Master of Quantitative Finance

QF624 Machine Learning and Financial Application

Project Report

Bankruptcy Detection Using Machine Learning

Bai Chen

Wang Yunxiang

Wen Chongji

Zhang Penghui

Table of Contents

List of Figures

List of Tables

1. Introduction

The 2007/2008 financial crisis has made credit risk management a priority. Hence, the likelihood of bankruptcy is of paramount importance to financial institutions, fund managers, lenders, governments, and other financial market players.

There has been intensive research from academics and practitioners regarding the models and characteristics. Ohlson (1980) was one of the first researches to apply logistic regression to default estimation [1]. Ohlson's approach was followed by several other researches due to the ease of running logistic regression, but the inaccuracy of the popular model was also pointed out by Begley [2]. Furthermore, the lack of a reliable theoretical framework within the field of corporate finance to examine bankruptcy remains a problem.

In recent years, more researches tend to explore other tools with advances in computer technology, especially with the development of artificial intelligence and machine learning tools. Since bankruptcy analysis is similar to pattern-recognition problems, it is promising to develop algorithms to classify and discriminate the possibility of bankruptcy. However, Wang et al [3] found that the model performance depended highly on the specific characteristics of the features adopted in classification and on the data structure. Therefore, the flexibility of model construction in the prediction of bankruptcy remains an interesting area to be explored in this project.

The objective of this project is to use machine learning to conduct bankruptcy detection. Three single supervised models (Logistic regression, Decision Tree, and SVM) and three types of ensembled model (Random Forest, AdaBoost, and XGBoosting) are explored and the model performance will be compared.

2. Data Description and Processing

2.1. Data Description

The dataset we collected is about bankruptcy prediction of Polish companies. The data was collected from Emerging Markets Information Service (EMIS), which is a database containing information on emerging markets around the world. The bankrupt companies were analyzed in the period 2000-2012, while the still operating companies were evaluated from 2007 to 2013. The data contains financial rates from the 5th year of forecasting period and corresponding class label that indicates bankruptcy status after 1 year. There are totally 5910 instances

(financial statements), 410 represents bankrupted companies, 5500 firms that did not bankrupt in the forecasting period. The financial statements contain 64 attributes as listed in Appendix-A.

2.2. Data Processing

Missing values is the first problem we met with the data. According to the column graph, some companies miss attributes like X37 (current assets – inventories)/ long-term liabilities, X27 profit on operating activities /financial expenses, X45 net profit/ inventory, and etc. In practice, the different industries have their unique attributes, causing the missing values in data included all emerging market companies. The K-Nearest Neighbours (KNN) algorithm is applied to fill the empty value by estimating values of the closest points, based on other variables.

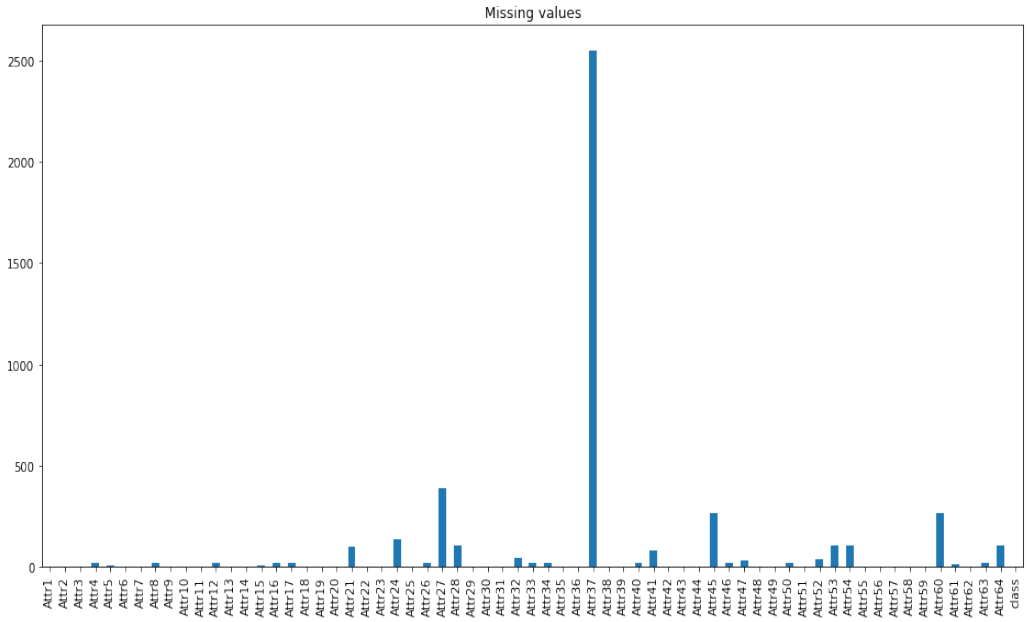

Figure 2-1 Missing Values in Raw Data

Imbalance data is the most common problem in machine learning. In our data, there are 410 represents bankrupted companies, 5500 firms that did not bankrupt in the forecasting period. The ratio of the two type instances is 7:93. Since the purpose is to the detective the minority

There are different approaches to solve imbalance data such as resampling, random under-sampling, informed oversampling and etc. We applied the SMOTE algorithm to our project, which synthesizes new minority instances between existing minority instances.

3. Model Comparison

3.1. Target Indicator: F-2 Score

Before going through data modeling, we would like to emphasize on the Precision and Recall. In our case study, precision means "Of our predicted bankrupt companies, how many are actual bankrupt ones", while recall tells us " Of actual bankrupt companies, how many we can predict correctly". From the investors' perspective, low precision can cause investors to miss some good companies whereas low recall can lead investors to invest in potential bankrupt companies. For this reason, we want to focus more on Recall rather than Precision. However, we do not want to ignore precision completely. What we would like to do is to add more weight on recall score and less weight on precision. Therefore, F2 score is used in our model evaluation. Here is the equation of f2 score when beta equals to 2:

$$F_\beta = \frac{1}{\left(\frac{1}{\beta^2+1}\frac{1}{P} + \frac{\beta^2}{\beta^2+1}\frac{1}{R}\right)}$$

, where P is the precision and R is the recall.

3.2. Classification Models

Some classical models are used in our classification. Here are the classification models: Logistic regression, Decision tree, SVM, Random Forest, Adaptive Boost, and XGBoosting. We used "Stratified KFold cross-validation " with five splits and shuffling operation.

```
skf=StratifiedKFold(n_splits=5,shuffle=True,random_state=2019)
```

With the help of "Pipeline" function, a series of operations were incorporated in the pipeline. An example of a pipeline is shown below:

```
pipe1=make_pipline_imb(scaler,SMOTE(random_state=2019),LogisticRegression(solver='lbfgs',random_state=2019
))
```

We firstly scaled the data, then did an oversampling using SMOTE, followed by a classification model. Table 3-1 and Figure 3-1 summarize the different scores for each model.

Table 3-1 Model Performance Using Default Parameters

	Accuracy	Precision	Recall	F1 Score	F2 Score
LogisticReg	80.80%	22.00%	69.27%	33.37%	48.41%
DecisionTree	85.55%	27.93%	63.90%	38.40%	50.19%
SVM	81.32%	22.20%	67.32%	33.38%	47.84%
RandomForest	91.69%	41.99%	48.78%	44.87%	47.06%
AdaBoost	86.04%	27.89%	63.66%	38.73%	50.58%
XGBoosting	89.53%	36.57%	68.54%	47.58%	58.21%

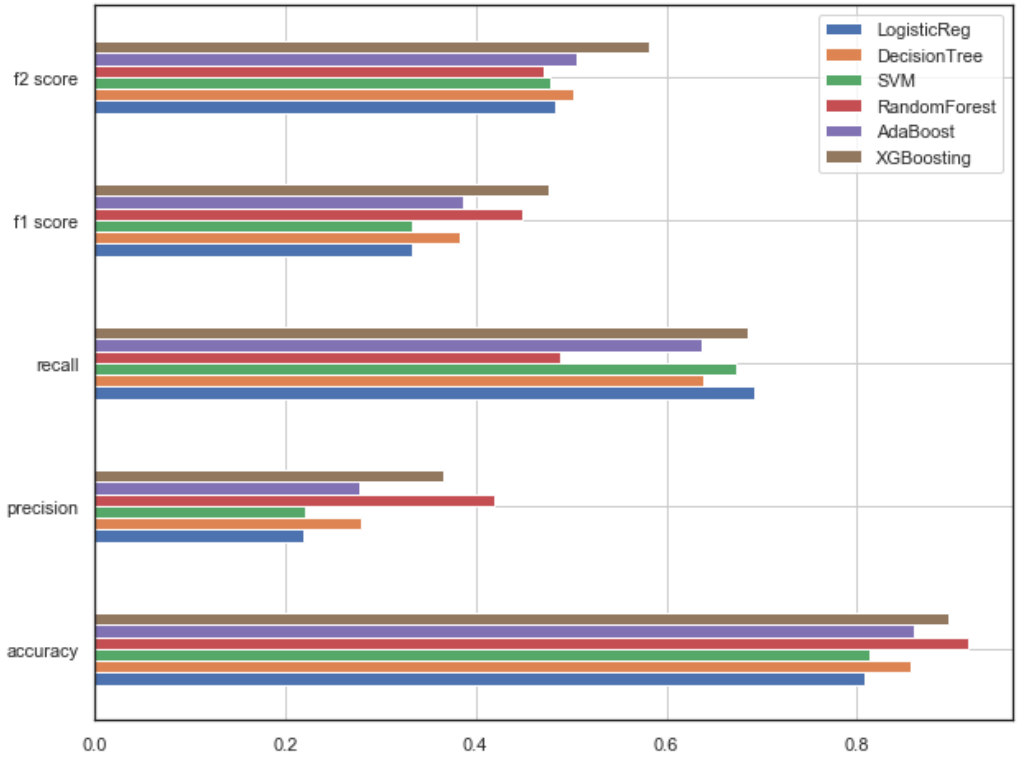

Figure 3-1 Model Performance Using Default Parameters

As mentioned earlier, F2 score is a more import score than others. Before optimization, XGBoosting outperformed others in terms of F2.

3.3. Optimization

In the optimization process, we noticed that there is a trade-off between precision and recall scores. Figure 3-2 illustrates the trade-off effect of logistic regression as an example.

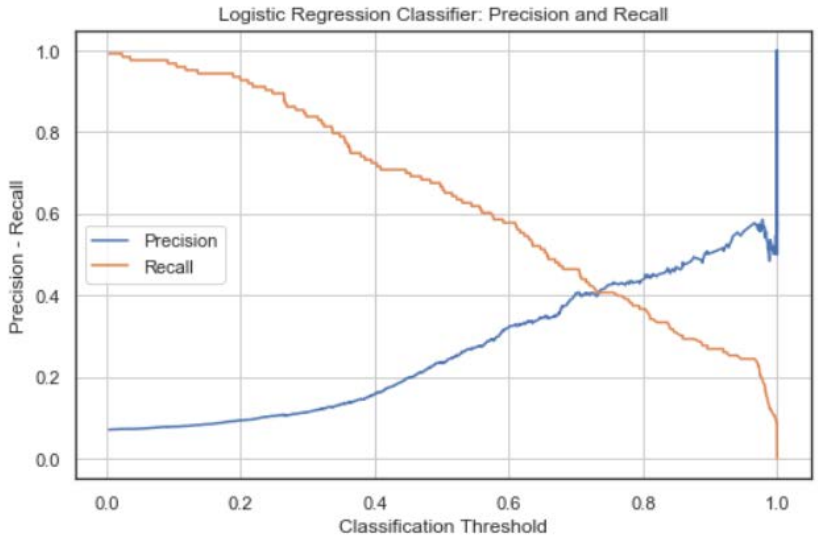

Figure 3-2 Logistic Regression Precision-Recall Trade-Off by Adjusting Threshold（Default 0.5）

It is observed that precision and recall move in the opposite direction when the threshold is changing. In addition, this logistic regression did not perform well because precision is high as the recall is extremely low. Our desirable model is that both precision and recall are acceptable.

All the models are optimized using GridSearchCV function. Below is how we implemented optimization, where C is regularization parameter, class_weight assigns various weights to classes, max_depth is maximum depth in decision tree and n_estimators is number of estimators used in model.

```
LogReg_param_grid = [{'classifier__C':[ 0.001, 0.01, 0.1,1,10],
                      'classifier__class_weight':[{0:0.01, 1:0.99}, {0:0.80, 1:0.20},{0:0.20, 1:0.80},{0:1, 1:20}]}]

DT_param_grid = [{'classifier__max_depth':np.arange(5,25,1),
                  'classifier__class_weight':[{0:0.01, 1:0.99}, {0:0.80, 1:0.20},{0:0.20, 1:0.80},{0:1, 1:20}]}]

SVM_param_grid=[{'classifier__C':[ 0.01, 0.1,1,10],
                 'classifier__class_weight':[{0:0.01, 1:0.99}, {0:0.80, 1:0.20},{0:0.20, 1:0.80}]}]

RF_param_grid=[{'classifier__n_estimators':[10,20,30,40,50],
                'classifier__class_weight':[{0:0.3, 1:0.7}, {0:0.80, 1:0.20},{0:0.20, 1:0.80},{0:1, 1:20}],
                'classifier__max_depth':np.arange(5,15,1)
                }]

AdaBoost_param_grid=[{'classifier__n_estimators':[10,20,30,40,50] }]

xgb_param_grid=[{'classifier__max_depth':np.arange(5,15,1),
                 'classifier__n_estimators':[20,30,40,50,70,100],
                }]
```

Please take note that we did not optimize too many parameters as GridSearchCV takes a very long time.

After optimization the scores are summarized in the table below:

Table 3-2 Performance Comparison of Optimized Models

	Accuracy	Precision	Recall	F1 Score	F2 Score
LogisticReg	39.86%	9.69%	91.95%	17.53%	34.06%
DecisionTree	90.22%	35.82%	50.24%	41.77%	46.45%
SVM	36.58%	9.05%	89.76%	16.44%	32.22%
RandomForest	91.32%	41.36%	58.54%	48.42%	54.00%
AdaBoost	85.26%	27.33%	67.32%	38.84%	52.01%
XGBoosting	92.10%	45.25%	63.41%	52.69%	58.60%

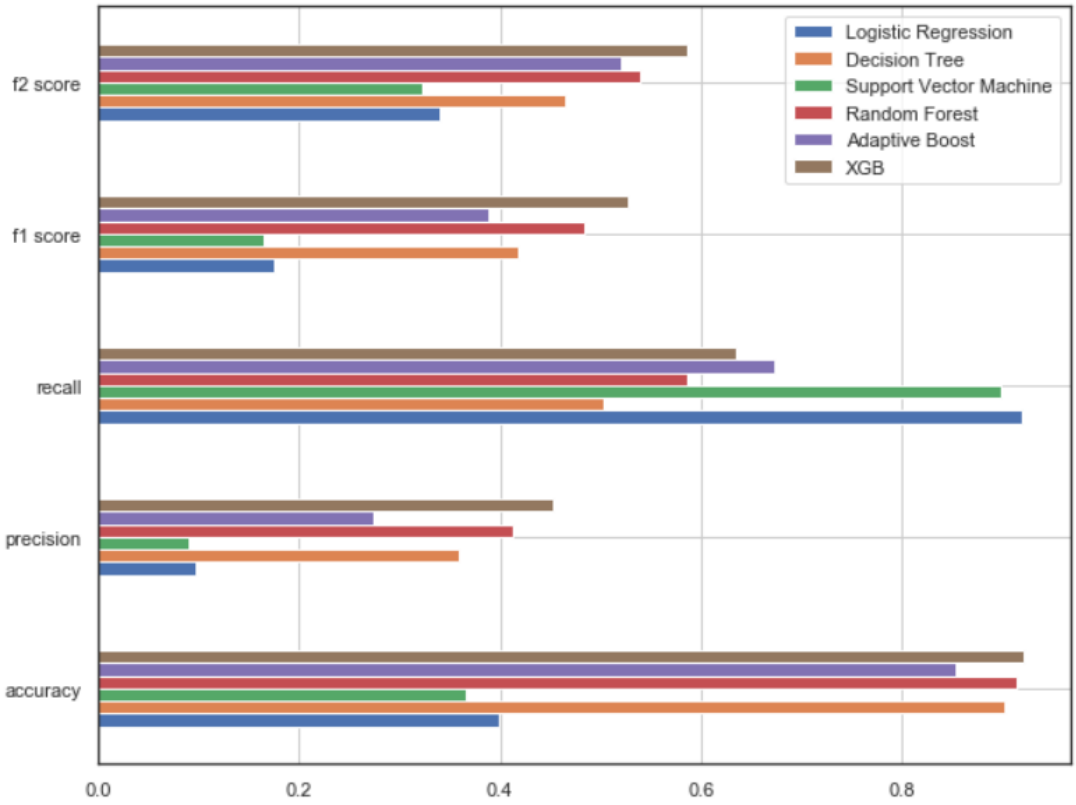

Figure 3-3 Model Performance Using Optimized Parameters

Again, XGBoosting outperformed others in terms of f2. Logistic regression and SVM models return poor scores. Furthermore, ensemble models are better than single models.

3.4. Model Comparison with ROC AUC score

ROC measures overall performance for each model. Intuitively, ROC tells us the number of mistakes made to achieve correct predictions. It is obvious that XGBoosting is the best. Rando forest and adaptive boosting rank 2nd and 3rd. Moreover, a model is bad if the ROC falls below the diagonal line. It means that even a random guess is better than the model prediction.

Figure 3-4 plots the ROC AUC Scores for each model. To quantify ROC performance, AUC score is calculated and summarized in Table 3-3 below:

Table 3-3 Summary of AUC Score

	AUC Score
LogisticReg	0.8057
DecisionTree	0.7181
SVM	0.7778
RandomForest	0.8934
AdaBoost	0.8469
XGBoosting	0.9167

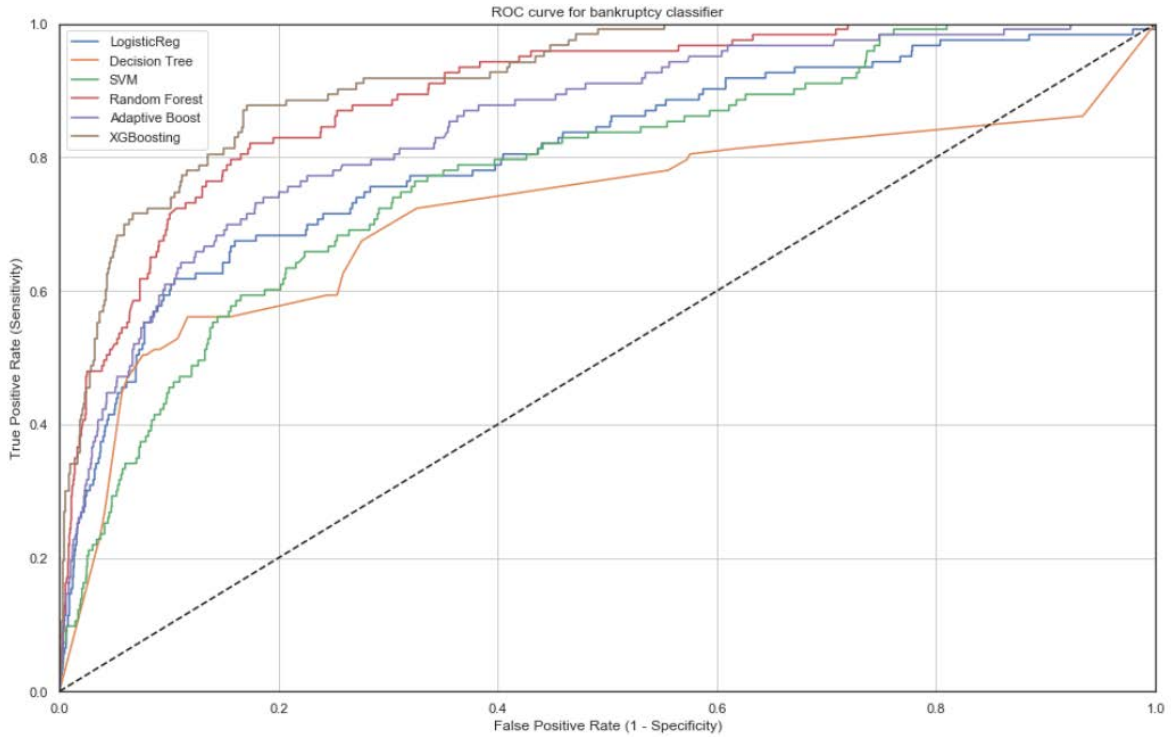

Figure 3-4 Model Comparison with ROC AUC Score

Clearly, we observe XGboosting , random forest and adaptive boost return us better result.

4. Financial Interpretation

Using the optimized XGboosting model, we have the following feature importance rank:

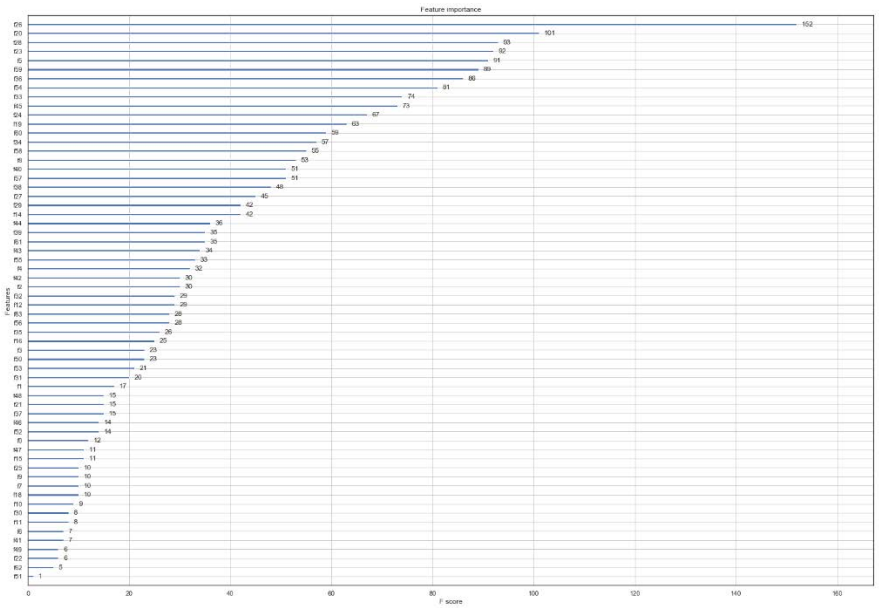

Figure 4-1 Importance Ranking of Features

As the graph shows, the top ten features are:

(net profit + depreciation) / total liabilities, inventory / sales, working capital / fixed assets, net profit / sales, liquid assets / operating expenses, long-term liabilities / equity, total sales / total assets, constant capital / fixed assets, operating expenses / short-term liabilities, net profit / inventory.

Since some features are correlated with each other, we summary those features by their accounting classification and here are the three important perspectives for detecting bankruptcy.

Profitability: Accounting terms relevant to profit, like (net profit + depreciation) / total liabilities, net profit/sales are critical features to detect bankruptcy. Firms without promising profit are less likely to maintain their business thus go bankrupt.

Turnover ratio: Accounting terms relevant to inventory, like inventory/sales, net profit/inventory, are also important features since the turnover ratio indicates the production and sale process condition. If a firm has high inventory but low sales, the goods are very likely to outmode and face the depreciation risk.

Capital Structure: Accounting terms relevant to the balance sheet, like working capital / fixed assets, long-term liabilities/equity are influential to detect the firms` financial health. Illiquid asset or short on cash is a dangerous signal that shows both poor management and potential bad income quality.

5. Conclusion

In this project, we use classical supervised learning model and ensemble learning, and their optimized model to predict bankruptcy, with financial index as input.

One critical issue of our project is imbalance data. The number of bankrupted instances is small than the whole sample size, resulting in overfitting toward the large one. In practice, we pay more attention to the bankrupted instance and recall ratio, since the investor tends to be more conservative and risk-aversion.

Overall, the XGBoosting in the ensemble learning performs best than the other five models. In the test sample size, the accuracy, precision, recall, F1 score and F2 score are 92.1%, 45.25%, 63.41%, 52.69%, and 58.6% respectively.

Our model results show that profitability, turnover ratio, and capital structure ratio are the most important feature aspects in detecting bankruptcy possibility.

References

[1] Ohlson, JA (1980). Financial ratios and the probabilistic prediction of bankruptcy. Journal of Accounting Research, 18(1), 109-131.

[2] Begley, J., Ming, J., & Watts, S. (1996). Bankruptcy classification errors in the 1980s: An empirical analysis of Altman's and Ohlson's model. Review of Accounting Studies, 1(4), 267-284.

[3] Wang, G., Ma, J., & Yang, S. (2014). An improved boosting based on feature selection for corporate bankruptcy prediction. Expert Systems with Applications, 41(5), 2353-2361.

Appendix – A List of Financial Statement Attributes

X1 net profit / total assets
X2 total liabilities / total assets
X3 working capital / total assets
X4 current assets / short-term liabilities
X5 [(cash + short-term securities + receivables - short-term liabilities) / (operating expenses - depreciation)] * 365
X6 retained earnings / total assets
X7 EBIT / total assets
X8 book value of equity / total liabilities
X9 sales / total assets
X10 equity / total assets
X11 (gross profit + extraordinary items + financial expenses) / total assets
X12 gross profit / short-term liabilities
X13 (gross profit + depreciation) / sales
X14 (gross profit + interest) / total assets
X15 (total liabilities * 365) / (gross profit + depreciation)
X16 (gross profit + depreciation) / total liabilities
X17 total assets / total liabilities
X18 gross profit / total assets
X19 gross profit / sales
X20 (inventory * 365) / sales
X21 sales (n) / sales (n-1)
X22 profit on operating activities / total assets
X23 net profit / sales
X24 gross profit (in 3 years) / total assets
X25 (equity - share capital) / total assets
X26 (net profit + depreciation) / total liabilities
X27 profit on operating activities / financial expenses

X28 working capital / fixed assets
X29 logarithm of total assets
X30 (total liabilities - cash) / sales
X31 (gross profit + interest) / sales
X32 (current liabilities * 365) / cost of products sold
X33 operating expenses / short-term liabilities
X34 operating expenses / total liabilities
X35 profit on sales / total assets
X36 total sales / total assets
X37 (current assets - inventories) / long-term liabilities
X38 constant capital / total assets
X39 profit on sales / sales
X40 (current assets - inventory - receivables) / short-term liabilities
X41 total liabilities / ((profit on operating activities + depreciation) * (12/365))
X42 profit on operating activities / sales
X43 rotation receivables + inventory turnover in days
X44 (receivables * 365) / sales
X45 net profit / inventory
X46 (current assets - inventory) / short-term liabilities
X47 (inventory * 365) / cost of products sold
X48 EBITDA (profit on operating activities - depreciation) / total assets
X49 EBITDA (profit on operating activities - depreciation) / sales
X50 current assets / total liabilities
X51 short-term liabilities / total assets
X52 (short-term liabilities * 365) / cost of products sold)
X53 equity / fixed assets
X54 constant capital / fixed assets
X55 working capital
X56 (sales - cost of products sold) / sales
X57 (current assets - inventory - short-term liabilities) / (sales - gross profit - depreciation)
X58 total costs /total sales
X59 long-term liabilities / equity
X60 sales / inventory
X61 sales / receivables
X62 (short-term liabilities *365) / sales
X63 sales / short-term liabilities
X64 sales / fixed assets