current input feature space comprise primarily of earnings data. These are ***event*** data and should relate to stock movements from right before the event happens to some future time after the event. The correlation between the event data and stock returns that start some time after the event is weak.

2. I have not been able to derive the 1d-to-30d stock returns by subtracting the real 0d-to-1d stock returns from the predicted 0d-to-30d stock returns, with or without aligning the statistics of distributions of these different returns.

   Reason: Daily stock movement is considered random and as a *Martingale*. The *Efficient Market Hypothesis (EMH)* states that today's stock price can not be used to predict tomorrow's (the PEAD phenomena is considered an anomaly). Generally speaking, it is not possible to easily '*derive*' the stock returns from 1 day to 30 day using known stock returns from 0 day to 1 day. Also equally importantly, the predicted 0d-to-30d returns are not so accurate.

3. It is also not possible to infer the 1d-to-30d stock movement direction by selecting those stocks whose predicted 0d-to-30d stock direction is forecasted to be ***Up*** but the real 0d-to-1d stock movement is ***Down***, or vice versa.

New argument

I would like to expand on the third experiment mentioned above. This experiment is a failure when I calculate the result using all the stocks in the test set.

However, I've noticed that ***when I look at only those stocks whose 0d-to-30d direction has been correctly predicted, I am always able to correctly infer the 1d-to-30d direction when the predicted 0d-to-30d movement direction and the real 0d-to-1d movement direction are opposite to each other***.

This may sound like something obvious, but it tells me that, ***it is possible to infer the 1d-to-30d stock movement direction using this "affirmed-direction" method, provided that the 0d-to-30d forecast is accurate enough***.

To put things in perspective, following is a series of experiments I've conducted:

- As given by the model, the classification accuracy of the 0d-to-30d stock movement direction for the stocks in Q3 2018 (about 900 of them) is around 59%.
  Of all these stocks, the accuracy of inferring the 1d-to-30d stock movement is merely around 50% using this method.

- Out of all these stocks, if I randomly sample more stocks whose directions have been correctly predicted and less stocks whose directions have been incorrectly predicted, so as to synthetically produce a new group of stocks which has an overall 0d-to-30d 'prediction accuracy' of **70%**, the accuracy of inferring the 1d-to-30d movement direction using the 'affirmed-direction' approach has gone up to **64%** in this group;

- Out of all these stocks, if I randomly sample more stocks whose directions have been correctly predicted and less stocks whose directions have been incorrectly predicted, so as to produce a new group of stocks which has an overall 0d-to-30d 'prediction accuracy' of **79%**, the accuracy of inferring the 1d-to-30d movement direction using this method has gone up to **75%**;

- If I only pick those stocks whose movement direction has been correctly forecasted, using this method the accuracy of interring the 1d-to-30d movement direction is **100%**;

Therefore, we can make an argument that, **in order to more accurately infer the movement direction for 1d-to-30d stock returns, we just need to improve the accuracy of forecasting the original 0d-to-30d stock movement directions. And we only try to infer the 1d-to-30d direction when the forecasted 0d-to-30d direction is opposite to the real 0d-to-1d direction. This way we can effectively shift the new target back to our original target.**

In the paper I should acknowledge that my data being analysed can be incomplete (lacking attribute values or certain attributes of interest, or containing only aggregate data), noisy (containing errors, or outlier values which deviate from the expected), and inconsistent. This on its own is a big area of work but nonetheless is an area with improvement opportunities. Any improvement in data quality, data preprocessing methods, model optimization techniques among other things can lead to improved accuracy of forecasting PEAD (0-day-to-n-day returns), which in turn could lead to improved accuracy of inferring the direction of 1-day-to-n-day stock movement.

3. Paper's New Focus

Following our last meeting discussion I would like to first revert the paper title back to *Capturing dynamics of post earnings announcement drift using genetic algorithm-optimized supervised learnings*.

Also, the paper's main contribution should no longer be about using model to identify stocks that can be grouped in a way that generate diminishing portfolio returns, nor about building long-short market neutral portfolios using the results.

The amended paper's contributions to the literature therefore will be:

1. For the first time in the literature (to the best of my knowledge), showcase how to use machine learning technologies (proprietary feature engineering, data preprocessing, feature selection, and model optimization) to build a system to process and analyse quarterly earnings data and predict stock's movement drift as driven by such data;

2. For the first time in the literature (to the best of my knowledge), use machine learning technologies to depict the Post Earning Announcement Drift phenomenon, by forecasting the 0-day-to-n-day stock movements as driven by earnings announcements under a variety of test scenarios with the classification accuracy going up to 63%. All the existing paper that study PEAD use factor based regression methods that are typically seen in finance papers.

3. In contrast to a wide group of existing literature which study and forecast the unactionable PEAD, i.e. 0-day-to-n-day stock movements, I present a simple method which can theoretically use the forecasted PEAD movement direction to infer the movement direction of 1-day-to-n-day stock returns. I call this simple method the 'affirmed direction' method.

   My theory is that, although PEAD forecast can not be actioned upon directly, increasing the prediction accuracy of PEAD will increase the accuracy of inferring the 1-day-to-n-day stock movement when using the 'affirmed direction' method. It should be quite clear that by any measure a highly accurate prediction of PEAD is extremely hard to come by but this theory makes PEAD forecast indirectly actionable.

   Prior to this, I will explain why earnings data can not be directly used to forecast stock movement that starts from S(t) with $t > 0$ and describe the range of failed attempts I've made to achieve that.

4. I will also briefly showcase the result that when stocks are ranked by their predicted 0d-to-30d returns, a set of portfolios that consist of a moving subset of ranked stocks are showing diminishing portfolio returns. Although the predicted returns are not actionable, such result presents as an interesting observation of what one may get when one attempts to forecast PEAD using supervised learning models, and can be good complements to what are being showcased in items 1 and 2. None of such results or observations appear to have been documented in the literature.