BofA GLOBAL RESEARCH

Signal over noise

Machine Learning primer for investors

Primer | 23 April 2019 | Equity and Quant Strategy | United States

Key takeaways

- This Machine Learning primer presents new tools for alpha generation outside of traditional statistical approaches.
- In investing, data proliferation is the new fuel and Machine Learning is increasingly becoming the tool of choice.
- We discuss key concepts of over-fitting, Clustering, PCA, Lasso, Ridge, Elastic Net, Random Forests, XGBoost, Deep Learning.

Everything you wanted to know about Machine Learning

BofA GLOBAL RESEARCH

Machine Learning is all around us, from smartphones, to smart cars, to drone technology, to social media, and naturally, in finance. New datasets are materializing in the form of unstructured (e.g. text) data, geolocation data, big data, etc., making the old tools obsolete. For investors, this presents an opportunity to leverage new tools outside of traditional statistics paradigms (e.g. linear regression) to model the increasingly more complicated factor relationships combined with these new datasets.

What does Machine Learning have to do with investing?

In investing, data proliferation is the new fuel and Machine Learning is increasingly becoming the tool of choice for security selection, forecasting, portfolio construction and trading execution algorithms. In addition, investors have accumulated an extensive list of economic, market and fundamental company data where many of these datasets are often correlated with one another. All these factors and datasets have to be distilled down in order to be used for price return prediction. Machine Learning has an emphasis on factor selection and consolidating the list of factors into fewer but more meaningful factors. Traditional prediction methods in finance will typically fail under these circumstances, whereas Machine Learning can be a highly adaptable method to achieve better results with more computational power and data.

What Machine Learning approaches are covered?

We discuss the perils of over-fitting in both Machine Learning and back-testing, and ways of overcoming these challenges. Supervised Learning is where we apply a model given a set of factors (for example, a stock's P/E ratio, 3 month price change and earnings revisions) to predict a response (such as financial returns or a binary response if a particular stock will outperform). The primary Supervised Learning models covered in this report are linear (Lasso, Ridge, Elastic Net) and tree based (Random Forests, Bagging and XGBoost). We also discuss unsupervised learning models (Principal Component Analysis and Clustering), where there is no explicit prediction from the model but the model can be effective when dealing with a large number of correlated factors.

Deep Learning/Natural Language Processing (NLP)

Deep Learning (DL) is also discussed. DL is a set of interconnected algorithms that are loosely based on the structure of a brain. The advances incorporating Deep Learning have had recent A.I. breakthroughs such as Google's AlphaGo program beating the best Go player in the world. This was achieved by combining Deep Learning with Reinforcement Learning(*). Deep Learning has experienced significant improvement in accuracy rates in fields such as image recognition and natural language processing but has yet to make a significant impact in financial markets. We also illustrated a Deep Learning model called long-short term memory (LSTM) in a recent note, applying it to Glassdoor reviews to show how it can help identify alpha opportunities.

What is Machine Learning?

Machine Learning has many different definitions, but at its core it uses a dataset to build a mathematical model in order to make predictions. The central idea is that the algorithm can learn to make a decision or prediction without explicit instruction. The complexity within Machine Learning varies greatly, ranging from a slightly altered linear regression to a deep neural network with millions of parameters.

Why apply Machine Learning to finance?

BofA GLOBAL RESEARCH

Further, most traditional methods in finance assume a linear relationship between financial returns and a list of factors. What about nonlinear relationships and interaction amongst factors? As per our strategy work (http://rsch.baml.com/r?q=CLAnKmZVlGi8fwiUeJegHg), we have detected an interaction between the VIX index and price momentum (i.e. if VIX is greater than 25, price momentum suffers and if VIX is below 25 then price momentums outperforms). Machine Learning can do this by identifying these types of interactions. In addition, it can incorporate the appropriate nonlinearities as most datasets in finance are not linear. This is where Machine Learning has the advantage over traditional statistical approaches in finance and can potentially lead to alpha if applied correctly.

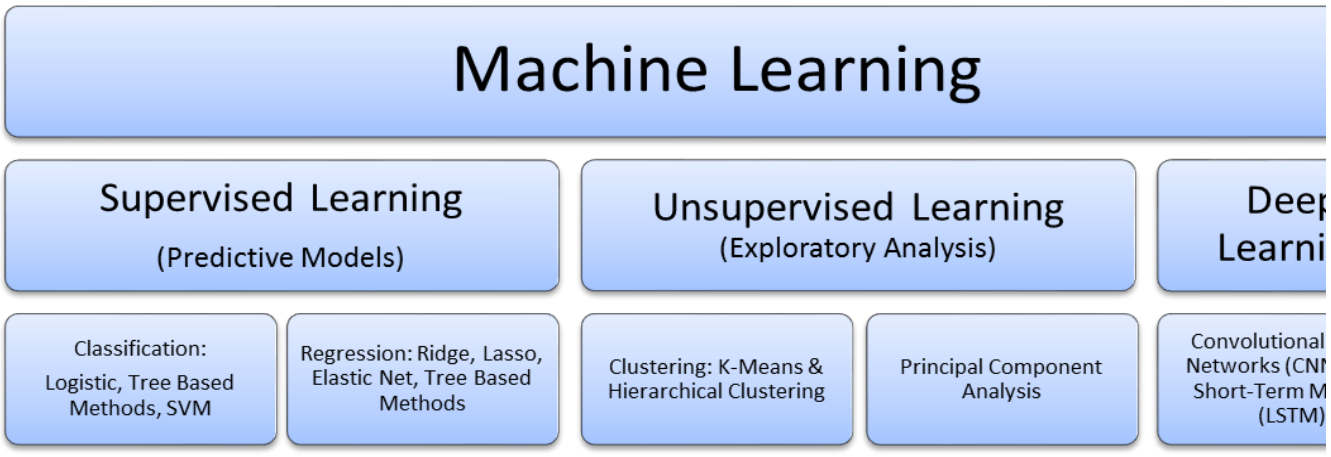
Machine Learning Overview

In this brief introduction we cover several Machine Learning techniques, which can be generally categorized into three types: 1) Supervised Learning, 2) Unsupervised Learning, and 3) Deep Learning. Supervised Learning is a technique that uses a model to predict an outcome for a given dataset. An outcome can be either discrete or continuous. The techniques that forecast discrete outcomes are called classification models, whereas techniques that forecast continuous outcome are called regression. Discrete outcomes can be binary responses (i.e. spam or not spam) but can also be categorical (i.e. movie review ratings from 1 to 10), whereas continuous outcomes can be financial returns or weather temperatures.

Unsupervised Learning is a tool for understanding the structure of the dataset. Compared with Supervised Learning, Unsupervised Learning has no outcome variable the model is trying to forecast. Rather, the main objective is to understand the relationships between datasets or different ways factors can be consolidated together.

Deep Learning techniques are loosely inspired by the human brain where it maps sets of inputs to neurons (i.e. units or nodes) with connections to obtain a final prediction. For Deep Learning, the majority of applications can be categorized as Supervised Learning, with other techniques being used for Unsupervised Learning.

Exhibit 1: Machine Learning Overview



Source: BofA Merrill Lynch US Equity & US Quant Strategy

Exhibit 2: Summary of Machine Learning models covered in this introduction

Supervised Learning	Random Forests	Computes each decision tree independently by randomly subselecting factors to include in the model but more robust than single model.
		Pros: can calculate factor importance as diagnostic tool, can usually work "out of the box" with default input parameters, no need to transform data, rarely over-fits
		Cons: Difficult to interpret, slow to compute predictions at real time, can be biased with categorical data
	XGBoost	Builds decision trees one at a time where each tree corrects for errors made by previous tree
		Pros: Usually performs better than Random Forests, achieves high accuracy rates, fast computation
		Cons: Harder set input parameters, prone to over-fitting (but can be avoided with proper specification), black box
	Ridge Regression	Linear regression model that prevents overfitting
		Pros: does not suffer from correlated factors (i.e. multicollinearity), can prevent over-fitting
		Cons: no factor selection, assumes a linear model
	Lasso Regression	Linear regression model that picks the top select factors that matter out of many
		Pros: factor selection, can prevent over-fitting
		Cons: suffers from correlated factors (i.e. multicollinearity)
	Elastic Net Regression	Linear model that combines Lasso and Ridge
		Pros: Powerful for large datasets where the number of factors might be in the thousands, able to select groups of correlated factors
		Cons: Computational cost, greater flexibility increases the probability of over-fitting
	Principal Component Regression	Uses Principal Component Analysis (PCA) to reduce a large number of factors into a smaller set to ultimately use inside of a regression
		Pros: Reduces multicollinearity, reduces a large number of factors into a smaller set of factors
		Cons: Not a factor selection method, meaning of the individual factors may be lost
Unsupervised Learning	K-Means Clustering	Identifies data points with similar characteristics and groups them into a predetermined number of clients
		Pros: Easy to use, fast to run, results are easy to interpret
		Cons: Need to specify number of clusters in advance, can get varied results when changing the 'distance' algorithm
	Hierarchical Clustering	Identifies data points with similar characteristics and creates hierarchical clusters in a tree structure
		Pros: Easy to understand, do not need to specify the number of clusters
		Cons: Results can vary depending on inputs chosen
	Principal Component Analysis (PCA)	Useful exploratory tool that aggregates and boils down a large number of (potentially correlated) factors into a smaller set of factors
		Pros: Helpful to make sense of large datasets by distilling into new factor indicators
		Cons:Non-linearities hard to model, meaning of the individual factors may be lost
Deep Learning	Convolutional Neural Networks (CNN)	Deep neural network models that are popular for image classification
		Pros: Fast to compute, very accurate for things like image classfication, good at handling three dimensional data
		Cons: Requires a lot of data to work properly, black box interpretations
	Long Short-Term Memory (LSTM)	Known for sequential classification problems such as: text labelling, speech recognition, time series.
		Pros: Can classify using long sequences (up to 30-50), can be highly accurate, addresses the vanishing gradient problem that Recurrent Neural Networks are known for
		Cons: Very slow at computing the model, black box interpretations

Source: BofA Merrill Lynch US Equity & US Quant Strategy

Over-fitting

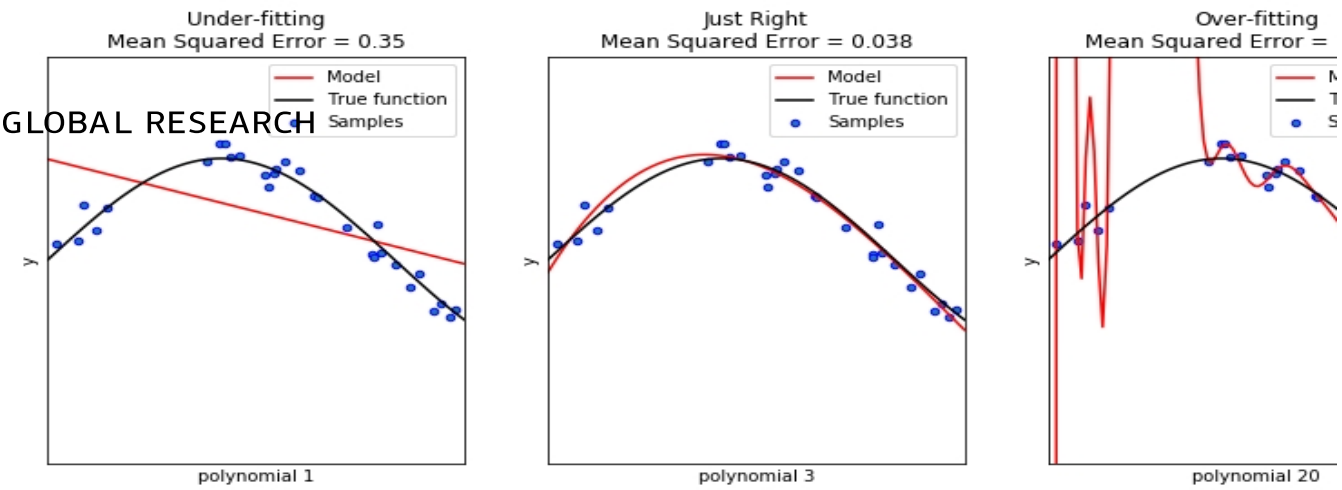
In Machine Learning (and back-testing), over-fitting is the single biggest risk of producing inaccurate forecasts. Put another way, an over-fitted model is one that applies too many parameters than is justified by the data in order to fit all the residual noise. Therefore, if a Machine Learning model is fitted against all the noise, it will likely perform poorly as a forecasting methodology.

The majority of the Supervised Learning models discussed in this report need to be steered in the right way in order to minimize over-fitting and thus minimize prediction errors. As a result, we dedicate a small section here before moving forward.

Find the balance between under-fitting & over-fitting

As discussed, an over-fitted model is a statistical model that typically contains more parameters than necessary for a given dataset such that the model only works well for the data it was analyzed on but not for incoming data or prediction. As per Exhibit 3, for a given toy dataset, high variance over-fits the model with over twenty parameters whereas high bias will under-fit a model with only one parameter (i.e. linear regression). In this case, the balance is found by the middle chart in Exhibit 3 with only fitting 3 parameters.

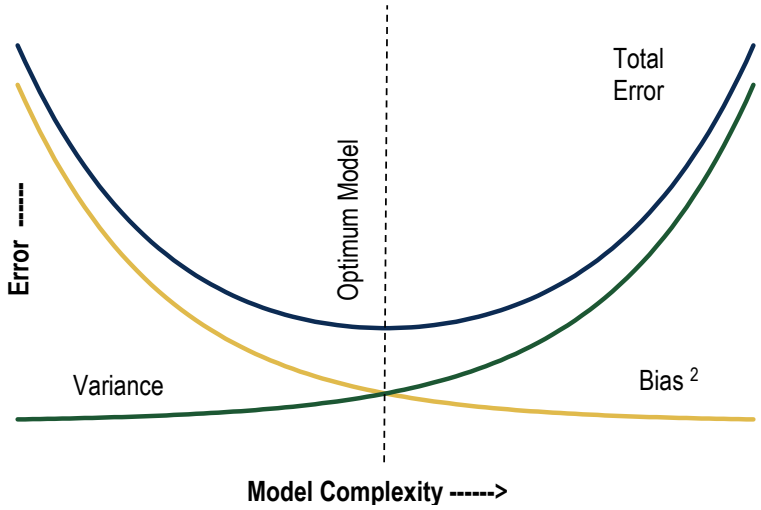
Exhibit 3: Underfitting, Overfitting, Variance-Bias trade off



Source: BofA Merrill Lynch US Equity & US Quant Strategy

We can define bias as the forecast error (i.e. the difference between the forecast and actual), whereas variance measures the variability of a forecast each time it is produced. If we increase the model complexity (i.e. the number of parameters), this leads to higher variance, whereas if we decrease the model complexity, it will lead to higher bias (i.e. linear regression) as per Chart 1.

Chart 1: More complexity leads to higher variance while less complexity leads to high bias



Source: BofA Merrill Lynch US Equity & US Quant Strategy

The desired goal is to have something that is in the middle in terms of complexity that will operate well on incoming data and for prediction. Most of the Machine Learning models covered in the following sections will have what is called a 'tuning' parameter that can be adjusted in order to find a good middle ground in order to balance variance and bias to help prevent over/under fitting a model. Please see Cross-Validation: A key way to minimize over-fitting section for how data sampling techniques can be combined with the tuning parameter to prevent a Machine Learning model from over-fitting. In the next section, we will explain what Unsupervised Learning is before we launch into the Supervised Learning section.

Unsupervised Learning

In this section we will cover Unsupervised Learning techniques known as Principal Component Analysis (PCA), K-Means Clustering and Hierarchical Clustering.

Principal Component Analysis (PCA)

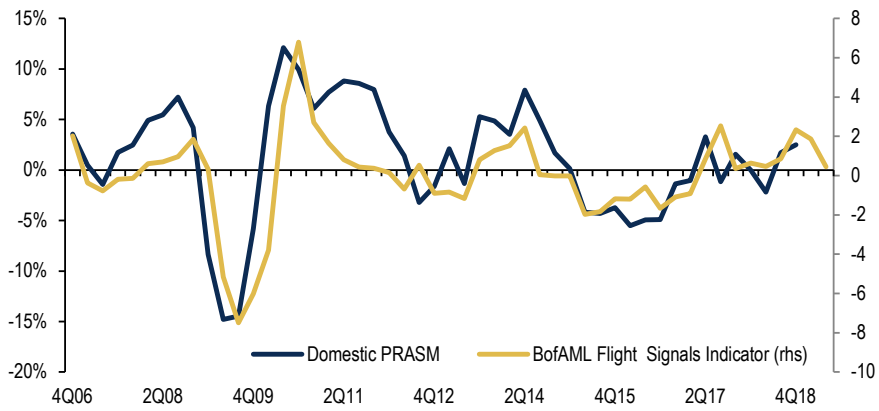
Description

When we are faced with a large number of correlated factors, principal component analysis (PCA) allows us to distill down into a small number of factors that explain most of the variability in the dataset. Principal component analysis is a mathematical technique that converts a set of variables into linearly uncorrelated variables called principal components.

Application: BofAML Flight Signals

Our Airlines team has recently launched the BofAML Flight Signals (http://rsch.baml.com/r?q=DvyaDleJND8twnDP94VAlw), which was developed using a PCA model, and is intended to be an indicator of how US domestic airline unit revenues (PRASM) could grow or contract over a 6 month horizon, providing investors with another tool to evaluate industry trends over a longer period. The list of factor categories included are ARC bookings data, crude oil prices, industry supply growth taken from published schedules, CEO Confidence data and aggregated BAC credit and debit card data on airline sector spend. A subset of the principal components that explains the majority of the dataset is taken. The result of this can be added together to create a series that best correlates with the target variable.

Chart 2: BofAML Flight Signals Indicator vs US airline domestic industry PRASM



BofAML Flight Signals back-tested performance reflects application of the indicator prior to its inception date as if the model had been in existence at that time. This does not reflect actual performance. It is not intended to be indicative of actual or future performance. The actual performance of our *Flight Signals* model may vary significantly from the back-tested performance. The back-tested performance results are based on criteria applied retroactively with the benefit of hindsight and knowledge of factors that may have positively affected its performance, and cannot account for all financial risks that may affect the performance of our model going forward.
BofAML Flight Signals is intended to be an indicative metric only and may not be used for reference purposes or as a measure of performance for any financial instrument or contract, or otherwise relied upon by third parties for any other purpose, without the prior written consent of BofA Merrill Lynch Global Research. This indicator was not created to act as a benchmark.
Source: BofA Merrill Lynch Global Research

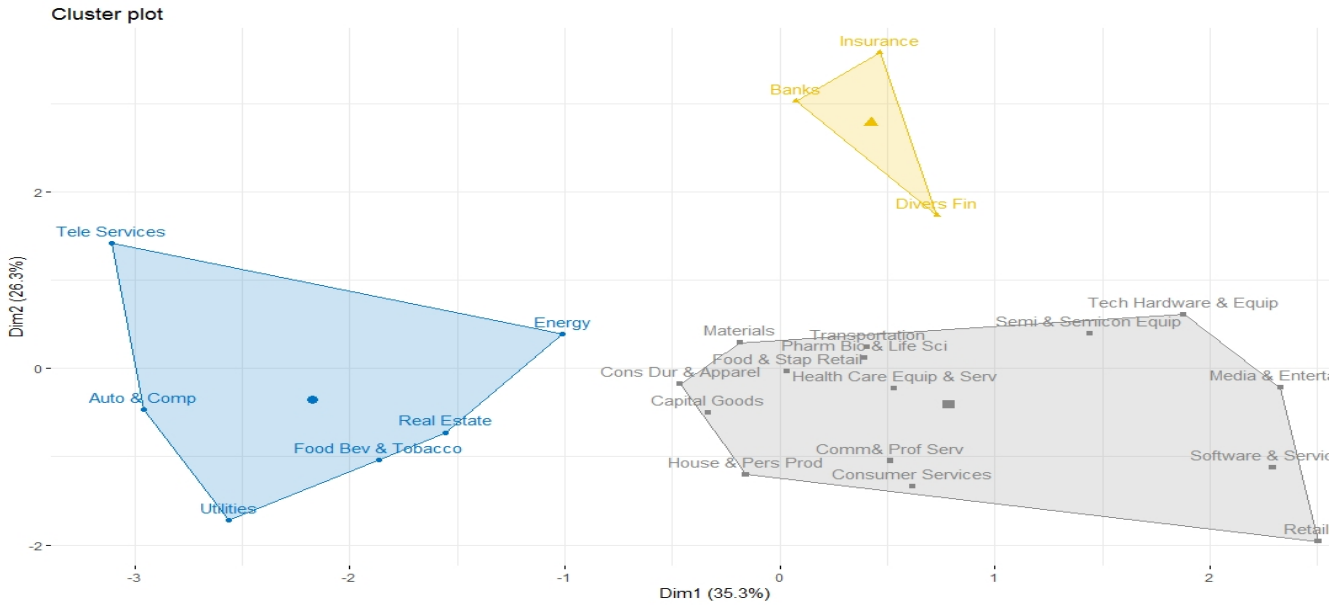
K Means Clustering

Put simply, clustering is an approach for finding subgroups using a set of factors. In general, the approach tries to partition the data into k clusters (defined by user) in which each data point belongs to the closest mean. A user must define how many clusters they want the algorithm to look for. Clustering approaches are useful for exploratory analysis or for creating more structure to then be fed into a supervised learning problem. Please see K-Means Clustering Algorithm section in the Appendix for implementation details.

Application: Grouping industry groups on financial ratios relative to market

Exhibit 4 illustrates a toy example using S&P 500 industry groups where for each group we have relative (versus S&P 500) ROE, dividend yield, net margin, forward P/E, dividend payout ratio, FCF yield and Net Debt/EBITDA for Mar 2019. Based on all these factors, Insurance, Banks and Diversified Financials are paired off with one another. The others are split into two other groups (loosely based on growth versus defensive/value). Telecom Services, Energy and Utilities are grouped together while Bio Tech, Semis, Software and Media & Entertainment. The unsupervised model is suggesting that some relationship exists between these industry groups using the financial ratio information that we provided.

Exhibit 4: K-Means clustering with 3 clusters



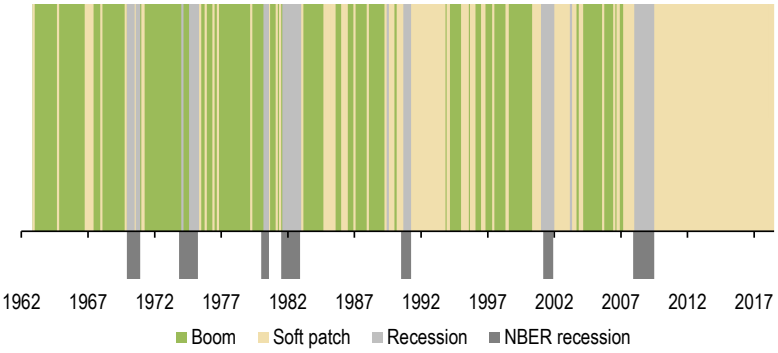
Source: BofA Merrill Lynch US Equity & Quant Strategy

Represents clusters using principal components (to be explained further down) with Dim1 representing PC 1 and Dim 2 representing PC 2

Application: Evaluating the phases of the business cycle via K-Means

Our US Economics team applied the k-means clustering algorithm to the FRED-MD database (monthly database of over 100 economic variables) to evaluate the phases of the business cycle (http://rsch.baml.com/r?q=KBCwgwJc27DHXjenw6qIVw). The team had three broad objectives: 1) to use data clustering to create an expansion / contraction indicator, 2) to see if the clustering algorithm finds differences between the current expansion and past expansions, and 3) to assess the performance of the algorithm in "real time," both as a leading recession indicator and as a signal for stock- and bond-market investors. Although it is the second-longest in postwar history (July will mark the longest expansion in history), it has basically been a very long soft patch, with standard peak business-cycle conditions in-check so far.

Chart 3: The phases of the business cycle according to big data



Source: BofA Merrill Lynch Global Research, FRB St. Louis

Hierarchical Clustering

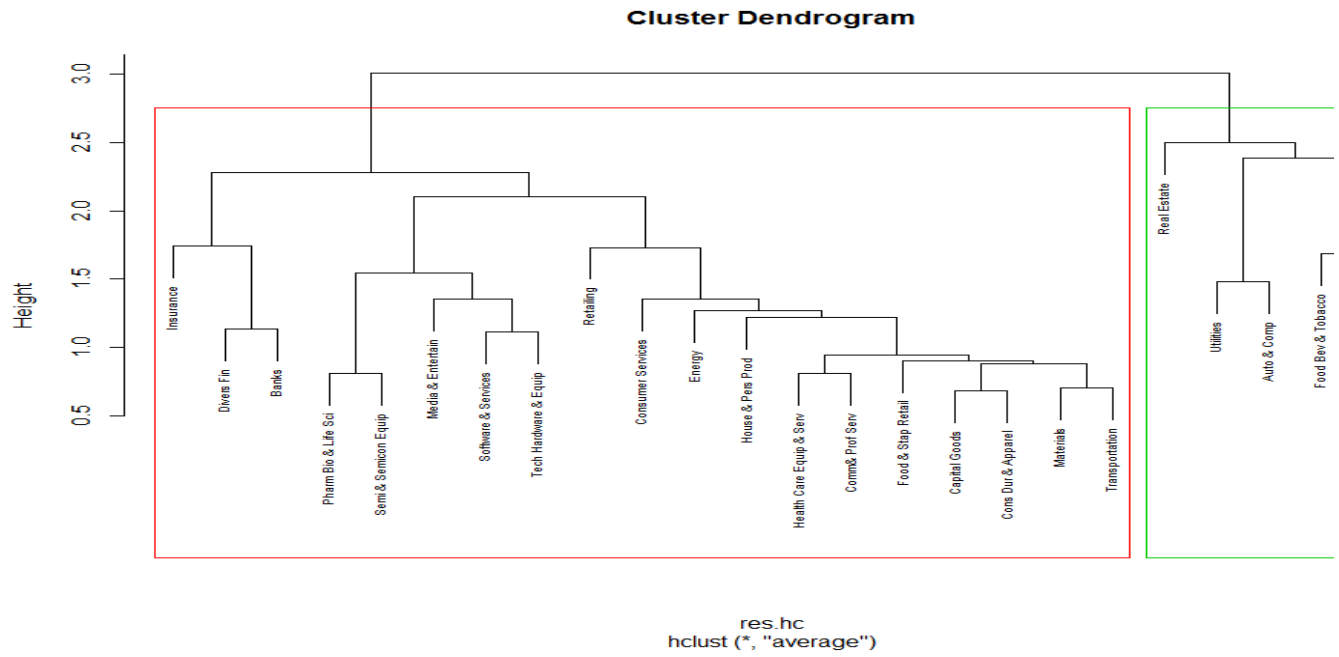
Description

BofA GLOBAL RESEARCH

Hierarchal clustering differs from K-means clustering in that it relies on hierarchy of clusters. In addition, one advantage is that hierarchical clustering does not need to have the number of clusters chosen by the user as it will identify this automatically (see appendix for technical details). We reused the K-Means toy example (as described above) as per Exhibit 5. Hierarchical clustering identified two main clusters without a user needing to decide this. Defensive/yield industry groups cluster together as per Real Estate, Utilities, Food Bev & Tobacco and Tele Services. Meanwhile, high growth stands out via the middle branch of the bigger tree (pharma and semis).

This approach provides more nuanced information with an attractive tree based representation (called Dendrogram). If two industry groups are fused on a horizontal line, then they are quite similar to one another based on their multiples defined in the K-Means example.

Exhibit 5: Hierarchical clustering example using industry groups



Source: BofA Merrill Lynch US Equity & Quant Strategy

Application: Using Hierarchical Clustering to identify best US ETFs to trade

Work from our Quantitative Investment Strategies have created a framework known as Dynamically Diversified Momentum (http://rsch.baml.com/r?q=Kju-MAmSXL-pee2vx8p1uA) which utilizes a combination of hierarchal clustering to identify diversification and momentum within US ETFs in order to construct a high risk-adjusted return long only portfolio. They use the correlation matrix of US ETFs based on returns to hierarchically cluster the strategies into groups of strategies which are correlated within the group and importantly uncorrelated across groups. They then go long only the top performing assets in each cluster, which improves returns due to momentum effects. The final condition is to use a holding period of several months (3-12) to allow strategies to rebound from occasional drawdowns. By doing this, minimizes correlation across clusters providing diversification benefits for improved risk-adjusted returns.

Supervised Learning: Linear Models

In this section we will cover Ridge, Lasso, Elastic Net and Principal Component Regression approaches.

Ordinary Least Squares (OLS) shortcomings

A typical linear regression model describes a relationship between an outcome variable Y and a set of X factors that are believed to predict Y. A model can be defined as:

$$Y = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p + \varepsilon$$

where one usually fits the model using Ordinary Least Squares (OLS) to determine the β's (i.e. coefficients) in the regression. The Least Squares part of the name for OLS comes when minimizing the below with respect to the β's

$$Sum\ all\ (y_i - \beta_0 - \sum_{j=1}^{p} \beta_j X_{ij})^2$$

where p represents the number of predictors included in the equation. However, classical OLS starts to break down when it includes a large number of factors or when multiple factors are correlated with one another, resulting in a model that is over-fit or where the coefficients are erroneous.

Solutions: Ridge, Lasso and Elastic Net

Machine learning seeks to address these issues by introducing shrinkage methods that aid in preventing the regression from over-fitting through a combination of regularization and cross-validation as already discussed above. Shrinkage methods come in three different flavors. They are known as Ridge, Lasso and Elastic Net and apply an adjustment to the traditional OLS model during the fitting process which generally reduces the coefficients to prevent over-fitting. This has two advantages: 1) it acts as a variable selection mechanism, and 2) it helps address over-fitting during the fitting process when applying cross-validation. The main difference between Ridge, Lasso and Elastic Net is the different forms of the penalty term.

Ridge Regression

Description

Ridge regression has an advantage over OLS as it can deal with the bias-variance trade-off which can be used to help prevent over-fitting. If we alter the fitting procedure with the penalty term of squared β below we get:

$$Sum\ all\ (y_i - \beta_0 - \sum_{j=1}^{p} \beta_j X_{ij})^2 + \alpha \sum_{j=1}^{p} \beta_j^2$$

As α (e.g. tuning parameter) increases, this leads to increased bias and decreased variance. However, if α decreases to zero, then the equation collapses to an OLS regression. The ultimate goal for a Ridge regression over OLS is to increase the forecast accuracy.

Application: Stock Selecting

A Ridge regression model is used to predict financial returns of the Russell 1000 utilizing multiple factor variables ranging from valuation, technical, revision, etc (see appendix for reference)(*). Per each period in Exhibit 6, we apply a time series cross validation (see Exhibit 18) that is used to forecast the subsequent period. The time series cross validation procedure identified the best predicting parameter and applies a larger weight. Similarly, it will identify factors that are poor predictors and assign a lower weight. For example, the technical factors are dominating in terms of overall strength while some of the fundamentals crop up as a strong second.

Exhibit 6: Ridge regression. Shows changing coefficients over time
Red indicates a negative relationship while blue is positive and how solid the color indicates the strength of relationship.

Source: BofA Merrill Lynch US Equity & US Quant Strategy

Lasso Regression

Description

Lasso is as an automatic variable selection technique that will remove factors that are not helpful (i.e. set weights to zero) while keep factors that are meaningful (i. e. non-zero weights). Mathematically, Lasso is very similar to Ridge regression except during the fitting procedure, instead of squared β term for a penalty it is an absolute value as per below:

$$Sum\ all\ (y_i - \beta_0 - \sum_{j=1}^{p} \beta_j X_{ij})^2 + \alpha \sum_{j=1}^{p} |\beta_j|$$

In some ways Lasso might be preferable to set some variable coefficients to zero if they add no value with prediction compared with Ridge which will shrink the irrelevant factor coefficients down to a small weight. However, with financial markets those factors might suddenly become relevant depending on the particular drivers of the market at that point in time, which would make it beneficial to have some exposure to the factor.

Application: Stock Selection

For the Lasso application, we reapply the same dataset as was described in the Ridge regression application section. Exhibit 7 shows that if a factor is adding minimal information, the Lasso model will assign a zero weight (this is indicated on the chart as being blank). Though the Lasso model also acts as a variable selection technique to identify the most relevant factors it differs from Ridge by being more decisive in the selection process. For example, using the latest data from 2014 to 2018, the market returns have positive relationship with one month price returns and a positive relationship with Forward earnings yield.

Exhibit 7: Lasso coefficients over time
Red indicates a negative relationship while blue is positive and how solid the color indicates the strength of relationship.

Source: BofA Merrill Lynch US Equity & US Quant Strategy

Elastic Net Regression

Description

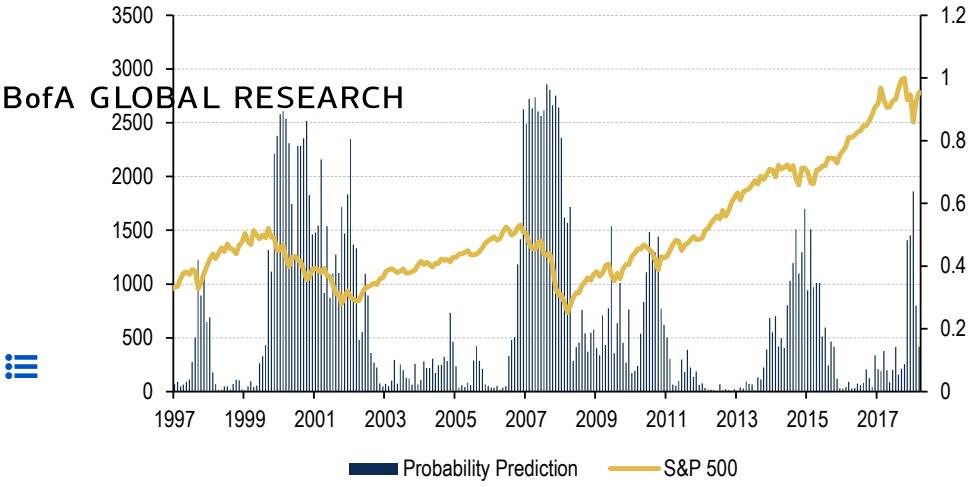
Elastic Net combines aspects of both Lasso and Ridge regression. If there are multiple correlated variables, Lasso will tend to only pick the strongest correlated factor among a correlated group. Elastic net can overcome these challenges by including a Ridge based penalty term.

$$Sum\ all\ (y_i - \beta_0 - \sum_{j=1}^{p} \beta_j X_{ij})^2 + \alpha \sum_{j=1}^{p} |\beta_j| + \alpha \sum_{j=1}^{p} |\beta_j|^2$$

Application: Predicting Peak to trough in the S&P 500

We use an Elastic Net model to predict peak to trough corrections for the S&P 500 by utilizing factors ranging from macroeconomic, fundamental, technical and sentiment data (see Table 4 for details). Chart 4 illustrates how the predicted probability changes as a result of not only the changes in the data but also the weights being learned by the Elastic Net model. Major spikes in the probability coincide with major market corrections as illustrated in the 2008 housing crisis and post dot com bubble. Similarly, minor spikes in probability seem to coincide with minor market corrections.

Chart 4: Elastic Net Peak to Trough probability prediction
A high probability indicates a peak to trough is likely to occur according to the model predictions



Source: BofA Merrill Lynch US Equity & US Quant Strategy

Principal Component Regression

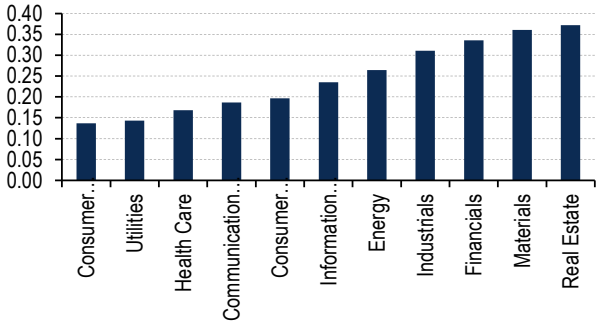
Description

Principal components regression is a technique that is based on principal components analysis (PCA). PCA is a statistical procedure that uses transformations to convert a set of correlated variables into a set of linearly uncorrelated variables called principal components. We recently applied this technique in a Macro Betas note, applying PCA to 40 macro variables which classified idiosyncratic versus macro sensitive US stocks.

Application: discovering the most predictive macro variables for a given stock

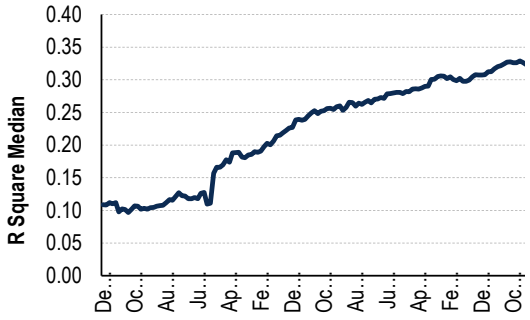
In PC regression, instead of regressing the stock returns against all macro variables, the principal components of the macro factors are used instead. We used a principal components (PC) regression to estimate a multivariate model, which helps reduce the dimension of our macro universe while also avoiding multicollinearity concerns. Chart 6 shows that the combination of all the macro factors has an increasing explanatory power (R-squared) on US stock returns over time (with the exception of a recent dip). At the sector level, Real Estate and Materials move the most with macro factors, vs. Consumer Staples and Utilities least (Chart 5).

Chart 5: Real Estate and Materials move most with macro factors while Consumer Staples and Utilities less so
Median R-squared from principal components regression per sector
12/2008-11/2018



Source: Haver & BofA Merrill Lynch US Equity & US Quant Strategy

Chart 6: Macro movements have increasingly mattered ov
Median R-squared of BofAML US covered stocks to all mo
factors based on rolling 10-year principal components regr
12/2005-11/2018



Source: Haver & BofA Merrill Lynch US Equity & US Quant Strategy

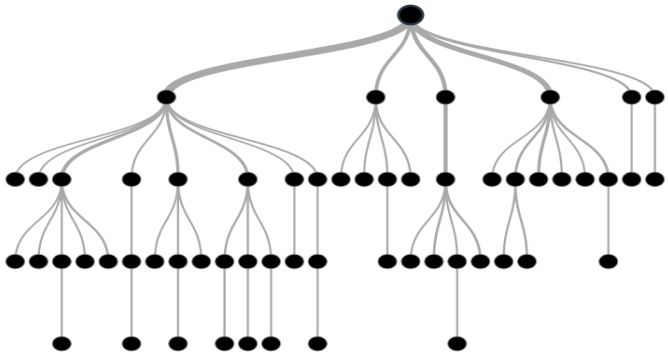
Supervised Learning: Tree-Based Models

Separating the tree from the forests

Before we can understand what a Random Forest is, we first have to understand what a single Decision tree is. As an example, imagine if you were trying to decide to play tennis with friends. The most predictive factor would be the weather forecast, followed by either humidity or wind conditions as per Exhibit 9.

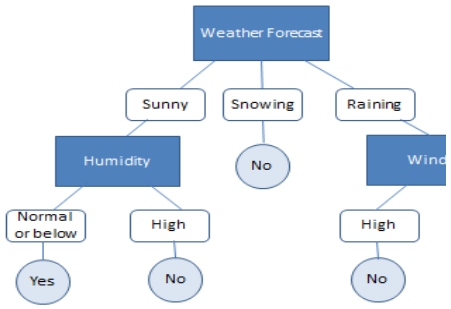
The outcome variable for Decision Trees is binary classification and can also be used for regression. A single Decision tree typically suffers from high variance but has low bias. This issue can be resolved by creating aggregations (i.e. ensembles) of many decision trees as will be explained in the Random Forests section further below.

Exhibit 8: General decision tree example



Source: Skymind.ai

Exhibit 9: Toy decision tree example on whether to play ter



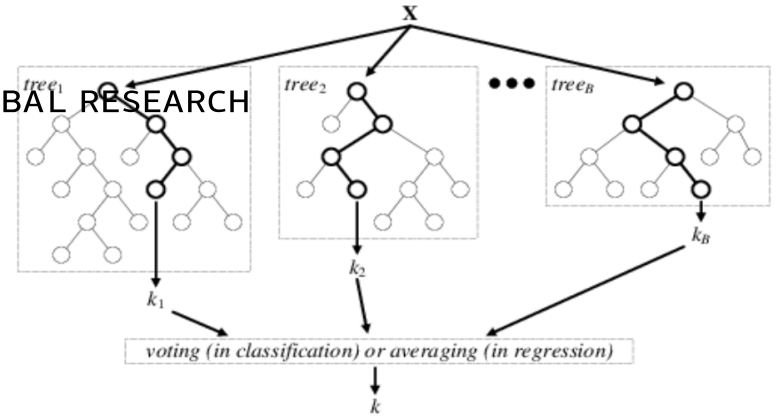
Source: BofA Merrill Lynch US Equity & US Quant Strategy

Random Forests

Description

Random Forests is a commonly used machine learning model designed for ensemble learning using a multitude of decision trees where each tree randomly samples a subset of predictor variables in order to make a prediction. Randomly sampling predictor variables decorrelates the trees from one another. This might sound strange, but supposing there is a strong predictor in the dataset amongst a range of moderately strong predictors, randomly selecting subsets of the predictor variables achieves two objectives: 1) it ensures you capture other information from the other predictors, and 2) it reduces the variance of your prediction by not heavily relying on one predictor. Exhibit 10 illustrates a general case of how many decision trees can be aggregated.

Exhibit 10: Shows example of how Random Forest is aggregated



Source: Herbinet, Individual Project Report, Imperial College of Science

Application: combing traditional quant factors with an alternative dataset (NLP)

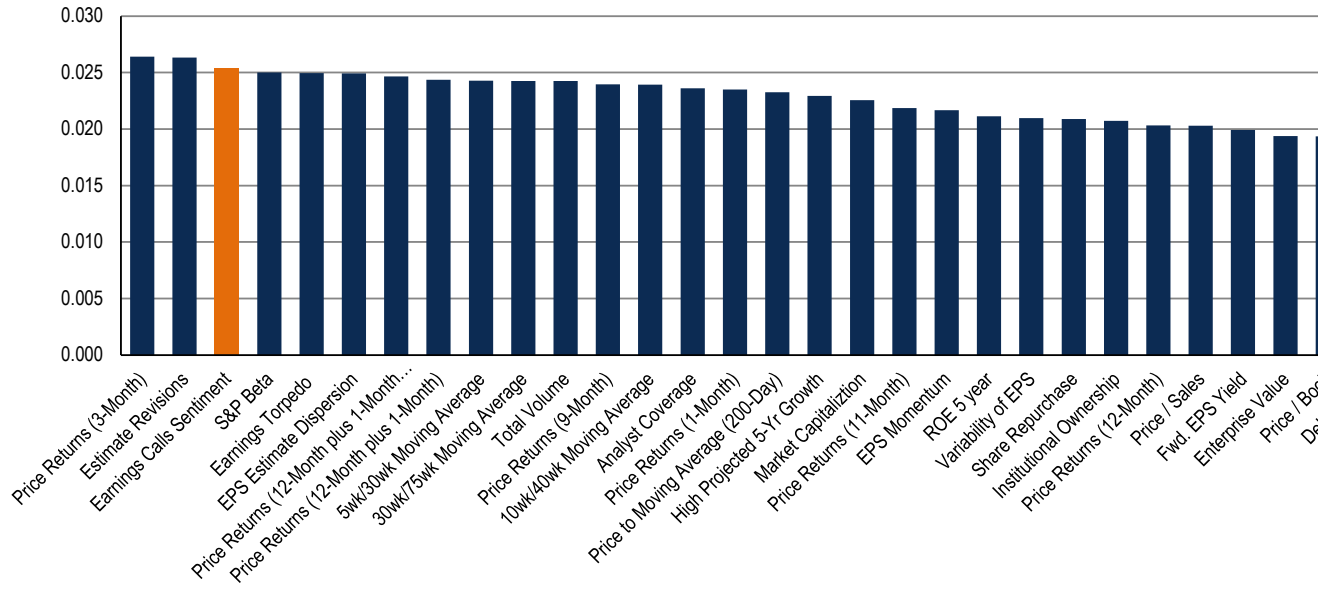
In our recent earnings sentiment note (http://rsch.baml.com/r?q=8a-QczDcP5250fVPs-yA!g), we ran a Random Forest model with 47 of our other quantitative (http://rsch.baml.com/r?q=DrvSc1bmpuMTkpyGA3UqXw) factors (valuation, earnings revisions, momentum, etc.) combined with earnings sentiment using a random forest model. Our back-testing found that earnings sentiment combined with our other quantitative factors improved the Sharpe Ratio to 0.91 from 0.67 from Mar 2014 to Feb 2019. Doing this would have intended to achieve three things: 1) Capture nonlinear relationships and interaction effects between factors using Random Forest model where typical Ordinary Least Squares (OLS) regression fails to detect. 2) Test the earnings sentiment signal on a monthly time horizon. 3) See how well the earnings sentiment factor held up in the presence of 47 other quantitative factors. If it held up in the presence of other factors, then this suggests it brings uncorrelated information not captured in traditional factors.

Identifying which factors mattered

As part of the diagnostic tools of a Random Forests model, the factor importance plot (Chart 7) illustrates which factors were the most effective for out-of-sample prediction. 3 month price change, earnings estimate revisions and earnings call sentiment are the top three factors for predicting next month relative returns.

Chart 7: Earnings call sentiment ranked high, in the top three overall, in terms of predictive power

Feature Importance shows each factors' predictive contribution to the Random Forest model from the last calibration of the walk forward analysis May 2018



Source: BofA Merrill Lynch US Equity & US Quant Strategy, Amenity Analytics, Factset
Back tested performance is hypothetical in nature and reflects application of the framework prior to its inception date and is not intended to be indicative of future performance

Bagging

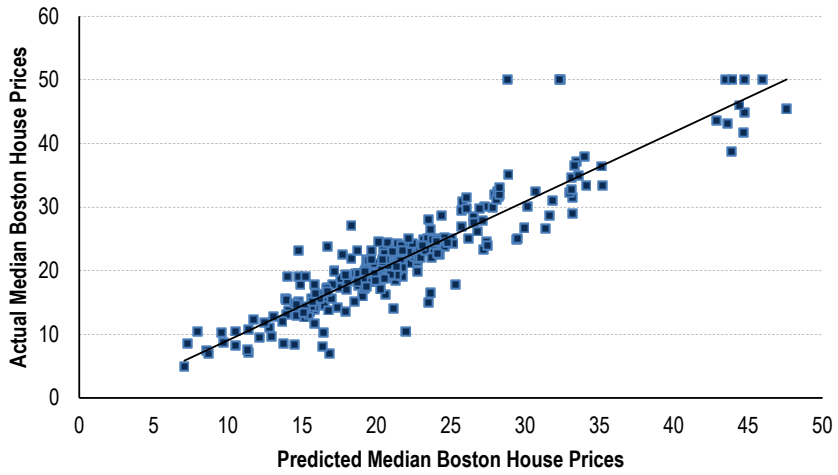
Description

Bagging is a variation of Random Forests. Conceptually the approach is the same, but instead of randomly selecting factors among the universe we randomly select observations without excluding factors. Bagging is a beneficial technique that seeks to reduce the variance of the predictions. Simply put, the algorithm estimates many decision trees using different subsets of the data and averages all the predictions from the decision trees.

Application

To illustrate a toy example of Bagging as per Hastie, etc all(*) , we utilize a famous Boston dataset that is commonly used within the data science community. There are thirteen factors ranging from crime per capita, average number of rooms per dwelling, tax rates, etc., in order to predict median home prices. As per Chart 8, we demonstrate the effectiveness a Bagging model on an out-of-sample (e.g. test dataset) that achieves a Mean Squared Error(*) of 13.16. Note that this can be further improved if one uses a Random Forest model with a default random factor selection of six to a Mean Squared Error of 11.31.

Chart 8: Toy example using Bagging to predict Boston Median House Prices
House prices are in $1000s



Source: BofA Merrill Lynch US Equity & US Quant Strategy

eXtreme Gradient Boosting (e.g. XGBoost)

Description

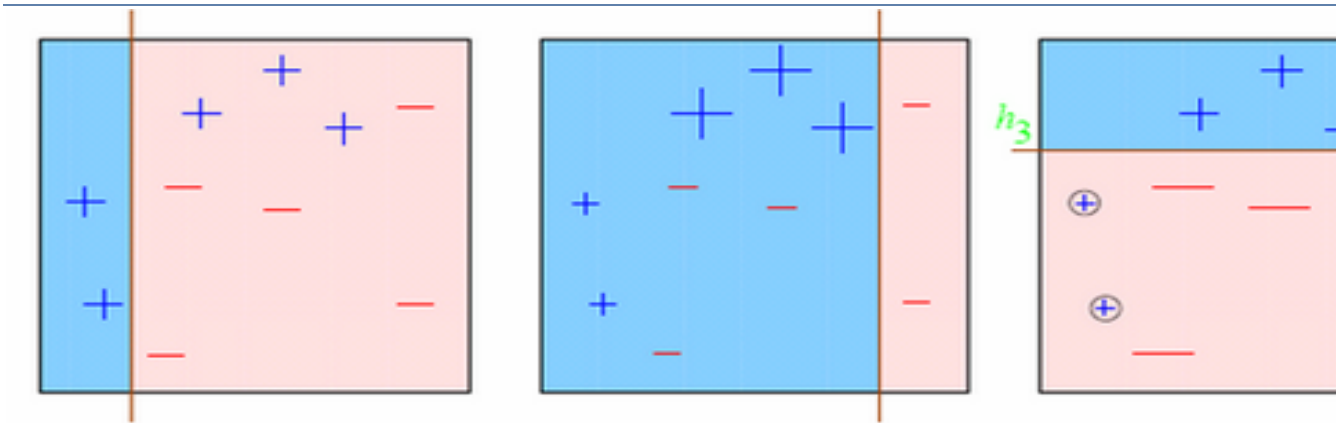
XGBoost (eXtreme Gradient Boosting) is a model that has gained in popularity in recent years, partly due to the fact that it is the model of choice for winning data science competitions on Kaggle (Kaggle(*) is the most popular data science website where companies can post open problems with data sets and offer monetary prizes). The model is good at learning what it doesn't know by iteratively learning from its mistakes to achieve improved prediction capabilities. The XGBoost is still using an aggregation of decision trees as is Random Forest, but it does it in a clever way so as to construct new decision trees that use the forecast error to make better predictions.

Further Description through toy example

Before we can understand XGBoost, we have to come to terms with the concept of boosting. The essence of a Boosting model (or algorithm) is that it is trying to learn from its mistakes (e.g. forecast error). Exhibit 11 is a scenario where the task of the model is to split the square whereby one side contains all positive values and the other side contains all negative values. The model applies either a vertical or horizontal line to represent predictions.

The first chart on the left misclassifies three positive examples (i.e. the three plus signs on the top right). The Boosting model will try to learn from this on the 2nd iteration (2nd chart) and assign higher weights to the three positive signs to minimize prediction error for the 2nd attempt (i.e. see how the vertical line shifts to the right to now classify all the positive signs correctly). However, this now results in misclassifying the three negative signs (in the bottom middle). The process keeps repeating as the weights adjust for the misclassified observations until the model minimizes the forecast error to an acceptable level.

Exhibit 11: Boosting algorithm weak learner example



Source: Schapire R. 2012, Boosting: Foundations and Algorithms

How boosting prevents over-fitting via "extreme"

As boosting is generally designed to over-fit, the "extreme" part of gradient boosting was introduced in order to prevent over-fitting by including a penalty term during the fitting procedure (conceptually this is similar to how the Ridge and Lasso sections incorporated a penalty term). In addition to the penalty parameter above that can be used to tune during the fitting procedure, there is also a list of hyperparameters ranging from learning rate (used to guide how fast the model can learn), max tree depth, etc. that can be used to prevent over-fitting(*). Deciding which parameters to choose can become overwhelming computationally as there can be many input levers (i.e. hyperparameters) in which a user can drive the XGBoost car, but a good rule of thumb is that the learning rate is the most important of them all.

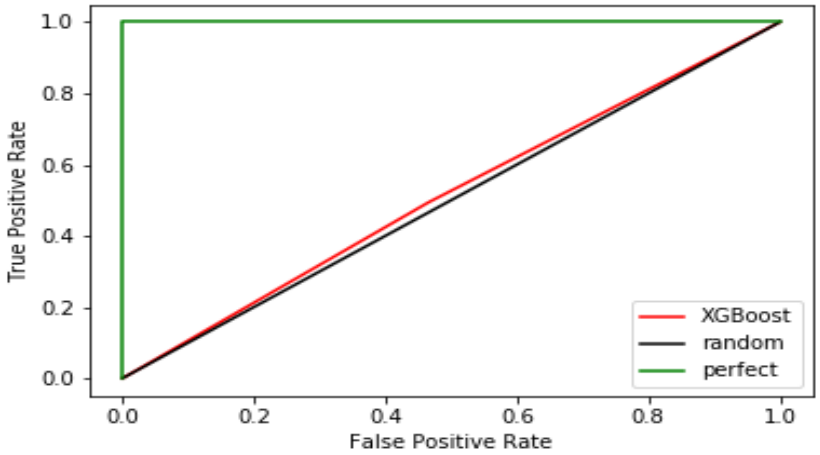
Application of XGBoost

As an application of XGBoost, we reused the dataset described in Ridge/Lasso regression where our aim was to predict out/underperformance of a stock (defined as being above or below the median returns of the Russell 1000) using valuation, momentum and earnings revisions factors among many others. We ran the XGBoost model from Jan 2008 to Dec 2016 and the out of sample period from Jan 2017 through Mar 2019.

Exhibit 12 illustrates the ROC(*) curve by showing the percentage of time the model would have predicted correctly (51.2%). After feeding the XGBoost model with over 70 quantitative factors, it is able to generate a probability of out/underperformance as it gets fed incoming data. One can then develop a trading strategy using the forecasts generated by the model. As per the chart, since the red line is above the black line this indicates that the XGBoost model has better than a 50-50 chance. The green line represents a perfect model. Most data science problems achieve higher accuracy rates (i.e. the red line lies closer to the left and nearer to the green line) but in finance if an accuracy rate is marginally above chance then this can still be a profitable trading strategy. As an example, constructing a long-short quintile strategy using the XGBoost predicted probabilities would have generated a back-tested Sharpe Ratio of 0.65 over the out of sample period.

Exhibit 12: XGBoost Receiver Operating Characteristic (ROC) curve
Red represents the percent accuracy of the XGBoost model, green is if it would have been a 100% accurate and black is consider random with 50-50 chance

BofA GLOBAL RESEARCH



Source: BofA Merrill Lynch US Equity & US Quant Strategy

Boosting versus Bagging/Random Forests

As per Prado's new book(*), he points out that Boosting's main advantage is that it reduces both variance and bias in predictions. However, correcting the bias comes at great risk of over-fitting. The main difference is that Boosting addresses under-fitting while Bagging and Random Forests addresses over-fitting. As mentioned previously, over-fitting is the primary concern in finance as the signal to noise ratio is low. Therefore, it is advisable to generally use bagging or random forests over boosting in finance for this reason. However, XGBoost which includes the penalty term as part of the fitting process can minimize over-fitting, making the model competitive with Bagging and Random Forests. We now turn to discussing Deep Learning models in the next section which have some similarities in terms of having many input levers (i.e. hyperparameters) available.

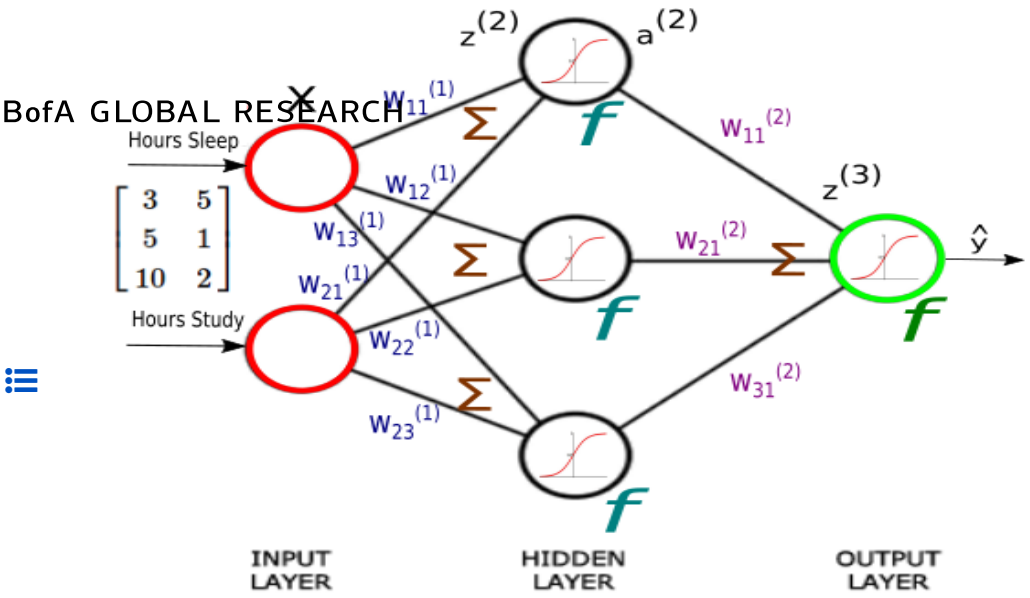
Deep Learning

Deep learning is a set of interconnected algorithms that are loosely based on the structure of a brain. The advances incorporating Deep Learning have had recent A.I. breakthroughs such as Google's AlphaGo program beating the best Go player in the world. This was achieved by combining Deep Learning with Reinforcement Learning(*). Deep Learning has experienced a significant improvement in accuracy rates in fields such as image recognition and natural language processing but has yet to make a significant impact in financial markets.

The basics of neural networks

To understand Deep Learning, we first have to understand what a neural network is. Neural networks were inspired by how our human brain connects information together. At a neural network's core, it is filled with straightforward calculations of multiplying weights with the dataset (similar to linear regression) followed by basic math transformations. We show a toy example in Exhibit 13 that uses the number of hours slept and studied to predict test scores. The first part of the exhibit demonstrates the input (e.g. the hours slept and hours studied) to be fed into a hidden layer (this step involves multiplying weights and nonlinear transformations(*)). The process repeats until a final output layer is produced to generate a forecast (e.g.. test score). The technical term for the process is known as forward propagation.

Exhibit 13: Neural Network Toy Example using number of hours slept and hours of study to predict test scores



Source: bogotobogo

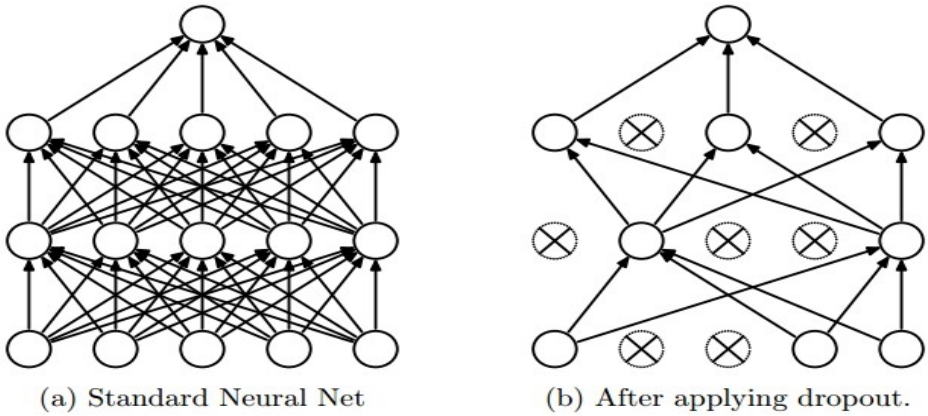
Getting deeper in neural networks

Typically the set of input levers (i.e. hyperparameters) that gets decided for neural networks is the learning rate (for the optimization), choice of nonlinear transformation (i.e. activation function), cost function, number of hidden layers and number of neurons for the hidden layers. For a neural network to be called a Deep Learning network, the rule of thumb is typically more than 3 hidden layers. In order to determine the best weights for prediction, a fitting process is applied starting from the final output prediction and working backwards. This is known as backpropagation where the goal is to minimize the forecast error retroactively.

Dropout: Preventing Neural Network Overfitting

As powerful as some of the deep learning algorithms are, some of the models used in NLP translation models can have tens of millions of parameters. As per the over-fitting section, as model complexity increases, so does the variance leading to over-fitting. Exhibit 14 shows an important contribution that came from Hinton(*), et al, where the key idea of *dropout* is to randomly drop units and connections in an attempt to prevent the algorithm from overly relying on them. Introducing *dropout* has significantly improved out of sample prediction for a wide range of deep learning models.

Exhibit 14: Neural Network with and without dropout



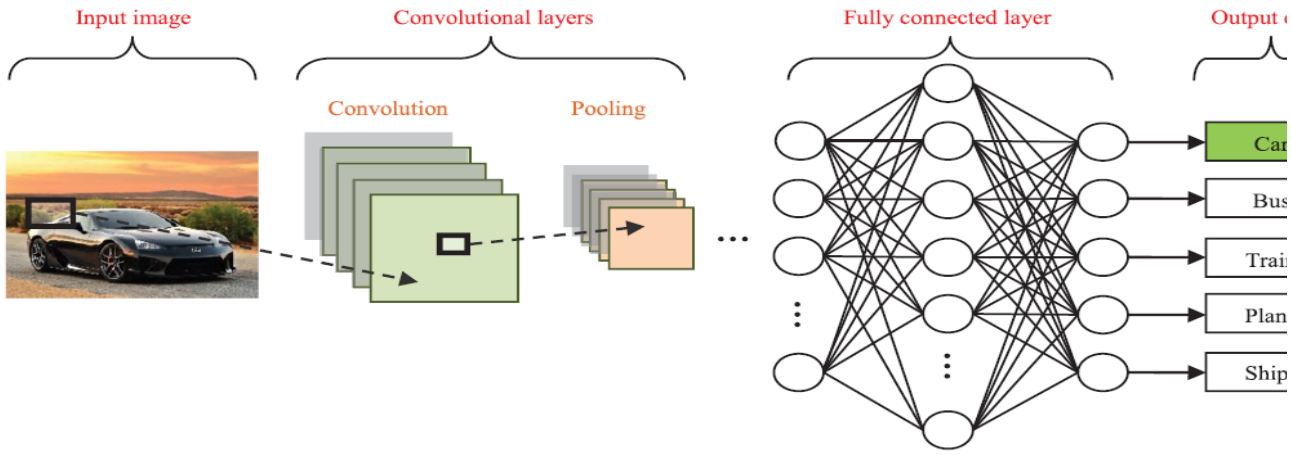
Source: Srivastava, Hinton, Krizhevsky, Salakhutdinov, 2014, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting."

Convolutional Neural Networks (CNN)

Description

Convolutional Neural Networks (CNN) are a type of artificial neural network that, instead of a single level input, use 3-D data (Exhibit 15). This is the model of choice for image recognition as most images can be broken down into several layers at the pixel level. CNN at its heart takes three-dimensional inputs to slice and dice into further three-dimensional hidden layers. The final objective is to feed it to a single hidden layer known as a fully connected one before it can associate the image to the right category (i.e. car). The computation of the weights is as described above using the backpropagation concept. See below for a simple example of how this can be applied to a time series financial example in a non-3-D case.

Exhibit 15: Convolutional Neural Network example of classifying image into the appropriate animal (i.e. car)

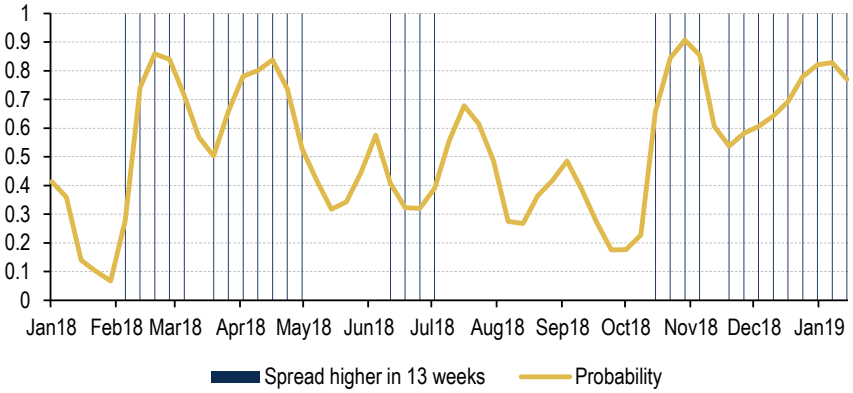


Source: Rawat, Wang, Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review

Application: predicting the S&P 500 returns over 10-year yield

We attempt to predict the forward difference between monthly total returns of the S&P 500 and US 10-year bonds (i.e. the spread). Here we are demonstrating a simple financial application of at least one convolutional hidden layer with an assumed 50 percent dropout (i.e. the percent of units and connections dropped as per above). In order to simplify the prediction, we codify the 13-week forward spread as 1 if above the current spread and 0 otherwise. We use macroeconomic, technical and fundamental factors as input variables (see Table 5 in appendix). The output of the model calculates the probability of higher or lower spreads. The model has been trained on the first 21 years of data 1990 - 2011 and then tested since 2011 - 2018.

Chart 9: Convolution Neural Network application predicting equity minus bond spread



Source: BofA Merrill Lynch US Equity & US Quant Strategy

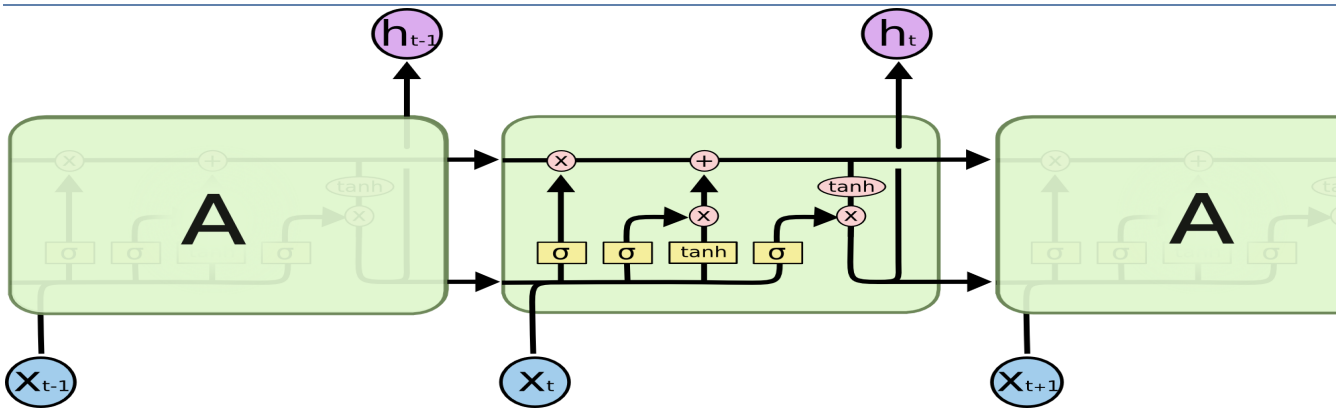
Back tested performance is hypothetical in nature and reflects application of the framework prior to its inception date and is not intended to be indicative of future performance

Long Short-Term Memory (LSTM)

Description

The particular branch of deep learning best suited for language prediction is called recurrent neural networks (RNN). RNN's are designed to handle sequenced data such as language or time series. Within RNN's, one can utilize a long short-term memory (LSTM). LSTM models have advantages over RNN models as they are composed of input, output and forget gates, allowing it to remember parts of the sequence that matter for out of sample prediction and forget parts of no consequence. Exhibit 16 demonstrates the intricate details of the algorithm with "X" representing the data, "h" as the hidden layer and some of the inner workings of the input, output and forget gates(*).

Exhibit 16: Long-Short Term Model (LSTM) algorithm details



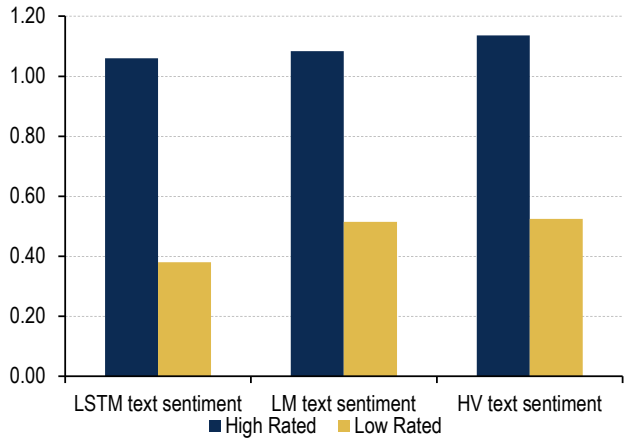
Source: Colah's blog

Application: training LSTM model to improve positive/negative text sentiment

We recently tested the efficacy of Glassdoor ratings sentiment data on returns and found that employee ratings can lead to better risk-adjusted returns. However, since traditional 'bag of words' or dictionary-based sentiment approaches (those that count positive vs. negative words per review) face criticism for disregarding word order, we tested the efficacy of training a deep learning long short-term memory (LSTM) model on 20,000 employee reviews from 2008-2009 to make predictions for 2010 to 2018 using the trained model. Please see the Natural Language Processing (NLP) section for more details on sentiment analysis.

Chart 10: LSTM Glassdoor sentiment is competitive to LM and HV based sentiments
Sharpe Ratios LSTM sentiment quintile stocks, Jan 2013 - Dec 2018, quarterly rebalancing

BofA GLOBAL RESEARCH



Source: BofA Merrill Lynch US Equity & US Quant Strategy & Thinknum
Back tested performance is hypothetical in nature and reflects application of the framework prior to its inception and is not intended to be indicative of future performance

We found that a deep learning sentiment fran comparable with LM and GI based approaches preliminary results indicate that the LSTM ap superior to the dictionary-based measure in i unattractive stocks but is inferior in identifyir stocks. This is a topic for further research.

Natural Language Processing (NLP)

As Natural language processing (NLP) is a very active application area within Machine Learning and the high percentage of new data being generated is in the form of text, we dedicate the section below to it.

Sentiment Analysis

Description

Since early 2000, sentiment analysis has grown into one of the most active areas of research within NLP. The goal for sentiment analysis is to computationally study sentiment towards events, topics, issues, products, etc. Text sentiment is simply defined as counting the number of positive relative to negative words as per what a given dictionary prescribes. The two most common dictionaries for sentiment analysis are Loughran-McDonald (LM) and Harvard General Inquirer (GI). LM is customized for financial text documents (10-Qs, earnings transcripts, etc.) while GI is more oriented toward social feedback. Traditionally, sentiment analysis is computed at the document or sentence level by deploying either a dictionary based look up, machine learning or deep learning model (as discussed in the application of LSTM above).

Application: Using text sentiment in employee reviews as a signal for alpha

As discussed above, our analysis of Glassdoor employer reviews identified potential alpha opportunity for stocks with higher vs lower Glassdoor ratings. The Glassdoor website maintains employer ratings as well as written reviews where employees can enter responses in a pros and cons section. Our analysis suggested that text sentiment has the potential for identifying alpha opportunity though we had found inconsistencies within reviews. For example, a reviewer might assign a positive rating to a company that was at odds with a strongly negative written response in the pros and cons section. In order to assess sentiment polarity, we applied both the LM and GI dictionaries given that Glassdoor falls somewhere in the middle of this spectrum. The tone of text can vary from the overall rating at times. For example, the 5-star rating below has a negative sentiment but is in contrast with the positive rating.

Pros

"The only pro I can think of is that it pays the bills, or at least some of them. It used to be a great place to work about 4 years ago"

Cons

"Low wages, company is always restructuring and a lot of people end up losing their jobs, little or no notice on store closures or layoffs, company will fire you or force you to quit so you don't draw unemployment or severance, upper management doesn't care about its employees"

Aspect Level sentiment using event categories

Aspect Level sentiment varies from the standard method of sentiment analysis as it considers both the sentiment and the target information, where target is defined as an entity or event. For example, in the sentence "the screen is very clear but the battery life is too short." the sentiment is positive towards screen (the first target) whereas it is negative towards battery life (second target). As a result, Aspect Level sentiment is able to classify sentiment per each target event within a document which can either be aggregated up into an overall sentiment or one can zoom into a relevant event sentiment of interest.

Application: Amenity Analytics specializes in Aspect Level

In our recent note (http://rsch.baml.com/r?q=8a-QczDcP5250fVPs-yA!g) using Amenity Analytics data, Aspect Level sentiment is applied to hundreds of target event classifications categorizing sentiment per each earnings call transcript. It uses NLP assisted by Machine Learning as part of its proprietary algorithm. Table 1 shows a summary of Amenity's event target categories average sentiment and total count with at least 10,000 observations for stocks traded on NYSE and NASDAQ from Jan 2010 to Feb 2019.

In general, Table 1 showed that of the 70 target event categories, the average sentiment was positive, suggesting that companies tend to speak in a positive tone during earnings calls. For example, each time a company would speak about "Record Results", it was in a positive tone on average. Further, it is not surprising to find events such as "tailwinds", "new product" and "record results" to be classified on average as positive whereas target events such as "headwinds" and "deception" are considered negative on average.

Table 1: "Tailwinds", "Record Results", "Investment - Internal" had the highest positive sentiment whereas "Headwinds", "Deception" and "Weathe negative sentiment. For example, each time a company would speak in regards to "Record Results", it was on average in a positive tone. Across the entire earnings call transcripts from Jan 2010-Feb 2019 this table contains the overall summary per the most frequent event types.

Event Name	Average Sentiment	Count	Event Name	Average Sentiment
Tailwinds	0.63	120145	Financial Commentary	0.31
Record Results	0.59	32525	Brand - Credibility - Image	0.29
Investment - Internal	0.57	56381	Financial Expectations	0.27
Strategic Alliance	0.55	42970	Facilities	0.27
Stock Buyback	0.54	33331	Supply - Demand	0.25
New Product	0.53	26963	Margin Results	0.23
Asset Sale	0.52	16094	Margin Commentary	0.22
Debt Financing	0.51	20650	Workforce	0.20
Strategic Commentary	0.51	47602	Legal - Regulatory	0.12
Contract - Agreement - Deal	0.50	52127	Price	0.08
Dividend	0.50	36295	Forecast Change	0.07
Market Share	0.49	98496	Tax	0.04
Restructuring	0.47	18181	Cost	0.01
Service - Product Deal	0.41	32060	Currency	-0.07
Product Update	0.39	31738	Inventory	-0.10
Financial Results	0.38	192484	Consumer Trend	-0.13

Emerging Market	0.37	18538	Merger - Acquisition	-0.31
Customer Traffic	0.37	59958	Competition	-0.40
Business Commentary	0.37	199708	Weather	-0.41
Executive Change	0.34	11596	Deception	-0.56
Capacity - Production	0.33	104935	Headwinds	-0.66
Liquidity	0.33	15754		

Source: Amenity Analytics

The average sentiment is shown per event type as which can range from 1 to -1 where 1 is the highest sentiment and -1 is the lowest sentiment. The calculation for sentiment is (positive count - ne (positive count + negative count + 1). Amenity Analytics has over 500 event categories but we limit it to event categories with at least 10,000 observations.

Additional Concerns

What happens when things permanently change?

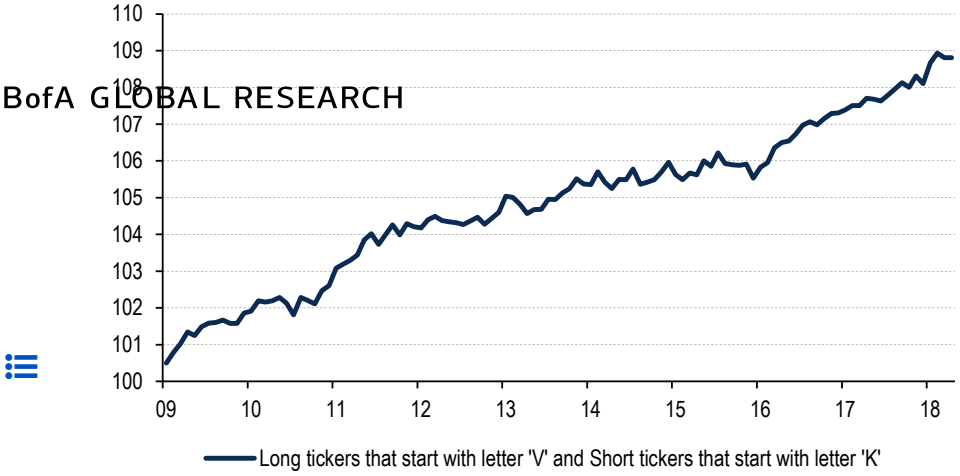
In finance, the so called signal to noise ratio is much lower than typically data science type problems. A typical data science problem could be trying to classify if an image is a dog or not, where these images will typically not change drastically throughout time. However, in the financial markets, the state of which things are changing fluctuates throughout time. Strategies will come and go. The mix of central bank policies have been updated, the mix of market participants is changing, geopolitical stresses come and go. The technical statistics term for this is called Non-Stationarity. In essence, this just states that things will change over time.

Multiple Testing & Spurious Correlations

Over-fitting can also be thought of as finding a model that works purely by chance given enough loops with different parameters. The main objective in data science is to discover relationships and insights from the data whereby actionable decisions can be made. Some data science outsourcing vendors claim they have searched hundreds of statistical models to ascertain what model is best suited for out of sample prediction. There is one major problem: if you try enough times you'll eventually find a pattern by sheer luck. Take Chart 11 for example(*), which shows a nonsensical strategy that goes long any ticker that starts with the letter 'V' and goes short any ticker that starts with the letter 'K' from the Russell 1000. This strategy would have generated a Sharpe Ratio of 1.29, but this is after trying 400 different combinations with the starting letters of the tickers. As noted by Harvey, given 20 random strategies, one will likely exceed by pure chance. Therefore, we recommend tracking the number of times a strategy is tried with different configurations. Please see Harvey for other statistical tests one can utilize to interpret results when multiple testing has occurred.(*)

Chart 11: Nonsensical Over-fitting Back-test shows after 400 iterations a researcher could find something by random luck

The strategy goes long tickers that start with the letter 'V' and short tickers that start with the letter 'K' in the Russell 1000



Source: BofA Merrill Lynch US Equity & US Quant Strategy

Other BofAML Machine Learning Research

Below we highlight additional BofAML research reports that apply machine learning to their investment analysis.

K Nearest Neighbors, Naïve Bayes, Support Vector Machines, Kalman Filter

We specifically call out our Global Rates and FX strategists' primer on the application of machine learning in Rates and FX markets (http://rsch.baml.com/r?q=X3WWQMXoeAQQVnX78wsqkw) given the wide coverage of machine learning models including K-nearest neighbors, Naïve Bayes, K-means clustering, Support Vector Machine, Kalman filter, and Principal Component Analysis.

Other miscellaneous Machine Learning models

Please see the below table for other additional Machine Learning models contained within BofAML research.

Recent BofA Merrill Lynch Global Research Reports

Title: Subtitle	Machine learning model	Primary Author
European Metals & Mining: Quant signals lining up for miners	Indicator developed via multiple models (Random Forests, Elastic-Net, Ridge, Support Vector Machines, Gradient Boosting, eXtreme GBT, k-Nearest Neighbors, and Mean Response)	Jason Fairclough
FICC Portfolio Monthly: Oiling the oil machine (http://rsch.baml.com/r?q=PcxRuXf35hnHvGBRZnD!fw)	Dynamic Correlation Filter, a Principal Component Analysis (PCA), Partial Least Squares (PLS), Lasso, and Ridge	QIS Research
GEMs Viewpoint: Introducing EMMI: a language-based sentiment indicator for EM central banks (http://rsch.baml.com/r?q=Z0Vm7NmyaI99oXzbn9lCGw)	Rules based NLP	David Hauner, CFA
Capital Goods - Global: China quant analysis suggests further sector gains in July (http://rsch.baml.com/r?q=!0s5cVykFbm3nWUEwzwVXg)	Indicator developed via multiple models (Random Forests, Elastic-Net, Ridge, Support Vector Machines, Gradient Boosting, eXtreme GBT, k-Nearest Neighbors, and Mean Response)	Mark Troman
FICC Portfolio Monthly: FX and commodities: chicken and egg (http://rsch.baml.com/r?q=55-5CWWmwDDLj9mdjCrObQ)	Various machine learning methods including random forest	QIS Research

References

James, Witten, Hastie, Tibshirani, 2013, "An Introduction to Statistical Learning with applications in R"

Goodfellow, Bengio, Courville, 2016, "Deep Learning"

Lopez De Prado, 2018, "Advances in Financial Machine Learning"

Appendix

The technical definitions and throughout, we are inspired and guided as per Hastie, Tibshirani, Witten, James in *An Introduction to Statistical Learning with Applications in R*.

Cross-Validation: A key way to minimize over-fitting

Several techniques have been developed to deal with over-fitting (e.g. cross-validation, regularization, early stopping or dropout in the case for deep learning) in Machine Learning.

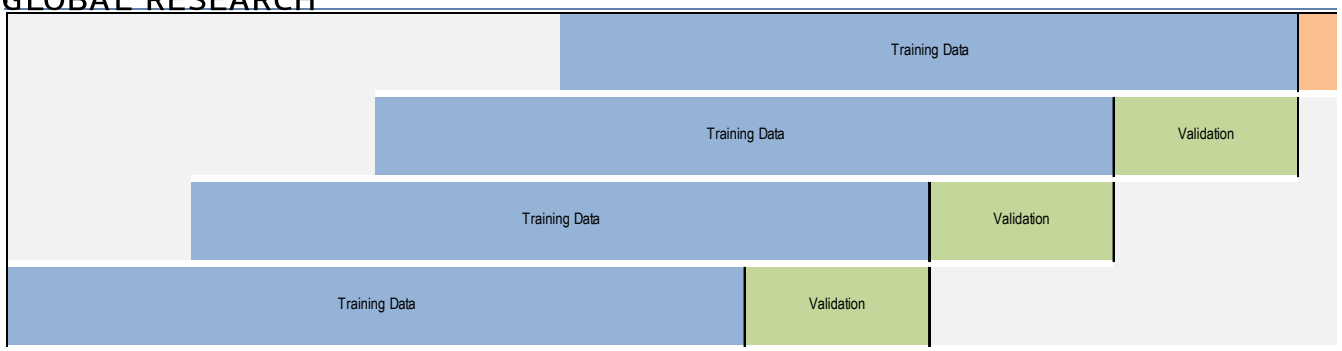
One of the ways to minimize over-fitting is through a data sampling recipe known as cross-validation. The key objective is to make sure a given dataset (i.e. training) that is used for a particular model can be predictive on an unseen dataset (i.e. testing). Specifically, the data sampling recipe that is primarily used in data science can be seen in Exhibit 17, whereby the data used for estimated the model is partition into training and validation datasets to eventually predict an holdout (out of sample period). In Exhibit 17, a 5-fold cross-validation is considered. This means that the same model is estimated with five different data samples trying to predict five different validation periods. When this procedure is complete, the process is to take the average mean squared error and find the best the parameters (i.e. regularization) that are associated with that model to make a final prediction on the holdout dataset (rule of thumb is that this is usually 20%). In finance, time dependence matters (i.e. we cannot use data from the present to predict the past), so we can modify the cross-validation procedure to ensure we do not include forward looking information by modifying the data sampling recipe (see appendix Exhibit 18).

Exhibit 17: Cross Validation example showing typically 5-fold data sampling process to predict an out of sample holdout period

Training Data	Training Data	Training Data	Training Data	Validation	H
Training Data	Training Data	Training Data	Validation	Training Data	H
Training Data	Training Data	Validation	Training Data	Training Data	H
Training Data	Validation	Training Data	Training Data	Training Data	H
Validation	Training Data	Training Data	Training Data	Training Data	H

Source: BofA Merrill Lynch US Equity & US Quant Strategy

Exhibit 18: Time series cross-validation example with 3-fold data sampling
This is similar to the traditional cross validation with the exception the time ordering is maintain. This makes sure that no future information can be past



Source: BofA Merrill Lynch US Equity & US Quant Strategy

Technical definitions

K-Means Clustering Algorithm

1. For 1 to K, generate a random number for each observation. This initializes the algorithm.
2. Loop until cluster assignments stop changing:
    a. For each of the K clusters, compute the cluster centroid. The kth cluster centroid is the vector of the p feature means for the observations in the kth cluster.
    b. Assign each observation to the cluster whose centroid is closest (where closest is defined using Euclidean distance).

Hierarchical Clustering Algorithm

1. For each of n observations and a distance measure (e.g. Euclidean distance) of all the pairwise dissimilarities. Treat each observation as its own cluster.
2. For i = n, n − 1, . . . , 2:
    a. Examine all pairwise inter-cluster dissimilarities for i clusters and identify the pair of clusters that are the least dissimilar (i.e. most similar). Fuse these two clusters.
    b. Calculate the new pairwise inter-cluster dissimilarities among the i − 1 remaining clusters.

Glossary: Jargon Buster

Table 2 points out some common relatives that Machine Learning has with Statistics in terms of terminology. If fact, models like Principal Components Analysis, Logistic Regression, K-Mean Clustering and Neural Networks have existed within the Statistics community long before they became considered part of the Machine Learning toolkit. It is followed by Table 3 that defines common jargon used in Machine Learning and the world of Big Data.

Table 2: Machine Learning to Statistics map

Machine learning terminology	Statistics terminology
training set	in sample
test set	out of sample
hypothesis	classifier
learning	fitting
response	Label
teacher	statistician
weights	parameters
Instance / example	data point
False positive	Type 1 Error

False negative	Type 2 Error
Features	Variables

Source: BofA Merrill Lynch Global Research

Table 3: **Buzzwords in the world of Big Data and Machine Learning**

	Description
Alternative data	Non-traditional data used in the investment process. Contrary to traditional datasets (e.g. PMI, CPI), alternative data tends to be unstruc find and/or large in volume (e.g. twitter feed, satelite images, news articles) that requires significant computational power to analyse.
Artificial intelligence(AI)	A computer science field that tries to teach machines to solve problems the way humans do (e.g. speech recognizing, image recognition
Big data	Usually used to describe datasets that are big and complex that traditional techniques and software are inadequate to deal with them. In also refers to techniques that extract value from data that aren't necessarily huge in size.
ChatBot	Software that conversates with a human. The chatbot tries to simulate human conversation with the help of artificial intelligence. SIRI an examples of them. Bank of America also launched Erica early in 2018 that answers client's daily banking questions.
Classification	It is supervised learning method where the output variable is a category, such as "positive" or "negative" or "Yes" and "No".
Cloud computing	Cloud computing refers to services provided remote servers over the internet, as opposed to on a local server, and its infrastructure is us by a third party vendor.
Clustering	Clustering is an unsupervised learning method used to discover the inherent groupings in the data. For example: Grouping customers or their purchasing behaviour.
Computer vision	A computer science field that tries to teach computers to visualize, process and identify images/videos in the same way that a human vis applications include facial recognition, self-driving cars, etc.
Cross validation	Cross Validation is a technique which involves reserving a particular sample of a dataset which is not used to train the model. Later, the on this sample to evaluate the performance.
Data science	The combination of computer science, statistics, modeling, and artificial intelligence. A cross-discipline focused on getting the most from rich technical environments.
Deep learning	A branch of machine learning. It often uses Artificial Neural Network (ANN) which adopts the concept of human brain to facilitate modelir a vast amount of data and this algorithm is highly flexible when it comes to model multiple outputs simultaneously.
Dropout	Is a technique to randomly drop neuron and connections within a neural network to reduce over-fitting.
Feature engineering	Converting available real life information into datasets that can be fed into standard algorithms.
Feature selection	The process that selects the key subset of original data features in an attempt to reduce the dimensionality of the training problem.
Hadoop	A collection of open-source software utilities that facilitate using a network of many computers to solve problems involving massive amou computation.
Neural Networks (or ANN)	Inspired by the biology of the brain, these ANN's are a huge network of numerous interconnected conceptualized artificial neurons which between themselves. These neurons have activation threshold, if met, are 'fired'. The combinations of these fired neurons results in lear
NLP	Natural Language Processing. A field which aims to make computer systems understand human speech, including techniques to proces categorize raw text and extract information.
Recommender system	Algorithms that attempts to recommend the most relevant items to a particular user, often based on their needs and interests. Real world include Netflix's movie recommendations, Bank of America Mercury's "Recommended for you" section.
Reinforcement Learning	A branch of Machine Learning. Unlike traditional algorithms that minimizes loss, these techniques focuse on actions that maximize rewa
scikit-learn	One of the most popular free machine learning libraries for the Python programming language
TensorFlow	An open source software library developed by Google. It is mainly used for machine learning applications such as building neural networ
Transfer learning	Transfer learning refers to applying a pre-trained model on a new dataset to solve a similar problem.

Source: BofA Merrill Lynch Global Research

Data Definitions

Table 4: Macro variables used for Elastic Net Regression

Macroeconomic	Fundamental	Technical
Conference Board US leading indicator	S&P 500 12M forward EPS	Composite Technical Indicator
China export trade (USD)	S&P 500 12M forward PE ratio	% of S&P 500 members with RSI higher t
OECD China leading indicator	S&P 500 12M trailing EPS	% of S&P 500 members with 52-week hig
Federal funds target rate	S&P 500 12M trailing dividend yield	RSI 30D
Global PMI	S&P 500 leverage (debt to equity ratio)	VIX index
ISM manufacturing	S&P 500 PB ratio	
ISM non-manufacturing	S&P 500 IT 12M trailing EPS	
OECD US leading indicator		
US non-farm payrolls		
Philly Fed business outlook		
US CPI		
US PPI		
US unemployment rate		
Sentiment / positioning	**Equities**	**Fixed Income / credit**
AAII bull / bear ratio	DAX Index	US HY credit (govt OAS)
CBOE put / call ratio	Dow Jones Industrial Average Index	US IG credit (govt OAS)
FMS growth expectations indicator	Dow Jones Real Estate Index	US 10Y TIPS
FMS improved global profit growth indicator	Hang Seng Index	US 3M T-bill
FMS net % overweight US equities	S&P 500 monthly close (lagged 1M)	USD 3M Libor
FMS valuation indicator	S&P 500 monthly close (lagged 3M)	US yield curve (2s10s)
S&P economic cycle factor rotator index	S&P 500 monthly high	
University of Michigan consumer sentiment index	S&P 500 monthly low	
Currencies	**Commodities**	
EURUSD	PSE gold and silver index	
USDCAD	Thomson Reuters core commodities index	
USDJPY		

Source: Bloomberg, BofA Merrill Lynch US Equity & US Quant Strategy

Table 5: Macro variables used in Convolutional Neural Networks

US Payrolls	DXY
ISM Manufacturing	VIX
ISM Prices paid	MOVE
US capacity utilization (% of total capacity)	BAA Spread
Initial jobless claims	SPX EPS
Univ of Michigan Consumer Expectations Index	SPX earnings revision ratio
US avg earnings	SPX PE Ratio
Fed effective rate	SPX PB Ratio
US CPI core	SPX net debt per share
US yield curve (2-10)	SPX DPS
US 10Y	SPX members with 52-week highs
Gold	SPX members with RSI higher than 70
WTI	SPX difference between upper BB and current price
Copper	

Source: Bloomberg

Quantitative Factor Definitions

BofA GLOBAL RESEARCH

Please find the quantitative factor definitions listed below

Earnings Yield: Trailing 12-month EPS divided by month-end price

Forward Earnings Yield: Rolling 12-month forward EPS divided by month-end price

Price/Book Value: Month-end price divided by the most recently reported book value per share.

Price/Cash Flow: Month-end price divided by the most recently reported cash flow. Cash flow is defined as earnings post extraordinary items plus depreciation.

Price/Free Cash Flow: Month-end price divided by most recently reported free cash flow. Free Cash flow is defined as earnings post extraordinary items plus depreciation minus capital expenditures.

Price/Sales: Month-end market value divided by most recently reported sales.

EV/EBITDA: Enterprise Value (Equity Market Capitalization + Long Term Debt + Short Term Debt + Preferred Stock + Minority Interest - Cash & Cash Equivalents) divided by EBITDA (Reported Net Income + Special Items - Minority Interest + Interest Expense + Income Tax Expense + Depreciation and Amortization) - most recently reported.

Free Cash Flow/EV: Free Cash Flow divided by Enterprise Value (Equity Market Capitalization + Long Term Debt + Short Term Debt + Preferred Stock + Minority Interest - Cash & Cash Equivalents). Free Cash Flow is defined as the earnings after extraordinary items plus depreciation minus capital expenditures.

Dividend Yield: Indicated dividend divided by month-end price.

Dividend Growth: The growth between trailing 4-quarter total common dividends and year-ago trailing 4-quarter total common dividends.

Share Repurchase: The year-to-year change in shares outstanding

Rel Str - 30Wk/75Wk MA: The ratio of the 30-week moving average of price to the 75-week moving average.

Rel Str - 5Wk/30Wk MA: The ratio of the 5-week moving average of price to the 30-week moving average.

Rel Str - 10Wk/40Wk MA: The ratio of the 10-week moving average of price to the 40-week moving average.

Price/200-Day Moving Average: A ratio between month-end closing price and average closing price over the last 200 days.

Price Return - 12-Month Performance: Absolute price return over the last twelve months.

Price Return - 9-Month Performance: Absolute price return over the last nine months.

Price Return - 3-Month Performance: Absolute price return over the last three months.

Price Return - 11-Month Performance: Absolute price return from one year ago, ignoring the most recent month.

Price Return - 12-Month and 1-Month Performance: Equal weighted rank of stocks by (1) highest price return over the last twelve months and (2) highest price return over the most recently ended month.

Price Return - 12-Month and 1-Month Reversal: Equal weighted rank of stocks by (1) highest price return over the last twelve months and (2) lowest price return over the last one month.

Most Active: Stocks have the highest monthly share trading volume.

Earnings Momentum: The difference between 12-month trailing EPS and year-ago 12-month trailing EPS divided by year-ago 12-month trailing EPS.

Projected 5-Year EPS Growth: The five-year EPS growth rate estimated by BofAML Fundamental Equity Research. If no BofAML estimate exist, then I/B/E/S Mean Long Term Growth Estimate is used.

BofA GLOBAL RESEARCH

Forecast Earnings Surprise: A forecast earnings surprise variable which compares BofAML estimates to those of the consensus after adjusting for the range of estimates. Stocks are ranked from 1 to 10, with 1 being among the most optimistic, relative to the consensus, 10 being among the most pessimistic. Consensus estimated earnings data are courtesy of I/B/E/S. If the projected Surprise is greater than 13 standard deviations, the stock is excluded as an outlier.

EPS Estimate Revision: The difference between the I/B/E/S FY1 estimate and that of three months ago divided by the absolute value of I/B/E/S FY1 estimate of three months ago.

Equity Duration: An adaptation of our Dividend Discount Model which measures the interest-rate sensitivity of a stock. Longer durations (higher numbers) suggest more interest-rate sensitivity.

Earnings Torpedo: I/B/E/S FY2 estimate less latest actual annual EPS divided by month-end price.

Return on Equity One-Year Average: Net income divided by average equity provided.

Return on Equity Five-Year Average: Five-year average return on equity.

Return on Equity One-Year Average (Adjusted for Debt): The ROE of companies with higher debt levels are considered lower than those of companies with lower debt levels based on their debt-to-equity ratios.

Return on Equity Five-Year Average (Adjusted for Debt): The average five year ROE of companies with higher debt levels are considered lower than those of companies with lower debt levels based on their debt-to-equity ratios.

Return on Assets: Net income plus interest and taxes as a percent of average total assets.

Return on Capital: The sum of net income, interest expense and minority interest, as a percent of average total invested capital which is inclusive of long-term debt, preferred stock, common equity, and minority interest.

Beta: A measure of non-diversifiable risk. It is calculated using regression Strategy incorporating 60 months of price performance versus that of the S&P 500.

Variability of EPS: The degree of variability in quarterly EPS over the past 5 years. Stocks are ranked from 10 to 1 with 10 being the most variable.

EPS Estimate Dispersion: The coefficient of variation among I/B/E/S FY2 estimates. Presented as a decile rank.

Institutional Ownership: Those companies with the lowest proportions of float-adjusted shares held by institutional owners are considered more neglected.

Analyst Coverage: Those companies with the lowest number of analysts submitting ratings to FirstCall.

Firm Size: Month-end market value.

Altman Z score: Predicts whether or not a company is likely to enter into bankruptcy within 1-2 years.

Skewness: Returns the Pearson's moment for coefficient of skewness

Ulcer Index: Measures the depth and duration of drawdowns in prices from earlier highs

Put minus Call market: Put at the money implied volatility minus Call at the money implied volatility.

Trending

Report

The Flow Show (https://rsch.baml.com/r?q=sqSO0P-x1zETI1z8yB99Pw&e=mihail_turlakov%40sberbank-cib.ru&h=ZDBsXA)

Champagne for Stocks, Beer for Bonds

Michael Hartnett 2021-Mar-11

BofA GLOBAL RESEARCH

prd - apac - node1 - r7