Corporate Valuation using Machine Learning

Stanford University CS229 Project Report

Aditya Mittal, Simon Ejsing, Javier Mares Romero

Project Overview

Terabytes of information on more than 47 million companies around the world are available for analysis. Wharton Business School's WRDS software makes it trivial to obtain arbitrarily large datasets of the most up-to-date information on any aspect of those companies.

Quoting Google's Peter Norvig on his talk on "Theorizing from data: Avoiding the capital mistake"

"Rather than argue about whether this algorithm is better than that algorithm, all you have to do is get ten times more training data. And now all of a sudden, the worst algorithm ... is performing better than the best algorithm on less training data."

The super-exponential increase in price-performance of information technologies is enabling fast and cheap analysis of vast amounts of data. This can bring about suggests that qualitatively deeper insight is feasible. It is fair to assume that the most advanced insight to be drawn from these data will be produced by a process combining machine and human intelligence. In the case of quantitative analysis, computer clusters can observe millions of market transactions.
A good example of this synergy is the FatKat hedge fund which incorporates the insights and style from the world's number one hedge fund manager James Simons.

This project aims to find subtle, persistent patterns in successful companies' attributes using machine learning algorithms. Stanford has access to a long list of business databases through WRDS. We used Amadeus, a pan-European database containing information on over 9 million private and public companies. The type of data used includes yearly or quarterly balance sheets, income statements, cash-flow statements… in other words, any numerical attribute of a company.

Data Acquisition

The concept is to aggregate the information from all databases on each company into a matrix, the rows corresponding to a single company and the columns to each attribute. Unfortunately, we have found that our data sources are sparse in the sense that a lot of data values are reported as not available. Therefore, a very small dataset of 482 companies was at first used to avoid handling missing values. This dataset was generated by prune ever sample with missing entries from the training dataset. However, this method of dealing with missing values caused too much data loss.

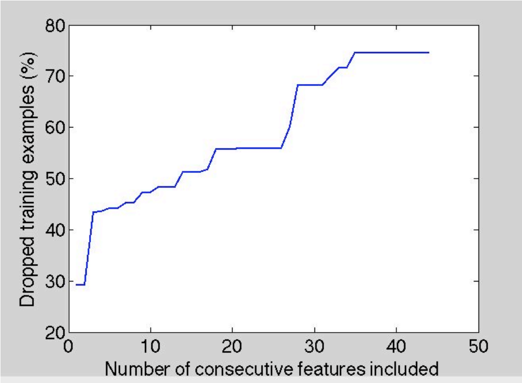

Figure 1-Data Loss

Therefore, it seems intuitive to try to recover some of the lost data. In our first attempt we insert zero values for every missing value, regardless of which feature they correspond to. This approach has the obvious drawback of introducing erroneous data points into the dataset, but we have found that the gain in valid data more than up weighs the introduced errors. Anyway, the testing process should reveal whether or not this approach is better than to prune examples with missing data.

An even better approach is to try and estimate the true value of the missing data points. This seeks to reduce the introduced errors. A simple approach is to set the missing data to be the observed average of that particular feature, while a more elaborate approach would be to try to approximate the distribution on the feature using a normal distribution and then randomly draw the missing values from this distribution. However, we stress that whichever labeling is used, these cannot be recovered in this manor, as it would lead to miscalculations of the test error percentage.

Visualizing the Data

Data is visualized early on to get an idea of the relationship between trends and to find any spurious data. Here is an example:

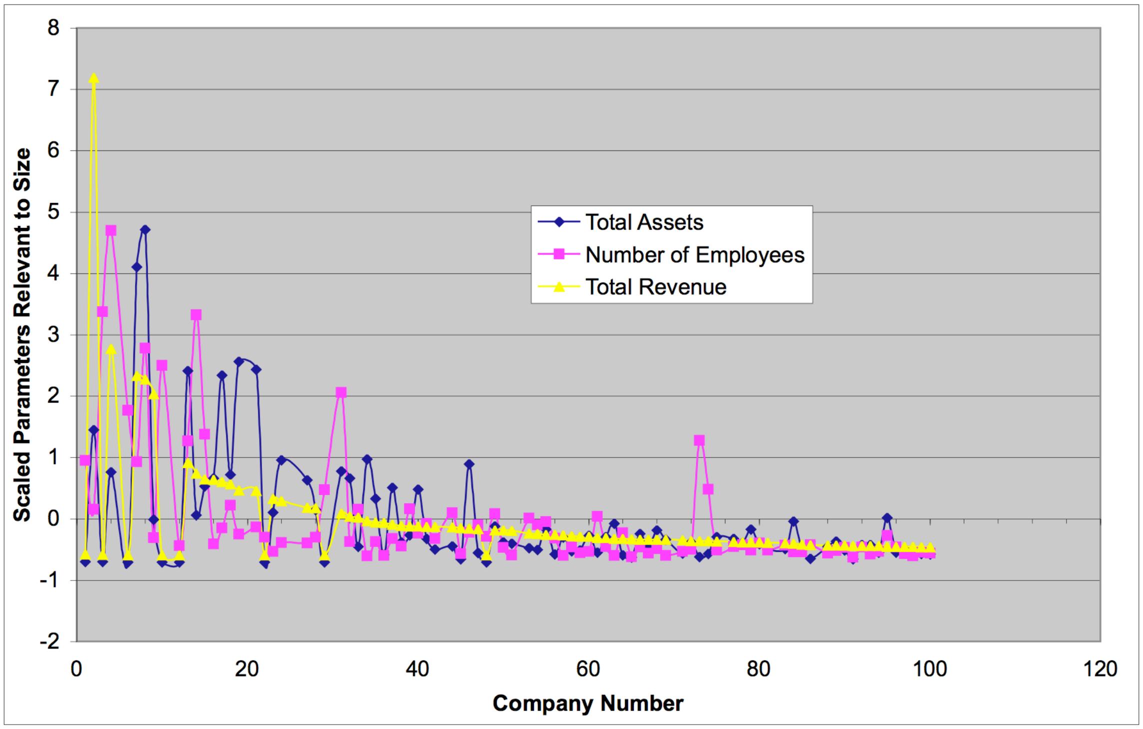

Figure 2 – Correlation between features representative of the size of a company

In finance when making a decision based upon the size of a company, decision makers use three different aspects. Some use Total Assets, others use Number of Employees, and yet others use the Total Operating Revenue of a company to make this decision. We expect that the three features should be highly correlated and we represent our expectation in the graph above and observe that it is the case that the three features are highly correlated. This leads us to a more general realization about the nature of the features in our databases that many of the features are highly correlated. This inspires the removal of features that are highly correlated from the feature set derived directly from the database.

Extracting the Principal Components

Inspired by the above observation, in this part of the project we apply principal component analysis (PCA) to extract the feature vectors which are necessary to evaluating the valuation of a corporation. This reduces the dimensionality and hence the complexity of the problem. Before applying the PCA we preprocess the data to have zero mean and unit variance. We also use only features for which data exists for most companies and not the features which we have data only for very few of the companies in the data set. This way we end up considering only orthogonal components of the data and eliminating features that can be represented as linear combinations of other features. In our work for example, this algorithm successfully reduces datasets of 73 features to just about 20 features with only 5% relative error.

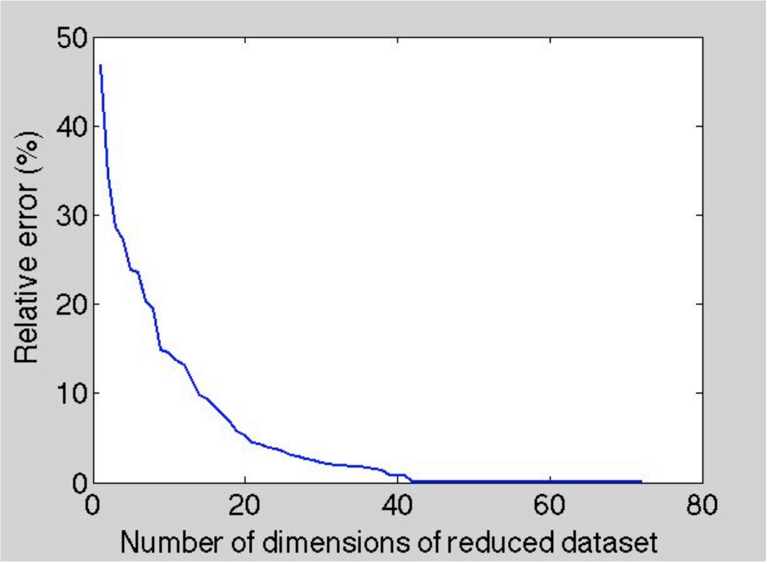

Figure 3 – Applying Principal Component Analysis to Financial Datasets

We predict that in financial data there is a high reduction of dimensionality by using PCA because many of the financial features are ratios and sums of other features.

Labeling the dataset

The labeling of the dataset defines our merit of performance of a company. In our project we use each company's "stock turnover" as the merit function. In particular, we are interested in whether a given company is performing better than the market. We approximate the market performance as the mean of the stock turnover for all companies in the dataset. If a company performs above average we label it with +1, otherwise -1. Labeling of the companies is done based on "stock turnover" from 2005, while all features are extracted annually from 2001 – 2003. The extracted features do not include data on "stock turnover."

Logistic Regression

After computing the principle components we use Logistic Regression learning algorithm to classify binary data. The purpose of this is to be able to apply this machine learning algorithm and classify binary data. In our code we have created a function which can take any vector of data and convert it to binary data around either the mean of the data vector or a set threshold value. The binary classification using logistic regression works pretty well when we apply it to vectors within the data. However, the classification fails when we apply it to the real test data from of stock turnover values from WRDS. This is because this dataset is very complex and closely related values occur even in higher dimensions and the algorithm is unable to find a hyper plane that can separate this data.

SVM and Cross Validation

After realizing the failure of logistic regression on our training dataset, we look into other algorithms. Our choice fell on the SVM implementation because we are able to adjust the C parameter to allow the separation boundary to be imperfect, thus circumventing the problem we encountered with logistic regression. We picked up from the SMO implementation we did in homework 2 and adjusted some minor details.

We were first able to successfully train the SVM on our small sample dataset of 382 companies. During our training we used a *tolerance* value of 0.001, a *C* of 1.0 and *max_passes* of 20. Having completed the training procedure we performed a test on 100 companies that was left out of the training set. We carried out this test in a k-fold leave out manor, and were able to obtain at best 33% test error, while at worst 52% error. Given the vague testing basis, this can easily be attributed to chance.

		Max	**Min Error**	**Max Error**
C	**Tolerance**	**Passes**	**(%)**	**(%)**
0.5	0.001	20	10%	45%
1	0.001	20	10%	30%
1	0.0001	20	10%	36%
2	0.001	20	11%	87%
1	0.0001	30	16%	67%
1	0.001	40	10%	29%

Hereafter we trained on the larger, but contaminated dataset of 630 companies using a variety of different SVM parameters. Table 1 lists the outcome of the k-fold cross validation for each set of SVM parameters along with the minimum and maximum test error. It is evident from the table that a large portion of parameters results in inconsistent behavior, which indicates that the training process did not converge. However, in our best case we were able to obtain a test error range of 10% – 29%, which is a much more accurate classification of the test data than the previous approach. We can thus conclude that our approach of introducing erroneous data points into the dataset to recover more data increases the performance of the SVM.

Results

We have promising results but at this moment we are still in the process of testing our results. All of these promising results are from testing on the same dataset. We are now testing on an even larger (1400 companies) and new dataset completely independent dataset from the same time frame to see how well the system performs. We are not sure we fully understand why our model fully works but so far it seems to converge and perform well. So far the k-fold errors have been 15%, 19%, 12%, and 21% which is pretty good.

References

[1] K. Pelckmans, J. De Brabanter, J.A.K. Suykens, B. De Moor. (2005). *Handling missing values in support vector machine classifiers*. Neural Networks 18. www.elsevier.com/locate/neunet

JSTOR Articles:

[2] "The Cross-Section of Expected Stock Returns", Eugene F. Fama; Kenneth R. French, The Journal of Finance, Vol. 47, No. 2. (Jun., 1992), pp. 427-465

[3] "The Conditional Relation between Beta and Returns", Glenn N. Pettengill; Sridhar Sundaram; Ike Mathur, The Journal of Financial and Quantitative Analysis, Vol. 30, No. 1. (Mar., 1995), pp. 101-116

[4] "Stock Returns, Expected Returns, and Real Activity", Eugene F. Fama, The Journal of Finance, Vol. 45, No. 4. (Sep., 1990), pp. 1089-1108

[5] "The Distinction between Merit and Worth in Evaluation", Yvonna S. Lincoln; Egon G. Guba, Educational Evaluation and Policy Analysis, Vol. 2, No. 4. (Jul. - Aug., 1980), pp. 61-71.

[6] "The Conditional Relation between Beta and Returns", Glenn N. Pettengill; Sridhar Sundaram; Ike Mathur, The Journal of Financial and Quantitative Analysis, Vol. 30, No. 1. (Mar., 1995), pp. 101-116.