Final Project - Bankruptcy Prediction

Team members:

```
107701012 應數四 楊書瑋
108302068 金融三 楊念臻
```

Introduction

The objective of this project is to build a Machine Learning Model that predicts the bankruptcy of a company. Some of the steps to follow are to prepare and pre-process the data before starting to analyze. Use different machine learning algorithms and evaluate their performance. Optimize the model and finally run the model on testing data to observe its output.

In this project, we select different models to predict the result and choose the one with the best performance. Finally, we use the models to predict the default probabilities and evaluate if the companies will bankrupt.

Data

The data were collected from the Taiwan Economic Journal for the years 1999 to 2009. Company bankruptcy was defined based on the business regulations of the Taiwan Stock Exchange. We use 80% of the data as the training data and 20% of that as testing data, and because the original name of the parameters in the data is long, we change them to the class label of bankruptcy, $X_2, \ldots, X_{95}$ to make the training process easier.

This dataset contains some important indicators, including ROA, current ratio, total asset turnover, total assets, current liability, etc. Thus, we think this is a good dataset to be used for bankruptcy prediction.

Some Attribute Information

Y - Bankrupt?: Class label
X_1 - ROA(C) before interest and depreciation before interest: Return On Total Assets(C)
X_2 - ROA(A) before interest and % after-tax: Return On Total Assets(A)
X_3 - ROA(B) before interest and depreciation after tax: Return On Total Assets(B)
X_{33} - Current Ratio
X_{37} - Debt ratio $\%$: $Liability \ / \ Total \ Assets$
X_{45} - Total Asset Turnover
X_{93} - Interest Coverage Ratio (Interest expense to EBIT)
X_{95} - Equity to Liability

Prepare

Libraries required

```
library(readr)
library(data.table)
library(tidyverse)
library(caret)
library(glmnet)
```

Obtain the data

Download `data.csv` from https://www.kaggle.com/datasets/fedesoriano/company-bankruptcy-prediction?datasetId=1111894&language=R

```
data <- read_csv("~/Documents/R/Final Project/data.csv", col_names = FALSE, skip = 1)
data <- as.data.table(data)
```

Pre-process

Missing values

```
any(is.na(data))
```

```
> any(is.na(data))
[1] FALSE
```

Data partition for validation

```
## train dataset 80%, test dataset 20%
train.index <- sample(x=1:nrow(data), size=ceiling(0.8*nrow(data) ))
train = data[train.index, ]
test = data[-train.index, ]
```

Methodology

Since this dataset contains 95 parameters, it will take lots of time to train. We would like to know whether we can get the same or better result if we use fewer parameters. In the other words, we would like to remove the redundant parameters. After removing them, we will use cross-validation for `glmnet` to predict its probability of bankruptcy.

First of all, we use some model selection methods introduced in the lesson, such as forward stepwise, and backward stepwise. In addition, we also try both stepwise and the parameters with a higher correlation with bankruptcy to fit the model.

Second, we use the parameters selected by the above model to fit the `glm` model. Here we use glm to fit the model due to binary response variables (bankrupt: 1, Non-bankrupt: 0). After using the `glm` models to predict the test dataset, we can compare the results from the different model selection methods.

Finally, we tried to predict their precise probabilities of bankruptcy, however, we were unable to obtain precise probabilities by using the `glm` model. In the glm model, we only get 0 or 1 as our result because we don't have a large enough dataset. Thus, we use the parameters obtained by selection to fit the cross-validation `glmnet` model. Then use them to predict the precise probabilities and compare the results.

Findings

First, we will see the predicted result from the glm model fitted by each model selection. Then, we will discuss their `cv.glmnet` predict results which is the precise probability of bankruptcy.

The following chart shows the relation between the real result(row-wise) and predicts the result(column-wise). We will focus on the last row to determine its accuracy. Because we want the model can increase the number of predicting bankruptcy and it bankrupts exactly, and decrease the number of predicting non-bankruptcy but it bankrupts.

Full parameter

	Non-bankrupt	**bankrupt**
Non-bankrupt	1309	9
Bankrupt	44	1

Clearly, if we use all parameters to predict, its performance is terrible. It only predicts 1 correctly and misses 44 bankruptcy. There, this case shows that the more parameters used don't mean the better performance it has.

5 high correlation parameters with bankruptcy

	Non-bankrupt	**bankrupt**
Non-bankrupt	1316	2
Bankrupt	40	5

In this case, we select the top 5 parameters with a high correlation with "bankrupt". Its result is better than the previous one.

15 high correlation parameters with bankruptcy

	Non-bankrupt	**bankrupt**
Non-bankrupt	1308	10
Bankrupt	36	9

Here, we set the top 15 parameters with a high correlation with "bankrupt". Apparently, its result is better than the previous two.

25 high correlation parameters with bankruptcy

	Non-bankrupt	bankrupt
Non-bankrupt	1310	8
Bankrupt	36	9

Here, we set the top 25 parameters with a high correlation with "bankrupt". Apparently, its result is the same as the top 15. However, it needs to take 25 parameters. Thus, we think the top 15 is better than this one.

Ordinary Least Square

	Non-bankrupt	bankrupt
Non-bankrupt	1306	12
Bankrupt	37	8

We believe that some parameters are redundant and the ordinary least square method is a good way to remove them. In this method, we use full parameters to fit the OLS model, then we remove the parameters with their coefficient is 0. It means the model doesn't need those parameters, so we remove them. Finally, we use the remaining parameters to fit the model. This selection is better than the full one. It uses fewer parameters but obtains a better result.

Forward Stepwise

	Non-bankrupt	bankrupt
Non-bankrupt	1307	11
Bankrupt	37	8

In this case, it starts from a null model(no parameters). Then, add one by one. After adding new parameters, it will calculate its AIC. It will stop adding the new parameters until AIC increases.

Backward Stepwise

	Non-bankrupt	bankrupt
Non-bankrupt	1308	10
Bankrupt	36	9

Instead, this case starts from a full model(all parameters). Then, remove them one by one. After removing the new parameter, it will calculate its AIC. It will stop adding the new parameters until AIC increases.

Both Stepwise

	Non-bankrupt	bankrupt
Non-bankrupt	1307	11
Bankrupt	35	10

In both stepwise, it can start from a null or full model, and we start from the null model. Similarly, it may add or remove the parameters according to AIC.

The total parameter used in each selection

	Full	Cor_5	Cor_15	Cor_25	OLS	Forward	Backward	Both
Number	95	5	15	25	66	41	45	39

Here, we can see how many parameters they use. We think they have similar performance except for the top 5 correlation. However, considering the number of the parameters used, the top 15 correlation has the best performance.

Now, let's look at the previous results. We can find that their performance isn't good enough though we know which selection may take them less time to train. Therefore, we can consider their predicted probability of bankruptcy.

In general, we see it as bankruptcy if its predicted probability exceeds 50%. However, empirically, if a company is rated as CCC or below(Standard & Poor’s credit rating), its probability of default is approximate 25%. So, we would like to set the different percentages as bankruptcy probability. It means if the predicted probability exceeds this percentage, we will see it as a bankruptcy.

Accuracy according to different percentages

$Accuracy = Predicted\ Bankruptcy\ /\ Total\ Bankruptcy$

	Full	Cor_5	Cor_15	Cor_25	OLS	Forward	Backward	Both
$> 50\%$	13.33%	13.33%	17.78%	15.56%	13.33%	13.33%	17.78%	15.56%
$> 25\%$	31.11%	22.22%	40%	28.89%	26.67%	35.56%	35.56%	33.33%
$> 10\%$	64.44%	68.89%	64.44%	68.89%	64.44%	68.89%	73.33%	73.33%

As we can see, if we set 50% as the bankrupt standard, all of them are unable to predict efficiently. However, we set 25% as the bankrupt standard, the accuracy increases significantly. Furthermore, if set at 10%, the accuracy increase more and more. In fact, in the real world, the default probability of a junk bond is over 1%. So, if the predicted probability is over 10%, one should be more careful before investing. Thus, we think using the `cv.glmnet` model to estimate bankrupt probability is still efficient, especially for the top 15 correlation selection and both stepwise selection.