Lehigh University

BUAN 352

Professor: Han Ye

Project Members: Juan Mozos Nieto, Sallie Wang, Weining Wu

Final Project:

Business Bankruptcy Prediction

December 5th , 2022

Introduction

Predictive business analytics pertains to the finding of patterns, trends or understanding of the structure of the data for the prediction of certain outcomes, or the attempt to do so through statistical computations. These statistical computations on the data are then used to develop predictive models, able to take in new data with a similar structure and output a prediction which can be used to draw conclusions that business could use to their advantage. This is therefore the purpose of this report, develop models using a dataset that will allow a business to predict bankruptcy. Over the past few years, there has been an increase in competition in most capitalist markets which has made business analytics a cutting edge for any business that is able to use it to its advantage, gaining leverage over the competition.

We considered this topic to be of special interest given the current state of the world's economic conditions. Over the past few years we have seen how most if not all countries in the world have been suffering from deteriorating economic conditions. This is mainly due to the COVID-19 pandemic and the restrictive measures, which caused a notable short term decrease in nations' economic output between 2020 and 2021. It could even be argued that there is another incoming recession in 2023 also stemming from the COVID-19 pandemic and the expansionary monetary policy undertaken at the time to reduce the negative economic consequences of the pandemic, which are now causing worldwide inflation. All in all these recessions have the power to increase the potential for bankruptcy. For example in the US bankruptcies reached their highest level in 10 years (Irum and Hudgins, 2021), and in a sample of countries in Europe and Asia, bankruptcy rates rose to 18% during the pandemic, doubling pre pandemic rates (Foy, 2020). It is therefore obvious that worsening economic conditions are a great threat to the

continued existence of private enterprises, and that is why we thought it was pertinent to conduct a predictive analysis to find the most significant predictors of bankruptcy.

Simply put, the objective of this report is to analyze the probability of a given company filing for bankruptcy using their financial data, and discovering which are the most significant indicators of bankruptcy probability. Bankruptcy being "a legal proceeding initiated when a person or company is unable to repay outstanding debts or obligations" (Tuovila, 2022). We wanted our analysis and findings on the topic to serve two different perspectives. One of those perspectives comes from the inside of a business, in other words a financial manager. A financial manager could use the analysis to predict the probability of bankruptcy of his or her company given their financial data. Looking now from the perspective of an outsider, like a current or prospective investor, the analysis could help in making investment decisions by shedding light into the probability of bankruptcy and the financial ratios that have the greatest effect on the probability of bankruptcy.

Data Understanding

In order to conduct our analysis, we used a data set we found on Kaggle, a webpage with a wide variety of datasets (Taiwan Economic Journal). We found a dataset of 6819 Taiwanese companies, meaning that the dataset originally had 6819 records. The dataset originally also had 95 variables. These variables represent financial metrics, ratios, and performance measures which are typically used in the financial industry to analyze the financial wellbeing of a company. A brief description of all the predictor variables, financial ratios and metrics, can be found in Table 1 of the Appendix. Some examples of the variables included in the data set are Operating Gross Margin, Revenue Per Share, or Interest Coverage Ratio. The dataset contained an additional variable, bankruptcy, our target variable which represented if the given company of

that record had gone bankrupt or not. It is a binary variable, with 1 indicating that the company had gone bankrupt and 0 indicating otherwise. This means that in total the dataset contained 96 columns. The original source of the dataset is the National Central University of Taiwan.

Data Preprocessing

Before developing any of the predictive models, it is important and very necessary to preprocess the dataset. In our case, this includes reducing the dimensions of the dataset, understanding the dataset through data exploration, and data preparation which involves data cleaning and data partitioning.

In the first place, we realized how the incredible amount of predictors (95) we had in the dataset could result in a drawback to the overall efficiency in building the predictive models. The readability of the output of the predictive models could be a problem with so many variables, running data exploration tasks were also going to be very tedious (especially if having to focus on a number of individual variables), and the high probability of most variables having little significance in the predictive models. For that reason, we used our own knowledge of the business world to select the predictors that are usually used in the financial industry to assess the financial position of a company. In this manner we were able to reduce the number of predictors from 95 to 17. There are no unprocessed missing values in the dataset after removing the unnecessary predictors. The predictors being selected and their definitions are shown in Table 1.

In order to get a better picture of the data we were working with, we first wanted to develop data visualizations to help us understand how the predictor variables related to the target variable; getting a better idea of the possible correlations between the predictor variables and the target variable. In order to do so, we first developed a correlation matrix for all of the variables in the data set using the “ggpairs” function from the GGally library (*Figure 1*). As stated before, the

large number of variables makes the readability of this visualization very complicated. Nevertheless, we can still see from the matrix that most of the predictor variables have a fairly low correlation (positive or negative) with the bankruptcy target variable (the rightmost variable), with correlation ranging from -0.261 to 0.250. We then used the output of the correlation matrix to select the three variables that had the highest correlation with the target variable and developed individual box plot visualizations to better understand their relation with bankruptcy. The variable with the highest correlation (-0.261) was ROA before interest and depreciation, judging by the boxplot we can see that companies with lower values for this metric tend to file for bankruptcy more often (*Figure 2*). The second variable with the highest correlation was debt ratio (0.250), judging by the boxplot visualization that companies with a higher debt ratio have a higher chance of filing for bankruptcy. The variable with the third highest correlation was Working Capital to Total Assets (-0.193), judging by the boxplot we can see that companies with a lower value for this variable tend to file for bankruptcy more often (*Figure 3*). It shall also be noted that to visualize the relationship between predictors and bankruptcy we preferred to use a boxplot over a scatter plot given that for a binary variable, scatterplots are unable to show the density and the distribution of the data as points tend to cluster along two vertical lines.

In terms of preparing the data for the predictive models we thought it was pertinent to first try and detect which predictor variables had outliers. In many cases, outlying data points can have a very significant effect on the output of predictive models by skewing or shifting the output of the model in a disproportionate manner. To detect which variables had outliers, we developed boxplot visualizations, these would show how many data points are outside of the acceptable range. The maximum threshold that determines if a point is considered an outlier is

the third quartile of the data plus 1.5 times the interquartile range. The minimum threshold that determines if a point is considered an outlier is the first quartile minus 1.5 times the interquartile range (*How to remove outliers from data in R,* 2022). In our case we found that most, if not all of our variables, had many outliers as shown by the boxplots (*Figure 1, Figure 2, Figure 3*). We were successfully able to remove all the outliers from the data, which reduced the number of records in the data from 6819 to 2902. Even though it might seem like being able to remove the outliers is a positive factor, we then realized that the clean dataset only contained 8 companies which had gone bankrupt, a very low bankruptcy sample which then resulted to be a limitation to the developed models. We will further expand on this phenomenon in the "limitations" section of this report.

Lastly in terms of data preparation, we decided to partition the data in order to avoid problems with overfitting. Overfitting takes place when attempting the "making an overly complex model to explain idiosyncrasies in the data under study. In reality, the data often studied has some degree of error or random noise within it. Thus, attempting to make the model conform too closely to slightly inaccurate data can infect the model with substantial errors and reduce its predictive power" (Twin, 2022). We therefore allocated 60% of the records from the dataset into the training set and the other 40% of the records into the validation set for model testing purposes.

Methodologies

Classification And Regression Tree (CART)

Compared with other algorithms, classification trees require less data preparation during pre-processing. Classification trees do not require normalization of the data and are not affected by outliers in the dataset. Therefore, we used a classification tree to create an overview of the full

dataset. We used 60% data as the training set to generate a model and use another 40% data to verify the accuracy of the model.

First, we used the default classification tree function in Rstudio to generate a default tree with the full data containing all outliers of the 17 variables we chose. The accuracy of the default tree was 96.48% as calculated by the figures displayed by its confusion matrix (*Figure 10*). According to the plot of the default tree (*Figure 9*), the natural purity of each leaf is less than 100%. At present, there are no obvious issues with overfitting of the model. However, to further ensure the simplicity of the model and its availability for any new data, we continued to prune the classification tree.

We grew the tree to full extent using 0.00001 as the complexity parameter when running the model. Rstudio then automatically uses cross validation to determine the best tree size (complexity parameter level). With the best complexity parameter, we generated a pruned tree. The accuracy of the pruned tree rose to 96.81% as calculated by the figures displayed by its confusion matrix (*Figure 12*). According to the pruned tree plot (*Figure 11*) and the model summary, important predictors that can help with prediction selected by classification tree include: total debt net worth, ROA, quick ratio, operating expense rate, debt ratio, current ratio, operating profit rate, and working capital to total assets.

Logistic Regression Model

After cleaning and data and eliminating any outliers, we ran a logistic regression model on all the 17 predictors (*Figure 5*) and found both very strong positive and negative correlations between our predictors and output. The top five strongest predictors include Operating Gross Margin, Gross Profit to Sales, Operating Profit Rate, Accounts Receivable Turnover Rate, and Quick Ratio. Out of the five, four of them have a strong negative coefficient. A negative

correlation in our data makes logistic sense as a unit of this financial metric increases, the likelihood of bankruptcy decreases. Take accounts receivable turnover rate for example, accounts receivable turnover rate in its essence predicts a firm's efficiency at collecting credit debt from customers. The more efficient a firm is at collecting credit debt, the higher the AR turnover rate, thus the lower the probability of that firm falling into bankruptcy. In terms of the accuracy of our regression model, we saw a high accuracy rate of 99.14% with an error rate of 0.86% from the confusion matrix created based on our model's prediction on the validation data (*Figure 6*).

Backwards Elimination

To further limit down to the most significant predictors, we ran a backwards elimination on our full logistic regression model. This backwards elimination limited the full model down to the five most significant predictors, including ROA, operating gross margin, quick ratio, total debt over total net worth, and cash turnover rate *(Figure 7*). The model reduced AIC from 96.69 to 80.14. The accuracy measure for our model after the backwards elimination also increased from the previous 99.14% to 99.31% with a reduced error rate of 0.69% *(Figure 8*). Making this model the most accurate thus far.

Since the backwards elimination method improved our model accuracy, we wanted to test out the stepwise elimination method to see if there are further improvements. However, we received the exact same output and accuracy rate as the backwards elimination. Which further proves that the five predictors chosen through the backwards elimination process are the five most significant predictors with the greatest influence on bankruptcy rate.

Limitations

One of the greatest limitations we encountered when working on the project was how to deal with outliers. When we first started developing the different logistic regression and the

classification tree models we found that the output of this was extremely skewed due to the high number of outliers in the data. We then decided it was sensible to remove all the outliers from the data in order to achieve a better performance of the different models. Nevertheless, when we did this we removed most of the records for companies that had gone bankrupt (Bankrupt. = 1), we were only left with eight bankrupt companies. This is a great problem as it means that the default model, which predicts all of the companies not to go bankrupt, would have a very high accuracy. This limited number may affect the accuracy of the logistic regression model and lead to the high false positive rate. The high positive rate results from the fact that most actual bankrupt companies are predicted not to go bankrupt by the model, which can be reflected by the Logistic Regression confusion matrix (*Figure 6*). Since the classification tree is not affected by outliers, it should be a more accurate model for the dataset we used in this analysis specifically. It successfully took into account all 220 bankruptcies in the raw data.

Another limitation we encountered when developing the logistic regression models was the significance of the predictor variables. Judging by the p-values as seen in Figure 5 and Figure 7, we can see that most of them have p-values greater than 0.05 or 0.1, these are cutoff values which determine the significance of a variable within a confidence interval, in our case most variables failed to have significance within a 95% or 90% confidence interval, specially for the full model. Nevertheless this problem was partially solved with the backwards elimination model, where all of the variables except for one resulted to be significant at least within a 90% confidence interval.

Conclusion

In conclusion, the logistic regression model after backwards elimination is our best model. It provides us with the top five most significant predictors and with a prediction accuracy

rate of 99.31%. Our classification tree model on the other hand, has a lower accuracy. Even without eliminating the outliers, our logistic regression full model returns an accuracy rate of 97.14% with an error rate of 2.86% (*Figure 13*), making it more accurate than our classification tree model. However, given that most bankrupt companies are deleted during the removal process of the outliers, the classification tree model is better as it draws a better picture on real world data

Relating back to the primary objective of our analysis, companies and investors can utilize our best model to predict their chance of bankruptcy given their financial measures, so they can take corresponding actions to make better investment or operational decisions. The predictors limited down through our backward elimination process is in particular helpful as in cases where data is limited, companies and investors can choose to focus on the five most influential financial measures to assess a firm's financial performance. All in all, the models we developed could result to be very helpful for a business manager, as a business manager could comparatively analyze his or her company's financial ratios with the coefficients of the logistic regression backwards elimination model and see which financial predictor will have the greatest effect on his or her company's probability of bankruptcy. The business manager could then implement different strategies to lower or increase the value of one of their financial ratios. From the perspective of an outside investor that is deciding whether to invest or not in a company, the prospective investor could calculate the ratios for a company, then plug them in into the equation developed by the backward elimination model to have a prediction of the probability of a given company to go bankrupt. Depending on the output from the equation the investor could then make a more informed investment decision.

Appendix

Figure 1: Correlation Matrix

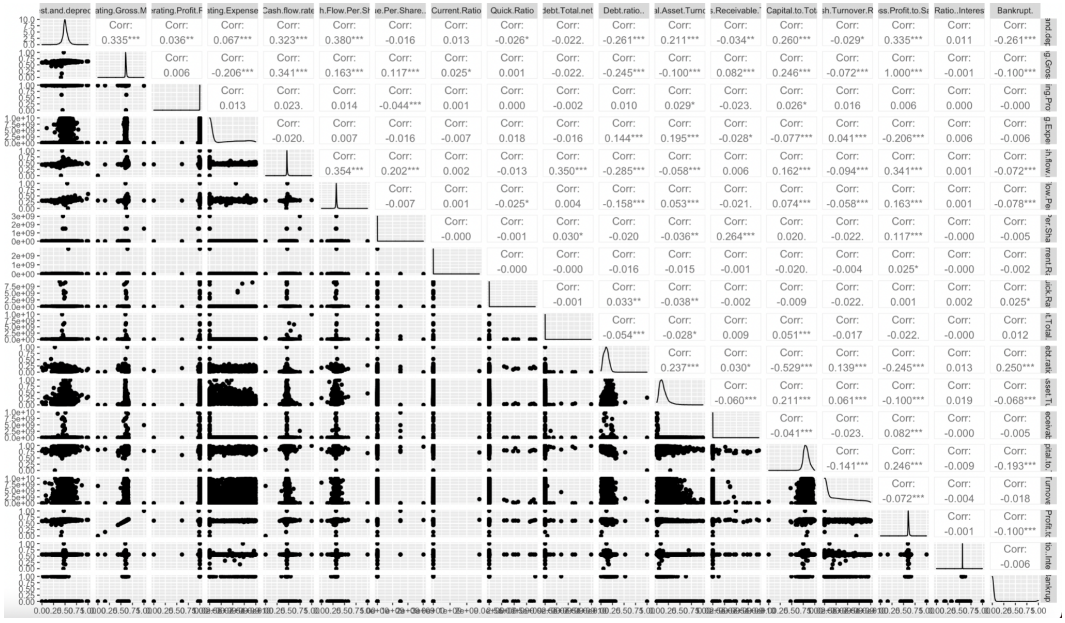

Figure 2: Boxplot visualization. ROA before interest and depreciation ~ Bankrupt.

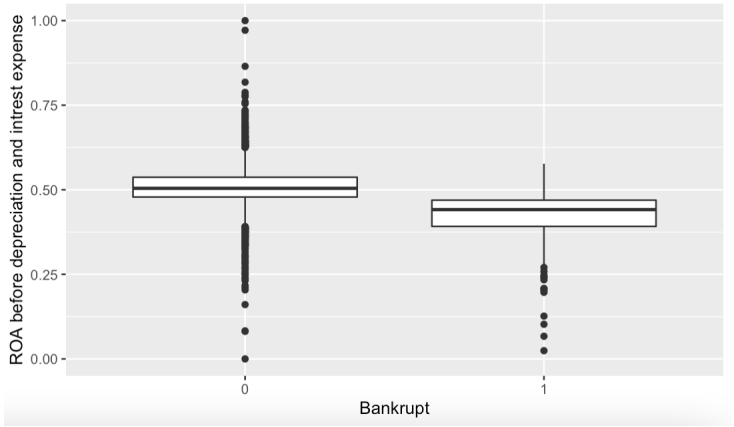

Figure 3: Boxplot visualization. Debt Ratio ~ Bankrupt.

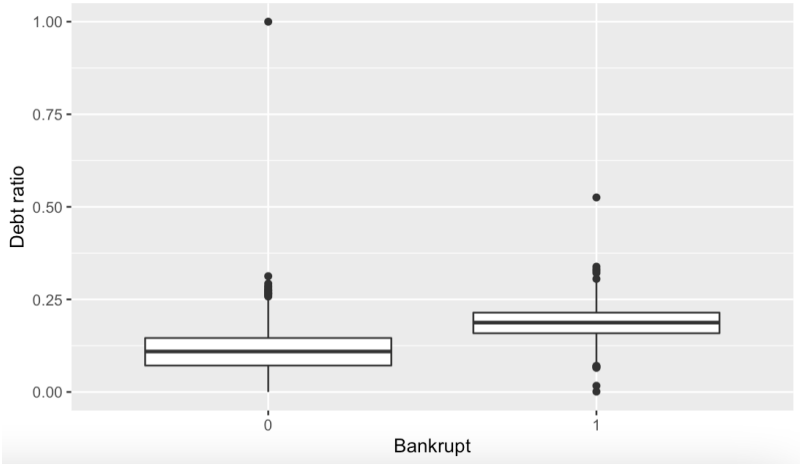

Figure 4: Boxplot visualization. Working Capital to Total Assets ~ Bankrupt.

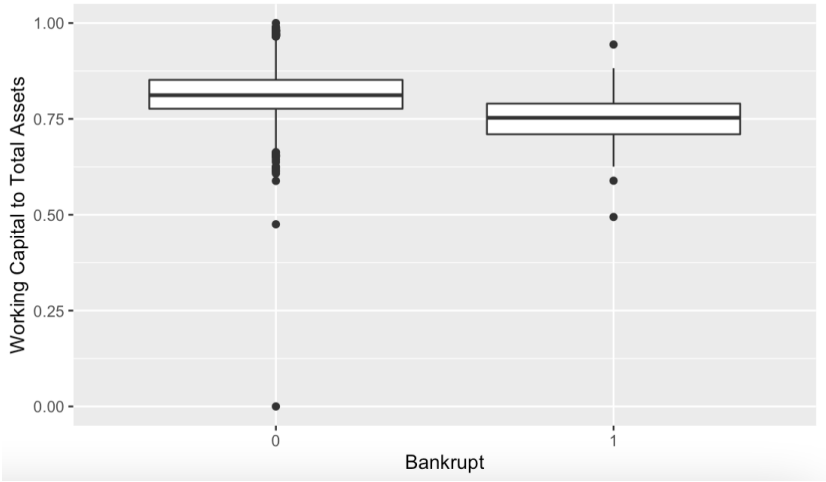

Figure 5: Logistic Regression Model (Summary output of full model)

```
Call:
glm(formula = Bankrupt. ~ ., family = "binomial", data = train.df)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-0.8601  -0.0501  -0.0176  -0.0048   3.4486

Coefficients:
                                                    Estimate          Std. Error z value Pr(>|z|)
(Intercept)                                     4966.5660686417550    8549.9947331575640   0.581  0.56132
ROA.C..before.interest.and.depreciation.before.interest   -26.9186747143145      19.5763334885983  -1.375  0.16911
Operating.Gross.Margin                        182561.6288716606214  211473.1914027775929   0.863  0.38798
Operating.Profit.Rate                          -5289.7529125973088    8600.4737017775678  -0.615  0.53852
Operating.Expense.Rate                             0.0000000001585       0.0000000001542   1.028  0.30418
Cash.flow.rate                                   131.2683188904731     229.5315579194317   0.572  0.56739
Cash.Flow.Per.Share                               55.7971219559159     116.6585633753956   0.478  0.63244
Revenue.Per.Share..Yuan...                        19.7700081718422      46.4232138064526   0.426  0.67021
Current.Ratio                                    199.0954897923150     447.7089657043733   0.445  0.65654
Quick.Ratio                                   -1005.0201732547114     389.5138976122160  -2.580  0.00987 **
Total.debt.Total.net.worth                       351.5710099762817     564.1109632861446   0.623  0.53313
Debt.ratio..                                     -12.2338850153506      68.6932310003629  -0.178  0.85865
Total.Asset.Turnover                               4.3117304337558      16.1034244351168   0.268  0.78889
Accounts.Receivable.Turnover                  -2194.9469295588201    1501.0627086645306  -1.462  0.14367
Working.Capital.to.Total.Assets                    1.7029072520209      27.9500929811494   0.061  0.95142
Cash.Turnover.Rate                                -0.0000000003593       0.0000000002076  -1.731  0.08345 .
Gross.Profit.to.Sales                         -182393.5183659748000  211478.7859643335396  -0.862  0.38843
Interest.Coverage.Ratio..Interest.expense.to.EBIT.   259.2813259681904     565.3123935872998   0.459  0.64648
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Figure 6: Confusion Matrix of Logistic Regression Model

```
> conf.mat
          Actual
Prediction    0    1
         0 1151    8
         1    2    0
```

Figure 7: Backwards Elimination on Logistic Regression Model (Summary output)

```
Call:
glm(formula = Bankrupt. ~ ROA.C..before.interest.and.depreciation.before.interest +
    Operating.Gross.Margin + Quick.Ratio + Total.debt.Total.net.worth +
    Cash.Turnover.Rate, family = "binomial", data = train.df)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-0.76808  -0.06927  -0.03104  -0.01324   3.14843

Coefficients:
                                                      Estimate         Std. Error z value Pr(>|z|)
(Intercept)                                         -80.4055548553014   32.9653102983734  -2.439   0.0147 *
ROA.C..before.interest.and.depreciation.before.interest  -18.8089039100212   11.3403724895802  -1.659   0.0972 .
Operating.Gross.Margin                               141.2683788447580   57.3944036004955   2.461   0.0138 *
Quick.Ratio                                         -492.4967359708086  212.7121807105868  -2.315   0.0206 *
Total.debt.Total.net.worth                           228.2258391264292  115.3214498682432   1.979   0.0478 *
Cash.Turnover.Rate                                    -0.0000000002590    0.0000000001838  -1.409   0.1588
---


Step:  AIC=80.14
Bankrupt. ~ ROA.C..before.interest.and.depreciation.before.interest +
    Operating.Gross.Margin + Quick.Ratio + Total.debt.Total.net.worth +
    Cash.Turnover.Rate

                                                    Df Deviance    AIC
<none>                                                   68.136 80.136
- Cash.Turnover.Rate                                 1   70.582 80.582
- ROA.C..before.interest.and.depreciation.before.interest  1   70.716 80.716
- Total.debt.Total.net.worth                         1   71.922 81.922
- Operating.Gross.Margin                             1   73.758 83.758
- Quick.Ratio                                        1   75.344 85.344
```

Figure 8: Confusion Matrix of Backwards Regression Model

```
> conf.mat
          Actual
Prediction    0    1
         0 1153    8
```

Figure 9: Default Tree Plot

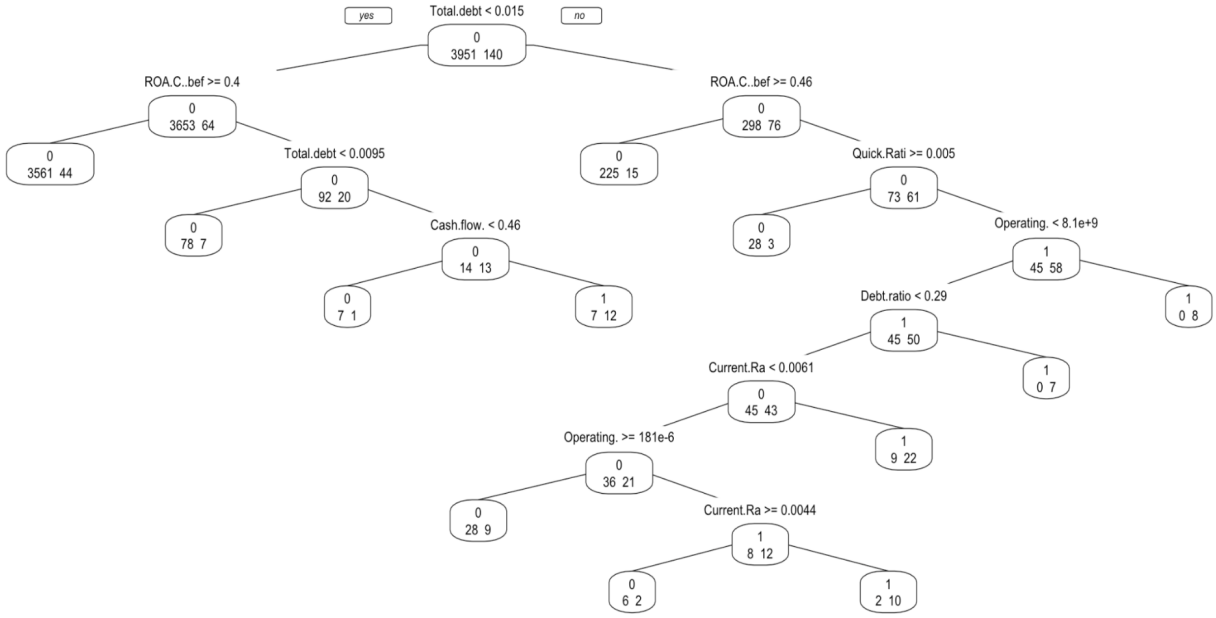

Figure 10: Default Tree Confusion Matrix

```
> default.ConMatrix
          Actual
Prediction    0    1
         0 2617   65
         1   31   15
```

Figure 11: Pruned Tree Plot

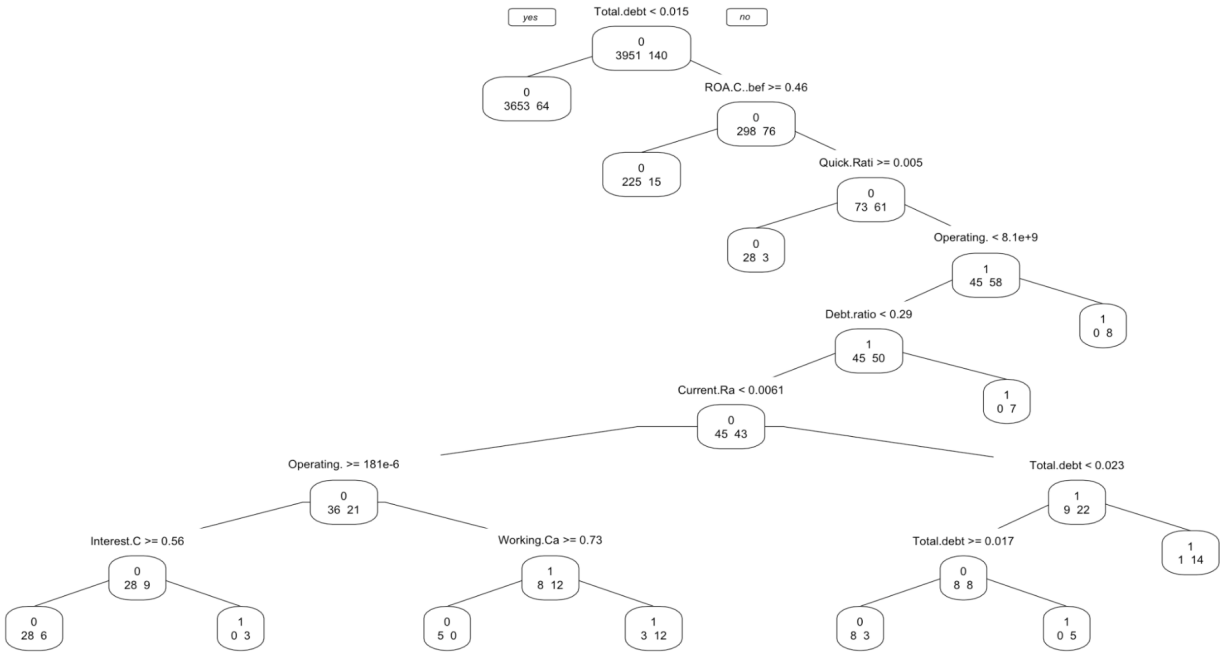

Figure 12: Pruned Tree Plot Confusion Matrix

```
> pruned.ConMatrix
          actual
prediction    0    1
         0 2630   69
         1   18   11
```

Figure 13: Logistic Regression Model Summary & Confusion Matrix without eliminating outliers

```
Call:
glm(formula = Bankrupt. ~ ., family = "binomial", data = train.df)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-5.1071  -0.2033  -0.1064  -0.0455   3.5535

Coefficients:
                                                      Estimate           Std. Error z value             Pr(>|z|)
(Intercept)                                     -175.81293887207141    105.11128619724448  -1.673              0.09440 .
ROA.C..before.interest.and.depreciation.before.interest   -18.15244470620730      2.10244544170883  -8.634 < 0.0000000000000002 ***
Operating.Gross.Margin                        51291.85471529551432  48752.30683815639350   1.052              0.29276
Operating.Profit.Rate                           177.92834274428799    108.87888110062127   1.634              0.10222
Operating.Expense.Rate                            0.00000000003701      0.00000000003144   1.177              0.23911
Cash.flow.rate                                    6.88666737475370     22.43940665167499   0.307              0.75892
Cash.Flow.Per.Share                               0.78188663710761      5.95643296166520   0.131              0.89556
Revenue.Per.Share..Yuan...                       -0.00000000438998      0.00000029326120  -0.015              0.98806
Current.Ratio                                  -130.88465395560559     63.08171543426722  -2.075              0.03800 *
Quick.Ratio                                       0.00000000001541      0.00000000013883   0.111              0.91162
Total.debt.Total.net.worth                        0.00000000370425      0.00000000136486   2.714              0.00665 **
Debt.ratio..                                     18.62689581037097      2.57933974107000   7.222    0.000000000000514 ***
Total.Asset.Turnover                             -6.41566895681987      1.46254240176206  -4.387    0.000011510714032 ***
Accounts.Receivable.Turnover                     -0.00000000059020      0.00000000067636  -0.873              0.38287
Working.Capital.to.Total.Assets                   3.80597400452534      4.14735096501622   0.918              0.35878
Cash.Turnover.Rate                               -0.00000000008462      0.00000000004093  -2.068              0.03868 *
Gross.Profit.to.Sales                         -51296.85531881396309  48752.76427531540685  -1.052              0.29272
Interest.Coverage.Ratio..Interest.expense.to.EBIT.   -1.54534587349006      4.78802180488556  -0.323              0.74688
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
> conf.mat
          Actual
Prediction    0    1
         0 2636   66
         1   12   14
```

Table 1: Selected Predictors and Their Definitions

Predictor	Definition
ROA Before Interest and Depreciation	Net Income / Total Assets. Indicates how profitable a company is in relation to its total assets.
Operating Gross Margin	Usually the variable costs that can be associated with production of goods.
Operating Profit Rate	Operating Income / Net Sales. Depicts how much profit a business is making for each dollar worth of sales it is making.
Operating Expense Rate	(Operating Expense + Cost of Goods Sold) / Net Sales. A lower operating expense ratio means that expenses are minimized relative to revenue.
Cash Flow Rate	Cash Flow / Current liabilities. A measure of the number of times a company can pay off current debts with cash generated within the same period.
Cash Flow Per Share	(Operating Cash flow - Preferred Dividends) / Total Common Shares Outstanding. A measure of a firm's financial strength.
Revenue Per Share (Yuan)	Revenue / Total Common Shares Outstanding. Identifies a company's productivity per share outstanding.
Current Ratio	Current Assets / Current Liabilities. Measures a company’s ability to pay short-term obligations or those due within one year.
Quick Ratio	Liquid Current Assets / Current Liabilities. Measures readily liquid current assets to the total amount of current obligations.
Total Debt to Total Net Worth	Total Liabilities/Total Net Worth. Net Worth = Assets - Liabilities - Intangible Assets. Compares the level of debt to the net worth of the company.
Debt Ratio	Total Debt/Total Assets. Measures the % of assets funded with debt.
Total Asset Turnover	Sales/Average Total Assets. A measure of how well a company's assets generate revenue.
Accounts Receivable Turnover	Net Credit Sales/Average Accounts receivable. A measure

	of a company's ability to collect credit extended to customers on sales of goods.
Working Capital to Total Assets	(current assets - current liabilities) / total assets. Looks at the net liquid assets to the total assets of the firm.
Cash Turnover Rate	Revenue/Cash. Informs the number of times that cash is turnover over a certain period of time.
Gross Profit to Sales	(Sales - Cost of Goods Sold)/Sales. Shows profit generated before selling expenses, interest expense, administrative expenses etc.
Interest Coverage Ratio	EBIT/Interest Expense. A measure of how well a company is able to cover interest obligations on outstanding debt using cash flows from sales.

Works Cited

Tuovila, A. (2022, November 23). *Bankruptcy explained: Types and how it works*. Investopedia. Retrieved December 4, 2022, from https://www.investopedia.com/terms/b/bankruptcy.asp

Hudgins, C., & Irum, T. (2021, January 5). *US corporate bankruptcies end 2020 at 10-year high amid covid-19 pandemic*. S&P Global. Retrieved December 4, 2022, from https://www.spglobal.com/marketintelligence/en/news-insights/latest-news-headlines/us-corporate-bankruptcies-end-2020-at-10-year-high-amid-covid-19-pandemic-61973656

Foy, M. (2020, October 29). *Small business failures in Europe would have doubled without pandemic relief, analysis finds: Haas News: Berkeley Haas*. Haas News | Berkeley Haas. Retrieved December 4, 2022, from https://newsroom.haas.berkeley.edu/research/covid-19-crisis-would-have-doubled-small-business-bankruptcies-in-europe-pandemic-relief/

How to remove outliers from data in R. Universe of Data Science. (2022, March 4). Retrieved December 4, 2022, from https://universeofdatascience.com/how-to-remove-outliers-from-data-in-r/

Taiwan Economic Journal. Taiwanese Bankruptcy Prediction Data Set [Data set]. Kaggle. https://www.kaggle.com/datasets/fedesoriano/company-bankruptcy-prediction

Twin, A. (2022, November 15). *Understanding overfitting and how to prevent it*. Investopedia. Retrieved December 4, 2022, from https://www.investopedia.com/terms/o/overfitting.asp

Project R Codes

```
rm(list=ls())
library(caret)
library(neuralnet)
library(gains)
library(ggplot2)
library(forecast)

# Read & clean up the file
business <- read.csv("Full_Data.csv")
business <- business[-1]  #remove the first column

#Correlation Matrix
library(GGally)
ggpairs(business)

#Converting Bankrupt target variable from numerical to categorical variable
business$Bankrupt. <- as.factor(business$Bankrupt.)
```

```
# CART
# using the data with all outliers

set.seed(1)
tree.train.rows <-  sample(1:nrow(business),nrow(business)*0.6)
tree.train.df <-  business[tree.train.rows,]
tree.valid.df <-  business[-tree.train.rows,]


library(rpart)
library(rpart.plot)

## Default tree ##
# only have one node
default.ct = rpart(Bankrupt. ~ ., data=tree.train.df, method="class")
prp(default.ct, type = 1, extra = 1, split.font = 1, varlen = -10)

default.pred <- predict(default.ct,tree.valid.df, type="class")

default.ConMatrix <- table(Prediction=default.pred, Actual= tree.valid.df$Bankrupt.)
default.ConMatrix

# Accuracy rate
(2617+15)/sum(default.ConMatrix)  #96.48%

## Prune Tree ##
## set the smallest value for cp
```

```
cv.ct <- rpart(Bankrupt. ~ ., data = tree.train.df, method = "class", minsplit=2, cp = 0.00001, xval
= 5)
# use printcp() to print the CP table.
printcp(cv.ct)
prp(cv.ct, type = 1, extra = 1, split.font = 1, varlen = -10)

# Prediction for unseen data set
cv.ct.pred = predict(cv.ct,tree.valid.df, type="class")
# Confusion Matrix
smallcp.ConMatrix <- table(prediction = cv.ct.pred, actual = tree.valid.df$Bankrupt.)
smallcp.ConMatrix
# Accuracy rate
(2583+23)/sum(smallcp.ConMatrix)  #95.53%


## prune tree
# cp = 0.12500
# only have one node
pruned.ct <- prune(cv.ct,
                   cp = cv.ct$cptable[which.min(cv.ct$cptable[,"xerror"]),"CP"])
prp(pruned.ct, type = 1, extra = 1, split.font = 1, varlen = -10)
summary(pruned.ct)
pruned.pred = predict(pruned.ct,tree.valid.df, type="class")
# Confusion Matrix
pruned.ConMatrix <- table(prediction = pruned.pred, actual = tree.valid.df$Bankrupt.)
pruned.ConMatrix
# Accuracy rate
(2630+11)/sum(pruned.ConMatrix)  #96.81%
```

```
#Box Plot Visualizations
p <- ggplot(data = business, mapping = aes(x = Bankrupt., y =
ROA.C..before.interest.and.depreciation.before.interest))
p + geom_boxplot() +
  ylab("ROA before depreciation and intrest expense") +
  xlab("Bankrupt")

p <- ggplot(data = business, mapping = aes(x = Bankrupt., y = Debt.ratio..))
p + geom_boxplot() +
  ylab("Debt ratio") +
  xlab("Bankrupt")

p <- ggplot(data = business, mapping = aes(x = Bankrupt., y = Working.Capital.to.Total.Assets))
p + geom_boxplot() +
```

```
  ylab("Working Capital to Total Assets") +
  xlab("Bankrupt")
```

```
#Boxplot visualizations and removing outliers
p <- ggplot(data = business, mapping = aes(y =
ROA.C..before.interest.and.depreciation.before.interest))
p + geom_boxplot()#possible

Q1 <- quantile(business$ROA.C..before.interest.and.depreciation.before.interest, 0.25)
Q3 <- quantile(business$ROA.C..before.interest.and.depreciation.before.interest, 0.75)
IQR <- IQR(business$ROA.C..before.interest.and.depreciation.before.interest)
business <- subset(business, business$ROA.C..before.interest.and.depreciation.before.interest >
(Q1 - 1.5*IQR) & business$ROA.C..before.interest.and.depreciation.before.interest < (Q3 +
1.5*IQR))

p <- ggplot(data = business, mapping = aes(y = Operating.Gross.Margin))
p + geom_boxplot()#possible

Q1 <- quantile(business$Operating.Gross.Margin, 0.25)
Q3 <- quantile(business$Operating.Gross.Margin, 0.75)
IQR <- IQR(business$Operating.Gross.Margin)
business <- subset(business, business$Operating.Gross.Margin > (Q1 - 1.5*IQR) &
business$Operating.Gross.Margin < (Q3 + 1.5*IQR))

p <- ggplot(data = business, mapping = aes(y = Operating.Profit.Rate))
p + geom_boxplot()

Q1 <- quantile(business$Operating.Profit.Rate, 0.25)
Q3 <- quantile(business$Operating.Profit.Rate, 0.75)
IQR <- IQR(business$Operating.Profit.Rate)
business <- subset(business, business$Operating.Profit.Rate > (Q1 - 1.5*IQR) &
business$Operating.Profit.Rate < (Q3 + 1.5*IQR))

p <- ggplot(data = business, mapping = aes(y = Operating.Expense.Rate))
p + geom_boxplot()

Q1 <- quantile(business$Operating.Expense.Rate, 0.25)
Q3 <- quantile(business$Operating.Expense.Rate, 0.75)
IQR <- IQR(business$Operating.Expense.Rate)
business <- subset(business, business$Operating.Expense.Rate > (Q1 - 1.5*IQR) &
business$Operating.Expense.Rate < (Q3 + 1.5*IQR))

p <- ggplot(data = business, mapping = aes(y = Cash.flow.rate))
```

```
p + geom_boxplot()#possible

Q1 <- quantile(business$Cash.flow.rate, 0.25)
Q3 <- quantile(business$Cash.flow.rate, 0.75)
IQR <- IQR(business$Cash.flow.rate)
business <- subset(business, business$Cash.flow.rate > (Q1 - 1.5*IQR) &
business$Cash.flow.rate < (Q3 + 1.5*IQR))

p <- ggplot(data = business, mapping = aes(y = Cash.Flow.Per.Share))
p + geom_boxplot()#possible

Q1 <- quantile(business$Cash.Flow.Per.Share, 0.25)
Q3 <- quantile(business$Cash.Flow.Per.Share, 0.75)
IQR <- IQR(business$Cash.Flow.Per.Share)
business <- subset(business, business$Cash.Flow.Per.Share > (Q1 - 1.5*IQR) &
business$Cash.Flow.Per.Share < (Q3 + 1.5*IQR))

p <- ggplot(data = business, mapping = aes(y = Revenue.Per.Share..Yuan...))
p + geom_boxplot()

Q1 <- quantile(business$Revenue.Per.Share..Yuan..., 0.25)
Q3 <- quantile(business$Revenue.Per.Share..Yuan..., 0.75)
IQR <- IQR(business$Revenue.Per.Share..Yuan...)
business <- subset(business, business$Revenue.Per.Share..Yuan... > (Q1 - 1.5*IQR) &
business$Revenue.Per.Share..Yuan... < (Q3 + 1.5*IQR))


p <- ggplot(data = business, mapping = aes(y = Current.Ratio))
p + geom_boxplot()

Q1 <- quantile(business$Current.Ratio, 0.25)
Q3 <- quantile(business$Current.Ratio, 0.75)
IQR <- IQR(business$Current.Ratio)
business <- subset(business, business$Current.Ratio > (Q1 - 1.5*IQR) & business$Current.Ratio
< (Q3 + 1.5*IQR))

p <- ggplot(data = business, mapping = aes(y = business$Quick.Ratio))
p + geom_boxplot()#possible

Q1 <- quantile(business$Quick.Ratio, 0.25)
Q3 <- quantile(business$Quick.Ratio, 0.75)
IQR <- IQR(business$Quick.Ratio)
business <- subset(business, business$Quick.Ratio > (Q1 - 1.5*IQR) & business$Quick.Ratio <
(Q3 + 1.5*IQR))

p <- ggplot(data = business, mapping = aes(y = business$Total.debt.Total.net.worth))
```

```
p + geom_boxplot()#possible

Q1 <- quantile(business$Total.debt.Total.net.worth, 0.25)
Q3 <- quantile(business$Total.debt.Total.net.worth, 0.75)
IQR <- IQR(business$Total.debt.Total.net.worth)
business <- subset(business, business$Total.debt.Total.net.worth > (Q1 - 1.5*IQR) &
business$Total.debt.Total.net.worth < (Q3 + 1.5*IQR))

p <- ggplot(data = business, mapping = aes(y = business$Debt.ratio..))
p + geom_boxplot()#possible

Q1 <- quantile(business$Debt.ratio.., 0.25)
Q3 <- quantile(business$Debt.ratio.., 0.75)
IQR <- IQR(business$Debt.ratio..)
business <- subset(business, business$Debt.ratio.. > (Q1 - 1.5*IQR) & business$Debt.ratio.. <
(Q3 + 1.5*IQR))

p <- ggplot(data = business, mapping = aes(y = business$Total.Asset.Turnover))
p + geom_boxplot()#possible

Q1 <- quantile(business$Total.Asset.Turnover, 0.25)
Q3 <- quantile(business$Total.Asset.Turnover, 0.75)
IQR <- IQR(business$Total.Asset.Turnover)
business <- subset(business, business$Total.Asset.Turnover > (Q1 - 1.5*IQR) &
business$Total.Asset.Turnover < (Q3 + 1.5*IQR))

p <- ggplot(data = business, mapping = aes(y = business$Accounts.Receivable.Turnover))
p + geom_boxplot()#possible

Q1 <- quantile(business$Accounts.Receivable.Turnover, 0.25)
Q3 <- quantile(business$Accounts.Receivable.Turnover, 0.75)
IQR <- IQR(business$Accounts.Receivable.Turnover)
business <- subset(business, business$Accounts.Receivable.Turnover > (Q1 - 1.5*IQR) &
business$Accounts.Receivable.Turnover < (Q3 + 1.5*IQR))

p <- ggplot(data = business, mapping = aes(y = business$Working.Capital.to.Total.Assets))
p + geom_boxplot()#possible

Q1 <- quantile(business$Working.Capital.to.Total.Assets, 0.25)
Q3 <- quantile(business$Working.Capital.to.Total.Assets, 0.75)
IQR <- IQR(business$Working.Capital.to.Total.Assets)
business <- subset(business, business$Working.Capital.to.Total.Assets > (Q1 - 1.5*IQR) &
business$Working.Capital.to.Total.Assets < (Q3 + 1.5*IQR))

p <- ggplot(data = business, mapping = aes(y = business$Cash.Turnover.Rate))
p + geom_boxplot()
```

```
Q1 <- quantile(business$Cash.Turnover.Rate, 0.25)
Q3 <- quantile(business$Cash.Turnover.Rate, 0.75)
IQR <- IQR(business$Cash.Turnover.Rate)
business <- subset(business, business$Cash.Turnover.Rate > (Q1 - 1.5*IQR) &
business$Cash.Turnover.Rate < (Q3 + 1.5*IQR))

p <- ggplot(data = business, mapping = aes(y = business$Gross.Profit.to.Sales))
p + geom_boxplot()#possible

Q1 <- quantile(business$Gross.Profit.to.Sales, 0.25)
Q3 <- quantile(business$Gross.Profit.to.Sales, 0.75)
IQR <- IQR(business$Gross.Profit.to.Sales)
business <- subset(business, business$Gross.Profit.to.Sales > (Q1 - 1.5*IQR) &
business$Gross.Profit.to.Sales < (Q3 + 1.5*IQR))

p <- ggplot(data = business, mapping = aes(y =
business$Interest.Coverage.Ratio..Interest.expense.to.EBIT.))
p + geom_boxplot()#possible

Q1 <- quantile(business$Interest.Coverage.Ratio..Interest.expense.to.EBIT., 0.25)
Q3 <- quantile(business$Interest.Coverage.Ratio..Interest.expense.to.EBIT., 0.75)
IQR <- IQR(business$Interest.Coverage.Ratio..Interest.expense.to.EBIT.)
business <- subset(business, business$Interest.Coverage.Ratio..Interest.expense.to.EBIT. > (Q1 -
1.5*IQR) & business$Interest.Coverage.Ratio..Interest.expense.to.EBIT. < (Q3 + 1.5*IQR))
```

```
summary(business)
str(business)

# data partition
set.seed(1)
train.rows <-  sample(1:nrow(business),nrow(business)*0.6)
train.df <-  business[train.rows,]
valid.df <-  business[-train.rows,]
```

```
# Logistic regression models
# Logistic Regression with all predictors
logit.reg1 <- glm(Bankrupt. ~.,
                  data = train.df,
                  family = "binomial")
options(scipen=999)
summary(logit.reg1)
```

```
#confusion matrix
valid.pred = predict(logit.reg1,
                     valid.df,
                     type="response")
# classify predicted probabilities into 0/1, threshold: 0.5
valid.pred.bin = ifelse(valid.pred>0.5, 1, 0)
conf.mat = table(Prediction=valid.pred.bin,
                 Actual= valid.df$Bankrupt.)
conf.mat
# Accuracy rate
(1151+0)/sum(conf.mat)  #99.14%
# Error rate
(8+2)/sum(conf.mat)  #0.86%
```

```
# backward regression
business.lg.back <- step(logit.reg1, direction="backward") #left with 5 most significant
predictors
options(scipen=999)
summary(business.lg.back)

# Confusion Matrix
valid.pred = predict(business.lg.back,
                     valid.df,
                     type="response")
valid.pred.bin = ifelse(valid.pred>0.5, 1, 0)
conf.mat = table(Prediction=valid.pred.bin,
                 Actual= valid.df$Bankrupt.)
conf.mat
# Accuracy rate
(1153)/sum(conf.mat)  #99.31%
# Error rate
(8)/sum(conf.mat)  #0.69%

#Accuracy on full model w/o eliminating outliers
(2636+14)/sum(conf.mat)  #97.14%
# Error rate
(12+66)/sum(conf.mat)  #2.86%
```

```
# Stepwise regression
business.lg.step = step(logit.reg1,
                        direction="both")
```

```
options(scipen=999)
summary(business.lg.step)

# Confusion Matrix
valid.pred = predict(business.lg.step,
                     valid.df,
                     type="response")
valid.pred.bin = ifelse(valid.pred>0.5, 1, 0)
conf.mat = table(Prediction=valid.pred.bin,
                 Actual= valid.df$Bankrupt.)
conf.mat
# Accuracy rate
(1153)/sum(conf.mat)  #99.31%
# Error rate
(8)/sum(conf.mat)  #0.69%
```