**Predicting Industry Classification from Financial Reporting Data**

**Introduction:**

The purpose of my project was to determine if I could successfully predict the industry of a company based on financial reporting data from 2011 to 2015.

**Data Sources:**

The financial data was obtained from an API hosted by Kimono Labs at the following link: <http://www.kimonolabs.com/sec/docs#FinancialsObject>. It provided access to data reported to the Securities Exchange Commission on form 10-K, which is a required annual report to the SEC on a broad array of financial indicators and is required of all companies with more than $10 million in assets and a class of equity securities that is held by more than 500 owners, regardless of whether the company is publicly traded or not.

For my initial dataset I pulled data form 2011 to 2015 for 484 companies currently on the S&P 500. For my testing dataset I pulled data on 231 other companies on the Russell 1000 Index that were not already in my list. The code that pulled data from the API can be found here: <https://github.com/wbconnelly/Bill_Data_Sci/blob/master/project%20draft/kimono_api.py> .

For industry classification I used a web scraper that pulled the Industry classification as listed on Google Finance (google.com/finance) under the company’s stock symbol. The code for the web scraper can be found at the following link: <https://github.com/wbconnelly/Bill_Data_Sci/blob/master/project%20draft/kimono_api.py>.

**Methodology:**

**Data formation:**

My initial step was to combine the data frame containing the financial reporting data with the data frame containing the sector classification of each company. Fortunately I was able to pull each dataset using the company stock symbol which created a convenient common element in each data frame. I combined them using the merge() function in python’s pandas library. After this step I had a data frame which contained both the financial reporting data and the industry classification. I called this data frame company\_data with the column Sector denoting the industry classification.

I attempted to eliminate the distortions that trends in the data would introduce. My thought was that if a company experienced significant growth in a factor such as revenue then at different points in time the model might recognize it as a different type of company. To accomplish this I grouped the data by the field company\_symbol and calculated the mean for each column. This resulted in a data frame with one row for each company and the value in that in each column for that row being the average value for that company over the five year period. I labeled this data frame company\_avg. The table below lists the sectors in the data set along with the number of companies belonging to each.

|  |  |
| --- | --- |
| Financials | 85 |
| CyclicalConsumerGoods&Services | 82 |
| Industrials | 68 |
| Technology | 61 |
| Healthcare | 42 |
| Non-CyclicalConsumerGoods&Services | 38 |
| Energy | 36 |
| Utilities | 27 |
| Basic Materials | 23 |
| None Found | 17 |
| Telecommunications Services | 5 |

Companies labeled with “None Found” are companies for which no industry classification could be found on google.com/finance.

**Dealing with missing values:**

To deal with missing values I first examined the data set before computing the average for each column by obtaining a count of missing values for each column. Columns for which there were over 300 missing values, meaning that over 300 of the companies in of the 484 had no data in this columns, were eliminated. This led to the elimination of accountingchange, deferredcharges, equityearnings, extraordinaryitems, otherequity, and researchdevelopmentexpense from the dataset. Other values such as 'amended', 'audited', 'Unnamed: 0\_y', 'preliminary', 'Unnamed: 0\_x', 'year', 'quarter', 'restated', 'company\_cik', 'usdconversionrate', 'periodlength', 'original', 'crosscalculated', 'discontinuedoperations ‘ that were merely report elements not related to financial data were eliminated as well.

As a final measure I attempted a Principal Components Analysis to identify features that did not contribute meaningfully to the total variance in the dataset. After performing the analysis and constructing the correlation matrix between the principal components and the variable in the dataset I was able to eliminate 'changeininventories', and 'inventoriesnet'. The final number of features in the dataset was 46. They are listed below:

|  |
| --- |
| 'capitalexpenditures', |
| 'cashandcashequivalents', |
| 'cashcashequivalentsandshortterminvestments', |
| 'cashfromfinancingactivities', |
| 'cashfrominvestingactivities', |
| 'cashfromoperatingactivities', |
| 'cfdepreciationamortization', |
| 'changeinaccountsreceivable', |
| 'changeincurrentassets', |
| 'changeincurrentliabilities', |
| 'commonstock', |
| 'costofrevenue', |
| 'dividendspaid', |
| 'ebit', |
| 'effectofexchangerateoncash', |
| 'goodwill', |
| 'grossprofit', |
| 'incomebeforetaxes', |
| 'intangibleassets', |
| 'interestexpense', |
| 'investmentchangesnet', |
| 'minorityinterest', |
| 'netchangeincash', |
| 'netincome', |
| 'netincomeapplicabletocommon', |
| 'otherassets', |
| 'othercurrentassets', |
| 'othercurrentliabilities', |
| 'otherequity', |
| 'otherliabilities', |
| 'preferredstock', |
| 'propertyplantequipmentnet', |
| 'researchdevelopmentexpense', |
| 'retainedearnings', |
| 'sellinggeneraladministrativeexpenses', |
| 'totaladjustments', |
| 'totalassets', |
| 'totalcurrentassets', |
| 'totalcurrentliabilities', |
| 'totalliabilities', |
| 'totallongtermdebt', |
| 'totalreceivablesnet', |
| 'totalrevenue', |
| 'totalshorttermdebt', |
| 'totalstockholdersequity', |
| 'treasurystock' |

I then used a loop to segment the data by Sector and I imputed the mean for each sector to that column. As a final measure I used the scaler.fit\_transform() function from Python’s sklearn library to center and scale the data by subtracting the column mean from each value and then dividing all values by the column standard deviation.

**Models Used:**

I primarily relied on Logistic Regression from the sklearn library for this project.

To classify companies by Sector in this project I needed to obtain target values for each category in the Sector column. To do this I used the get\_dummies() function, which produced one column for each industry (1 total) with a 1 indicating that the Sector value for that row was the column title.

Additionally to obtain a proxy metric for the importance of each feature by sector I created a data frame for each sector that calculated the variance in each column, sorted it in descending order and calculated how much of the cumulative variance was accounted for as each column was added to the feature set. This allowed me to control how much of the variance in the data I wanted to include in my model by setting thresholds in the loop that created logistic regression models on each sector based subset of the data. Setting the cumulative variance threshold very high (.99999999999999 for example) would include all the features, while progressively lowering it would tend to eliminate features. The code for creating these tables of the variance in each column is below:

var\_list = {}

for sector in sector\_list:

var\_tbl = pd.DataFrame(x[feature\_cols][x.Sector == sector].var().reset\_index())

var\_tbl.rename(columns = {0:'variance', 'index':'col\_title'}, inplace = True)

var\_tbl.sort('variance', ascending = False, inplace = True)

# calculate the cumulative variance accounted for by the features to

# this point in the table and divide by the total variance

var\_tbl['total\_var']= var\_tbl['variance'].cumsum(skipna = True)/var\_tbl.variance.sum()

var\_list[sector] = var\_tbl

In my first attempt I set the variance threshold very high to ensure all 46 features were used and observed the following results. I used a loop that looped through all sector names proceeded according to the following steps:

1. Select the features based on the variance threshold
2. Subset the scaled financial data to include only the selected features
3. Subset the data frame of dummy values to include only the column corresponding to the target Sector
4. Fit a logistic regression model unique to that sector.
5. Report the True Positives, True Negatives, False Positives, and False Negatives for each sector.

I obtained the following results

|  |
| --- |
| **None Found --- Accuracy 0.993801652893 Number of features 46** |
| True Positives: 15 |
| True Negatives: 466 |
| False Positives: 1 |
| False Negatives: 2 |
| **Basic Materials --- Accuracy 1.0 Number of features 46** |
| True Positives: 23 |
| True Negatives: 461 |
| False Positives: 0 |
| False Negatives: 0 |
| **Cyclical Consumer Goods & Services --- Accuracy 0.894628099174 Number of features 46** |
| True Positives: 47 |
| True Negatives: 386 |
| False Positives: 16 |
| False Negatives: 35 |
| **Technology --- Accuracy 0.95041322314 Number of features 46** |
| True Positives: 47 |
| True Negatives: 413 |
| False Positives: 10 |
| False Negatives: 14 |
| **Healthcare --- Accuracy 0.95867768595 Number of features 46** |
| True Positives: 27 |
| True Negatives: 437 |
| False Positives: 5 |
| False Negatives: 15 |
| **Non-Cyclical Consumer Goods & Services --- Accuracy 1.0 Number of features 46** |
| True Positives: 38 |
| True Negatives: 446 |
| False Positives: 0 |
| False Negatives: 0 |
| **Financials --- Accuracy 1.0 Number of features 46** |
| True Positives: 85 |
| True Negatives: 399 |
| False Positives: 0 |
| False Negatives: 0 |
| **Industrials --- Accuracy 0.981404958678 Number of features 46** |
| True Positives: 64 |
| True Negatives: 411 |
| False Positives: 5 |
| False Negatives: 4 |
| **Utilities --- Accuracy 1.0 Number of features 46** |
| True Positives: 27 |
| True Negatives: 457 |
| False Positives: 0 |
| False Negatives: 0 |
| **Energy --- Accuracy 1.0 Number of features 46** |
| True Positives: 36 |
| True Negatives: 448 |
| False Positives: 0 |
| False Negatives: 0 |
| **Telecommunications Services --- Accuracy 1.0 Number of features 46** |
| True Positives: 5 |
| True Negatives: 479 |
| False Positives: 0 |
| False Negatives: 0 |

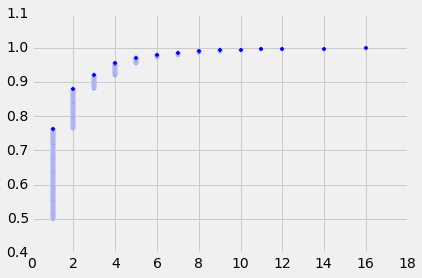
Using logistic regression on the training data I was able to correctly identify the sector with 100% accuracy for the sectors Telecommunications Services, Energy, Utilities, Financials, Non-Cyclical Consumer Goods & Services, and Basic Materials.

To examine the effect on increasing the number of features, and consequently the amount of total variance in the data set created a loop which recorded the number of false predictions as the variance threshold level increased. To do this is nested the previous loop in another loop which incremented the variance threshold each time starting at .5 and ending at one. I then plotted the number of false predictions, and total model accuracy for each sector against the increasing variance threshold and the number of features. Examples of the graphs produced are below.

For a sector such as financials where the model reached 100% accuracy we can see a clear and immediately positive relationship between adding high variance features and improving the model.

X axis = number of features

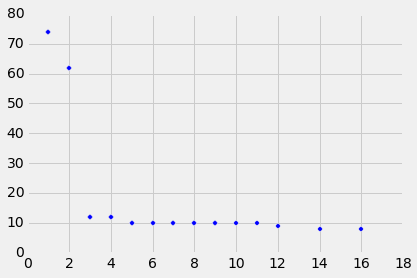
Y axis = total cumulative variance



Here we see that for the Financials sector almost all the variance in explained in fewer than 10 features.

X axis = number of features

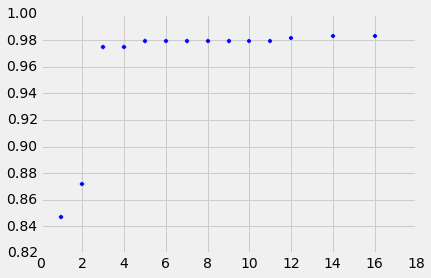
Y axis = total false predictions



Here we see that the total number of false predictions drops sharply after the first two features are added.

x axis = number of features

y = accuracy score

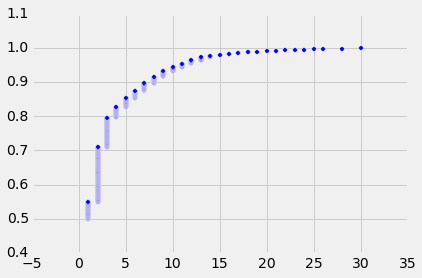


And here we can observe that the total model accuracy sharply rises after only two features.

For a sector such as Cyclical Consumer Goods & Services which had a significant number of False Positives and negatives the graphs look somewhat different and the relationship is more delayed and variable.

x axis = number of features

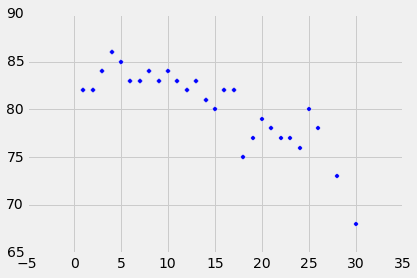
y axis = total cumulative variance



Here we see that the total variance accounted for by the features only approaches 100% after 25 features.

x axis = number of features

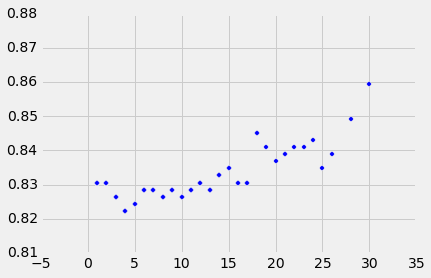
y axis = total false predictions



Here is can be observed that the total number of false predictions remains quite high at slightly less than 70 after the number of features reaches more than 30.

x axis = number of features

y = accuracy score



And here we can see that as the number of features approaches 30 the accuracy of the model only reaches approximately 86%.

This pattern is repeated for other sectors where the sectors for which the models were very accurate accounted for a high degree of variance in relatively few features, and relatively few features were able to account for almost all the variance. This confirmed to me that variance in the feature was a good proxy metric for how distinct they were and suitable for use in making predictions.

**Testing the models**

To test the model I downloaded financial reporting data from 2011 to 2015 for 230 additional companies from the Russell 1000 index of the 1000 largest publicly traded companies by market cap, which were not in my previous list. I also downloaded their industry classifications and merged the two data frames as before. To eliminate the influence of trends I centered and scaled the data as before and took the mean value over the five year period for each company and then centered and scaled the data to eliminate possible distortions caused by differences in scale.

To ensure the correct model was used for each sector in the testing data I created a dictionary and within the original for-loop I created a key -value pair which saved the fitted model in the dictionary as the value with the sector name as it’s key.

The initial results of model testing were not encouraging. The output from the initial testing is below:

|  |
| --- |
| **None Found --- 0.896103896104 number of features 46** |
| True Positives: 2 |
| True Negatives: 205 |
| False Positives: 9 |
| False Negatives: 15 |
| **Basic Materials --- 0.891774891775 number of features 46** |
| True Positives: 0 |
| True Negatives: 206 |
| False Positives: 11 |
| False Negatives: 14 |
| **Cyclical Consumer Goods & Services --- 0.848484848485 number of features 46** |
| True Positives: 2 |
| True Negatives: 194 |
| False Positives: 9 |
| False Negatives: 26 |
| **Technology --- 0.848484848485 number of features 46** |
| True Positives: 0 |
| True Negatives: 196 |
| False Positives: 11 |
| False Negatives: 24 |
| **Healthcare --- 0.861471861472 number of features 46** |
| True Positives: 2 |
| True Negatives: 197 |
| False Positives: 9 |
| False Negatives: 23 |
| **Non-Cyclical Consumer Goods & Services --- 0.91341991342 number of features 46** |
| True Positives: 0 |
| True Negatives: 211 |
| False Positives: 11 |
| False Negatives: 9 |
| **Financials --- 0.714285714286 number of features 46** |
| True Positives: 4 |
| True Negatives: 161 |
| False Positives: 7 |
| False Negatives: 59 |
| **Industrials --- 0.831168831169 number of features 46** |
| True Positives: 0 |
| True Negatives: 192 |
| False Positives: 11 |
| False Negatives: 28 |
| **Utilities --- 0.922077922078 number of features 46** |
| True Positives: 0 |
| True Negatives: 213 |
| False Positives: 11 |
| False Negatives: 7 |
| **Energy --- 0.891774891775 number of features 46** |
| True Positives: 1 |
| True Negatives: 205 |
| False Positives: 10 |
| False Negatives: 15 |

Only Cyclical Consumer Goods & Services, Healthcare, Financials, and Energy were sectors where the model was able to identify the sector correctly. All others had significant numbers of False Positives and False Negatives. In an attempt to produce better results I tried ensembling (number of iterations = 1000), but this produced no better results.

Adjusting the regularization parameter of the Logistic regression object also produced no better results, however I did note that as the regularization parameter increased the Financials industry was the first where the model was able to record a True Positive.

I also tried using a decision tree model but even at very high depth levels, the models were able to accurately fit the training data but broke down when trying to make predictions for the testing data.

**Conclusion:**

I attempted to build the models by using variance as an indicator of feature importance because with 74 original columns and nearly 2000 rows covering 484 companies and 11 sectors in the original dataset, exploring features manually to look for distinct ones in each sector, seemed prohibitive. While it was able to produce a model that was very accurate at learning the training data, I was not able to produce one that produced accurate results for out of sample data. Efforts at standardizing the data did little to improve it’s applicability.

I think a possibly beneficial way of pursuing this topic in the future would be to obtain more complete data that captured aspects of the industries that are truly unique. For instance I knew from experience that Energy companies typically pay more in dividends and when exploring the data I found that indeed the Energy companies in my dataset paid, on average, the most in dividends. Using only this feature I was able to correctly identify more energy companies than in any other sector. Similarly I also knew that research and development expenses were typically higher for technology and healthcare companies and data exploration confirmed this, and again as with Energy companies I was able to correctly identify companies in these industries. But using only these features was not as accurate as using many more features, even on the training data. Ultimately limits of time prohibited my exploring all the variable in the data to fin combinations that were truly unique to the industries, and I was forced to rely on a metric such as variance which I was able to use to automate feature selection.