So Just how value is value? (and is value the value you value…and by value I mean value not valuation value)

In 1992 Morningstar introduced its now infamous style box to the world. The idea behind the style box was no doubt a noble one. The ide was to “help investors and advisors determine the investment style of a fund.” All the advisor had to do was look and he/she would immediately know what kind of fund they were looking at and what kind of companies they were invested in. With the stock world organized so nicely and neatly, ETF’s made the advisor’s life even easier. An advisor interested in increasing exposure to S&P 500 growth stocks, can simply buy an ETF that tracks the S&P 500 Growth index and get exposure to the 282 S&P 500 stocks that are considered growth stocks. Likewise, an investor/advisor looking to increase exposure to S&P 500 value stocks can buy an ETF that tracks the S&P 500 Value index and gain exposure to 386 S&P 500 value stocks. Simple, elegant, cheap, just one minor detail to pay attention to…At the moment the S&P 500 has 504 stocks and 386 + 282 does NOT equal 504. So we checked another broad market index, the Russell 1000, and found 850 Value stocks and 445 Growth stocks but the Russell 1000 only has 1013 stocks at the moment. To be clear, this is by no mean new or enlightening information any Junior analyst knows and understands that there is some overlap between these ETFs and the underlying assumption is that it is because some companies can have both characteristics. Be that as it may, it is as advisors and fiduciaries it is our responsibility to understand to the best of our abilities what exactly it is that we are buying. Simply put, How value is value?

We approach this question using machine learning models from two different directions. In the first approach, we will use a classification algorithm to attempt to place stocks in the value or growth camp based on certain features. Our second approach uses a clustering algorithm to divide our stocks into two distinct clusters using the same features.

Machine learning – a quick intro.

Broadly speaking, the Machine learning world is divided into three sections, supervised learning, unsupervised learning, and deep learning.

Supervised learning models learn to predict target values from labelled data. In other words, we take a bunch of information we know, and line it up with known outcomes based on the same information. We then, tell the computer to “learn” the relationships that exist between the data and the known outcome. If our model is successful, we can apply it to data outside of the experiment.

Unsupervised learning models work a little differently. With unsupervised models we don’t have any labels so there is nothing to predict. So unlike the supervised models where I give my computer the problems and the answer to study from. Here I give my computer data and ask it to find relationship in that data. Unsupervised models are looking to do one of two things. Identify structure/clusters within data, or transform/ reduce data into simpler terms.

Deep Learning is the most advanced and complex form of machine learning. The primary set of models used for deep learning revolve around Artificial Neural Networks. Much like they sound, Neural networks are designed to work a lot like a human brain works. When a model receives information, it is fed to a series of “neurons”, these neurons apply some sort of transformation to the information and send the transformed information to the next set of neurons and so on until we reach an output stage. While Neural networks are the premier tools for things like facial recognition, fraud detection, visual art processing, natural language processing and others, they are less applicable for our purposes.

What goes into a Machine learning model?

Regardless of what kind of machine learning we do, the “ingredients” are generally the same.

1. An algorithm
   1. An algorithm is nothing more than a series of steps taken to solve a problem. We chose our algorithm based on the problem we are trying to solve, and the information we have.
2. Training data
   1. Think of this as a question bank for an exam. In any question bank there should be enough questions to ensure we understand the concepts, and all the answer must be available so we can check our work
3. Testing data
   1. Think of this as a practice exam for the same exam. After studying for the exam, to make sure we are prepared, we will use a practice test. The practice test should test the same exact concepts in the same exact way as the real test. If we have done a good job “studying” we expect to know at this point.
4. Evaluation data
   1. This is the big test. If we have a good model, we should do well.
5. A way to evaluate our results
   1. This could be a metric like accuracy, precision, recall, and others.
   2. Remember we are not always interested in the most accurate model. Sometimes other factors are more important. For example, If evaluating the results from a COVID – 19 test, I would rather have a model that captures 99% of True Positives at the expense of more false positives with an accuracy score of 85% vs a model that captures 90% of True positives and 90% of true negatives with an accuracy score of 90%. Why? Because the goal is to identify as many sick people as possible, and if someone has a false positive that should resolve itself in subsequent re-tests.

Model I – Gradient Boosted Decision Tree (Supervised Classification Model)

The first model we trained is called a Gradient Boosted Decision Tree Model. Gradient boosted decision trees are supervised machine learning algorithm. Decision trees work a lot like call centers do. We start with a question called a root node, and based on the answer we branch out to one of a number of next questions, and so on and so forth until we reach what’s called the leaf node – (normal people call this “the answer,” data scientists call it the label). What makes gradient boosted trees unique is what happens after the process is complete. Once the first “tree” is built the algorithm takes a look at how many answers the tree got wrong and starts building another one to improve on the first one, and so on until the model cannot improve on itself. Continuing with our Call center theme, this would be the equivalent of a call center that would call another call center if they couldn’t accurately solve all problems, which would in turn call yet another call center to assist with what it couldn’t solve until there is no improvement from call center to call center.

The second model I have selected is called the KMeans clustering algorithm. This unsupervised model aims to split unlabeled data into a prespecified number of groups. It does this constantly adjusting the area encompassing a cluster as each new item is added to the calculation. The model adjusts the area by continuously calculating points called centroids for each cluster. As these centroids change the clusters they form change as well. A good way to envision this is to envision a hurricane. The eye of the storm is the centroid, its location changes based on meteorological conditions as its location changes the area the storm effects changes (this is the cluster).

Results

Gradient Boosted Decision Tree

Our features:

Keeping in line with the industry norms, we have selected the following features to classify stocks.

* Price/Book - The price-to-book, or P/B ratio, is calculated by dividing a company's stock price by its book value per share.
* Forward price-to-earnings (forward P/E) - is a version of the ratio of [price-to-earnings](https://www.investopedia.com/terms/p/price-earningsratio.asp) (P/E) that uses forecasted earnings for the P/E calculation.
* EV/Cash Flow from Operations - Enterprise Value to Operating Cash Flow compares the total value of the company with its ability to generate cashflow from its business operations.
* GICS Sector Codes – This is a numerical representation of a stock’s Sector classification.

Feature exploration and descriptive statistics:

<INSERT BOX PLOTS HERE>

Our initial analysis of the data found significant dispersion within sectors for the valuation multiples in question. Outliers can adversely effect the outcome of even the most advanced and sophisticated models, as such we applied some transformations to reduce their impact.

Transformations applied to the data:

* Missing data was removed from the set – Not always the best solution but based on the variance of metrics was too high at all levels to consider alternative methods of handling missing data.
* Winsorization – Winsorization replaces the values above the 10th percentile (in this case) and below the 90th percentile with the observations at the 10th and 90th percentile. In other words if we had a list of numbers between 1 – 100, everything below 10 would be replaced with 10, and everything above 90 would be replaced with 90.

Our Training/Test Data:

Our training/test data consists of all the stocks in the S&P 500 Pure Value, and Pure Growth indices. We chose to use these stocks as they are the best representation of what a value/growth stock look like fundamentally. It’s important to note that we removed all real estate and financial stocks form the data set because the most of the valuation metrics used in the model are not usually applicable to these sectors. To generate our training/testing data properly, we randomly assigned all the stocks in the data set to be part of the training subset (75%) or part of the test subset (25%).

Our evaluation data:

Our evaluation data consists of all the stocks in the S&P 500. Here too we removed financials and real estate sectors to ensure that our model can evaluate stocks properly based on the metrics we chose.

Results

* Price to Book and Forward P/E were of about equal importance in the classification process, GICS sector and EV/OCF were less important.

<Insert Feature Importance>

* Our Model’s accuracy score on the training set was a 93.1%. This means, that during training the model identified about 93% of value stocks accurately based on our input features.
* Our Model’s accuracy score on the test set was 91.6%. This means that the model successfully identified value stocks 91.6% of the time based on input features. This similarity in accuracy scores is also indicative of a properly fitted model.
* Our Model’s accuracy score on the evaluation data was 52.03%. This is a significant drop in model accuracy from the training and testing scores.

<Insert Accuracy Plots>

Interpretation of results

* Our model is great for identifying stocks that are CLEARLY value or growth stocks as the case in the pure value and growth portfolios. Our model failed to perform where the delineation between value and growth was less obvious.

<Insert Prediction Plots>

* Our model had a particularly difficult time with growth classifications. When classified a stock as growth, it was wrong 46% of the time, whereas when a model predicted a stock was value, it was only wrong 2.1% of the time. This could indicate significant disparities between features within the value stocks as they are classified by the industry.

Given the results of the model above, other factors not captured by valuation have gone into the construction of value and growth indices. Furthermore, the model makes clear that value vs growth is not always as cut and dry as it feels, there is plenty of grey. Some potential improvements to our supervised model include.

* Reverse the process – instead of using valuation factors to classify stocks into value and growth (more accurately “not value”), Use growth metrics for the classification process.
* Identify different features – The list of multiples used to build the model is far from exhaustive. Including other features may result in a stronger model.
* Time – cross sectional data (lots of data but all from one point in time) can be funny sometimes particularly in the financial markets. We may have better success with average/median multiples over time for each company.
* Relax the assumptions value assumptions – Our training/testing data consists of companies that are for the most part clearly value or growth stocks. By using this type of training/testing set, we may be constricting our model too much.

The