

# Company Bankrutcy Prediciton

Data Science Final Presentation Group 4

2022-01-13

# Group Members

Group4 :

108703014 鄭宇傑

108703019 賴冠瑜

108703029 江宗樺

108703030 田詠恩

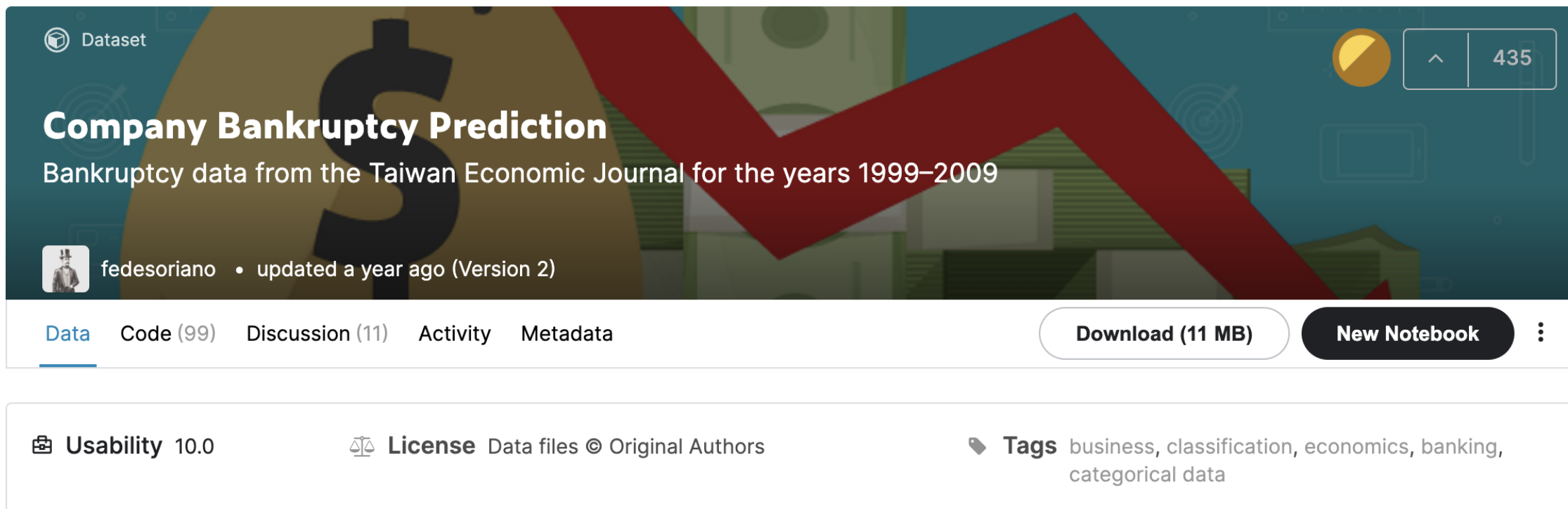
108304003 張瀚文

# Outline

# 資料介紹

# 資料介紹

1999 年至 2009 年數據來自自台灣經濟日報的統計。公司破產之定義為台灣證券交易所的業務規則而定。



The screenshot shows the Kaggle dataset page for 'Company Bankruptcy Prediction'. The header features a large background image with a red line graph and a dollar sign. The dataset title is 'Company Bankruptcy Prediction' with a subtitle 'Bankruptcy data from the Taiwan Economic Journal for the years 1999–2009'. The author is 'fedesoriano' and it was updated a year ago (Version 2). The page has tabs for 'Data', 'Code (99)', 'Discussion (11)', 'Activity', and 'Metadata'. There are buttons for 'Download (11 MB)' and 'New Notebook'. The bottom section includes 'Usability 10.0', 'License Data files © Original Authors', and 'Tags business, classification, economics, banking, categorical data'.

Dataset

## Company Bankruptcy Prediction

Bankruptcy data from the Taiwan Economic Journal for the years 1999–2009

fedesoriano • updated a year ago (Version 2)

Data Code (99) Discussion (11) Activity Metadata

Download (11 MB) New Notebook

Usability 10.0 License Data files © Original Authors

Tags business, classification, economics, banking, categorical data

# 屬性資訊

- X2 - ROA(A) before interest and % after tax: Return On Total Assets(A)
- X11 - Operating Expense Rate: Operating Expenses/Net Sales
- X13 - Cash flow rate: Cash Flow from Operating/Current Liabilities
- X33 - Current Ratio
- X92 - Degree of Financial Leverage (DFL)
- X85 - Liability-Assets Flag: 1 if Total Liability exceeds Total Assets, 0 otherwise
- X94 - Net Income Flag: 1 if Net Income is Negative for the last two years, 0 otherwise

# 資料分析

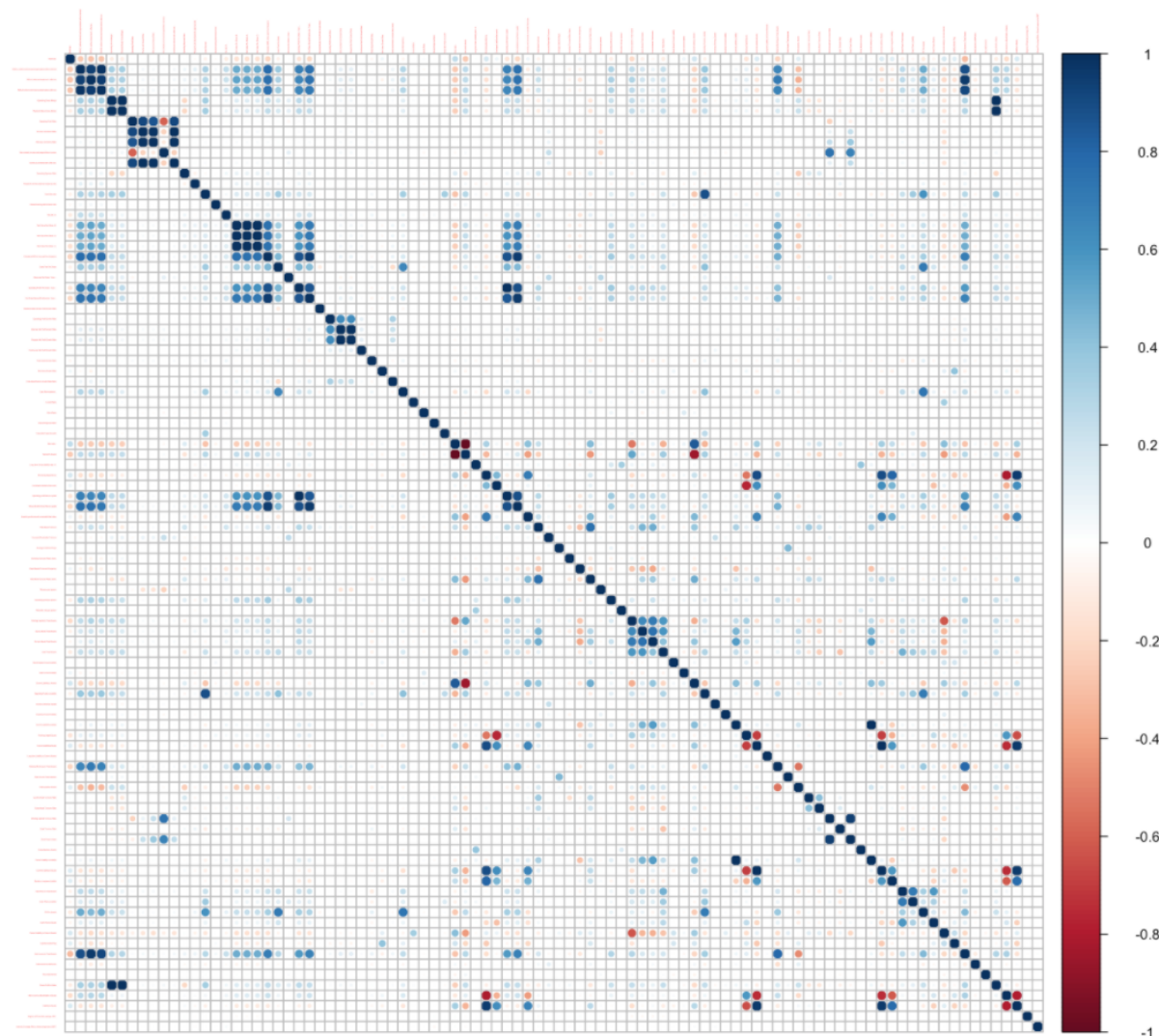
# Shiny App





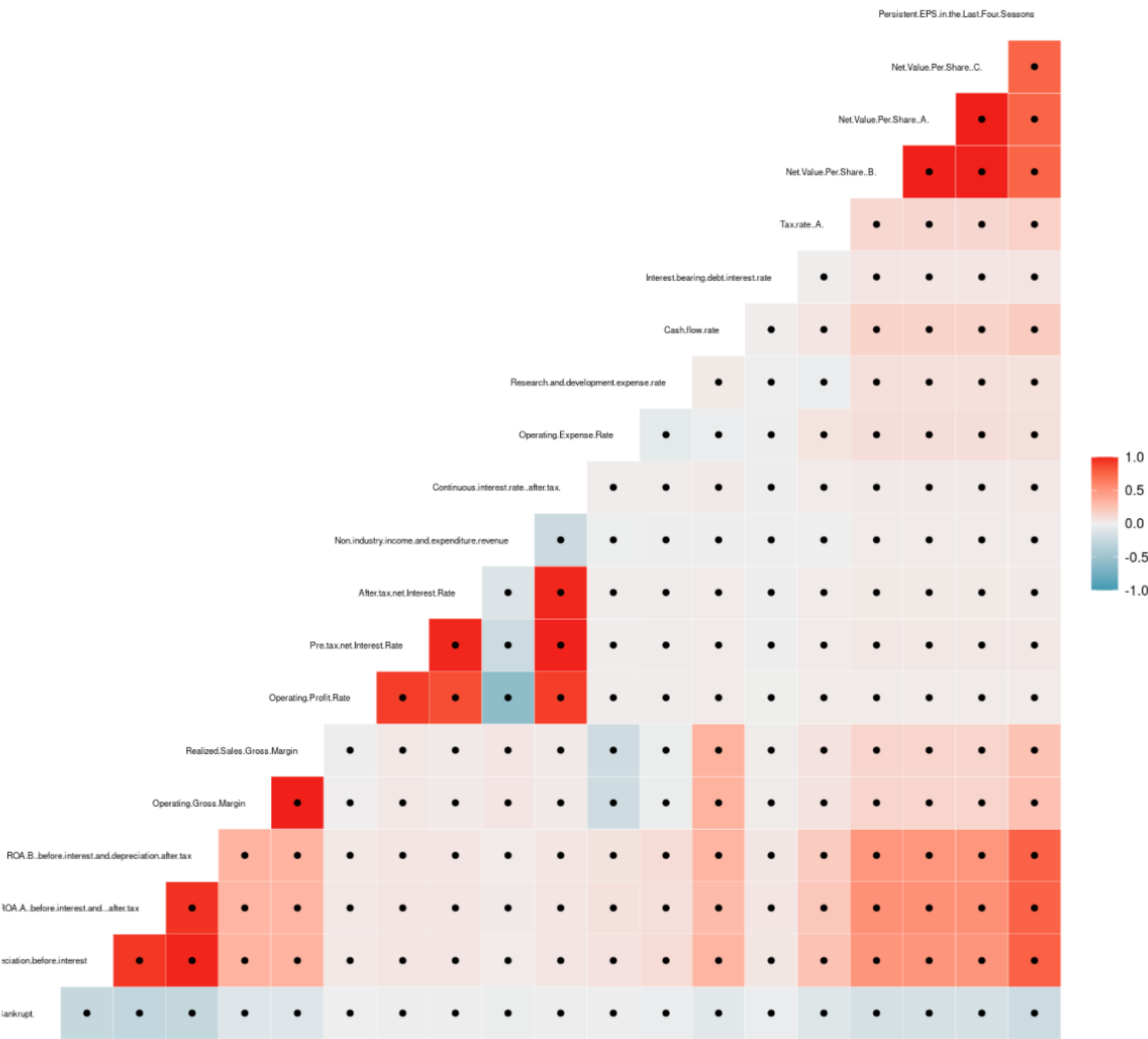
# 相關係數

該圖顯示了每個特徵對於對方的相關性。圖中 x 及 y 軸為資料的屬性，格子中的顏色越深，代表著兩屬性之間的相關性越高。



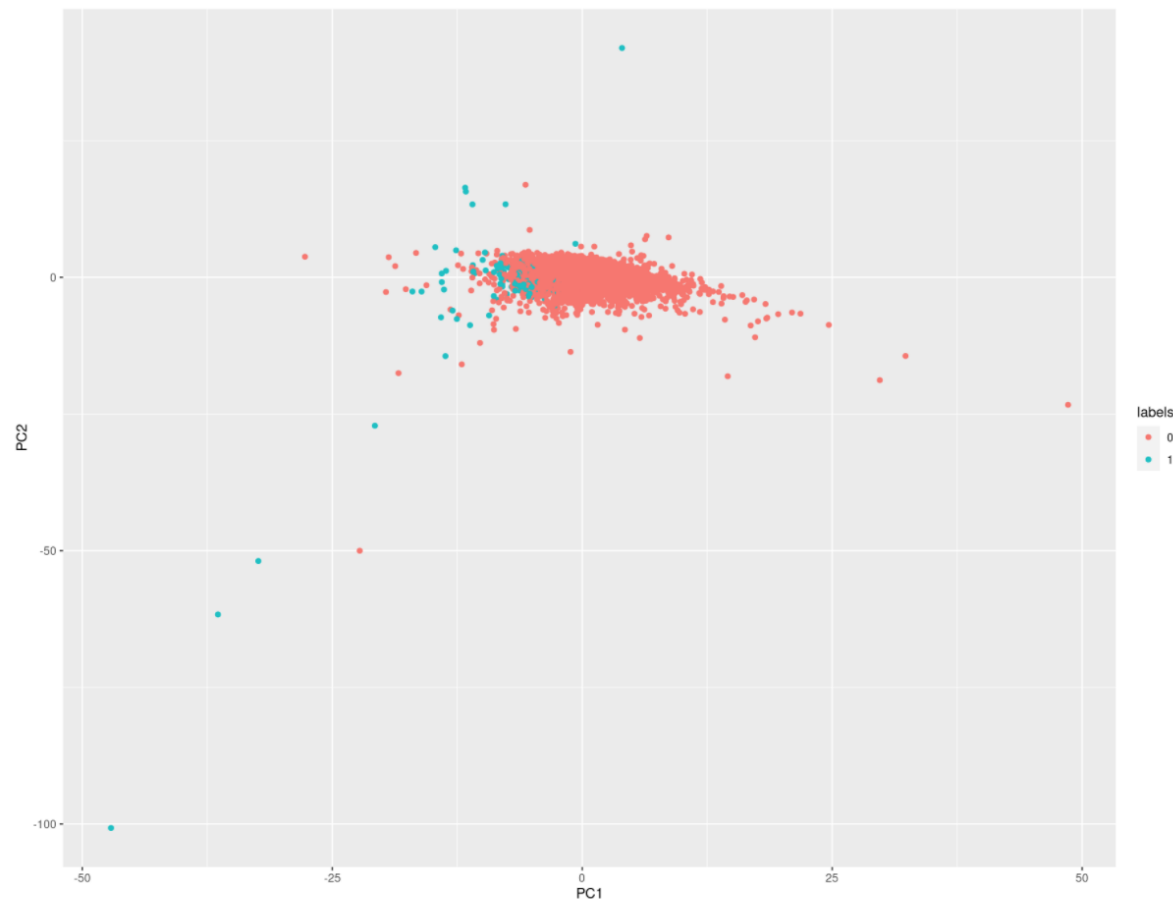
# 相關係數

該圖顯示了每個特徵對於對方的相關性。圖中 x 及 y 軸為資料的屬性，格子中的顏色越深，代表著兩屬性之間的相關性越高。



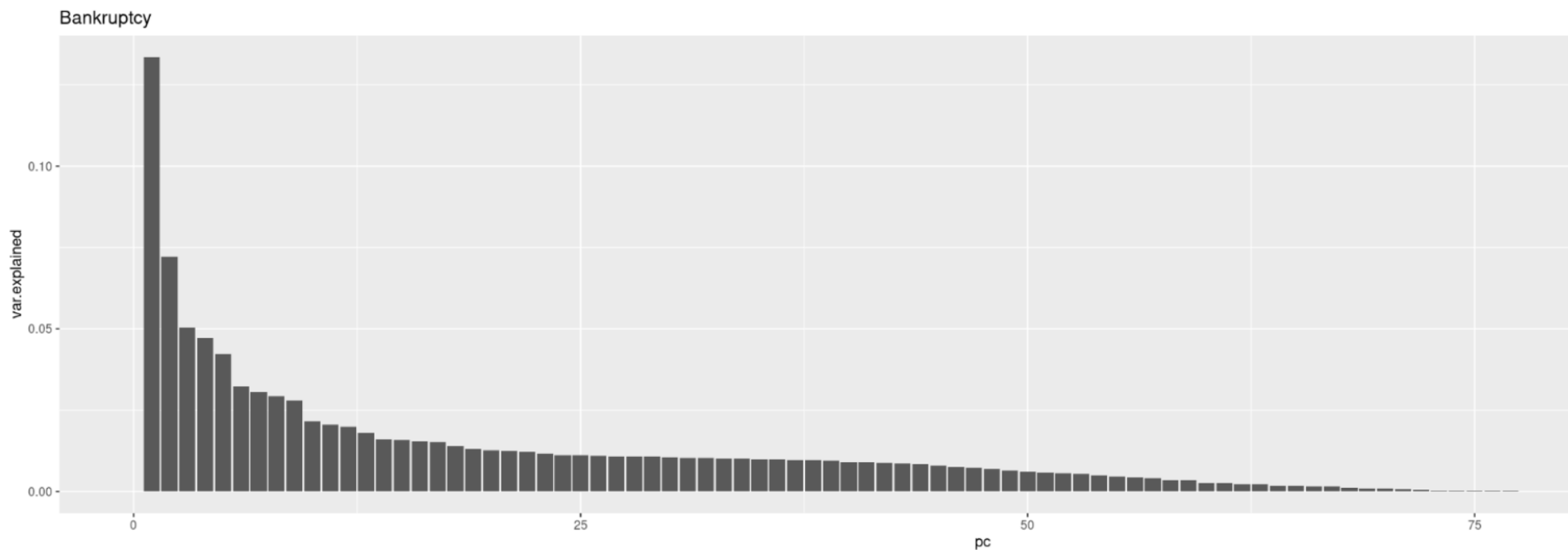
# 主成份分析

從第一主成份(PC1)到第二主成份(PC2)可以發現沒有明顯可以分辨破產與否的成分。

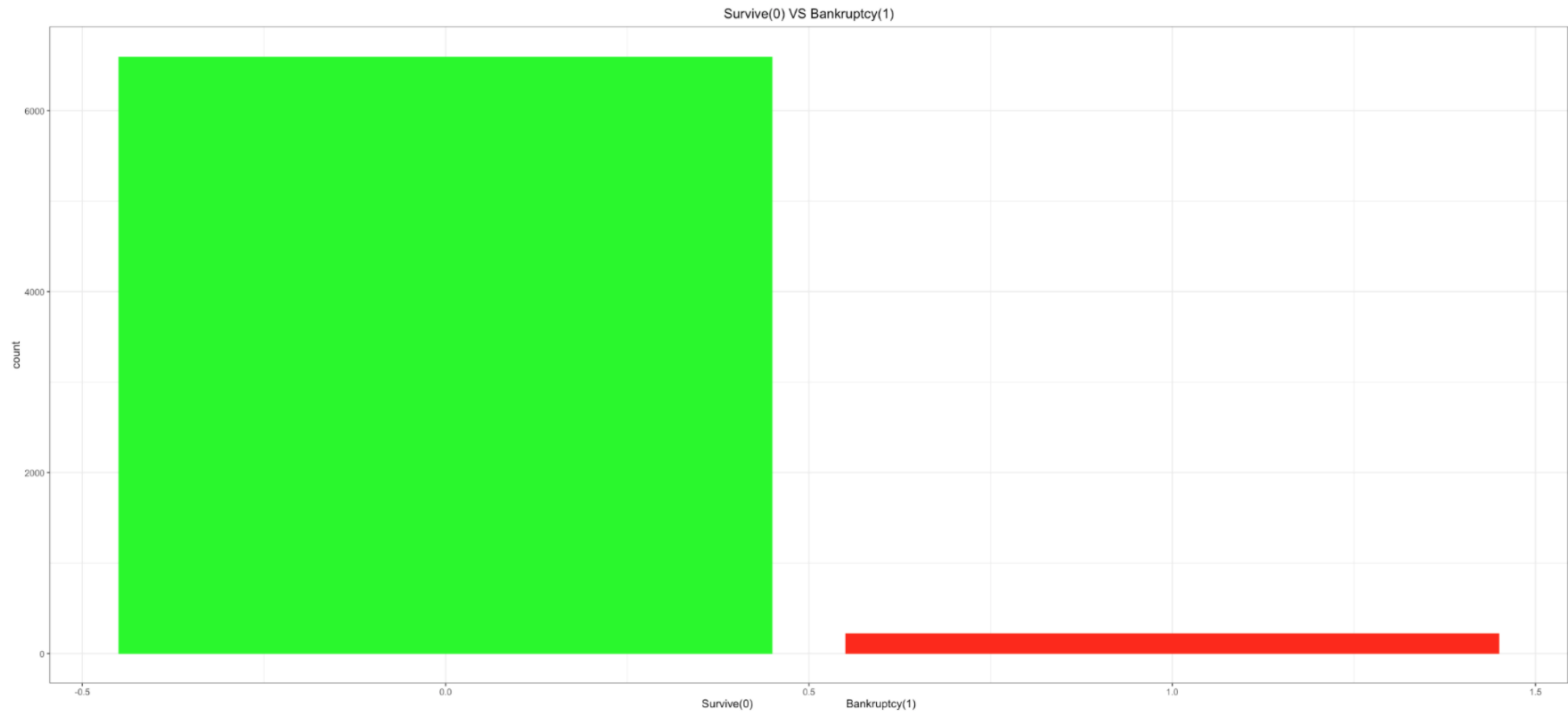


# 資料解釋度

我們可以看到大約 40 個主成份就解釋了超過 90% 的差異。



# 公司破產分佈



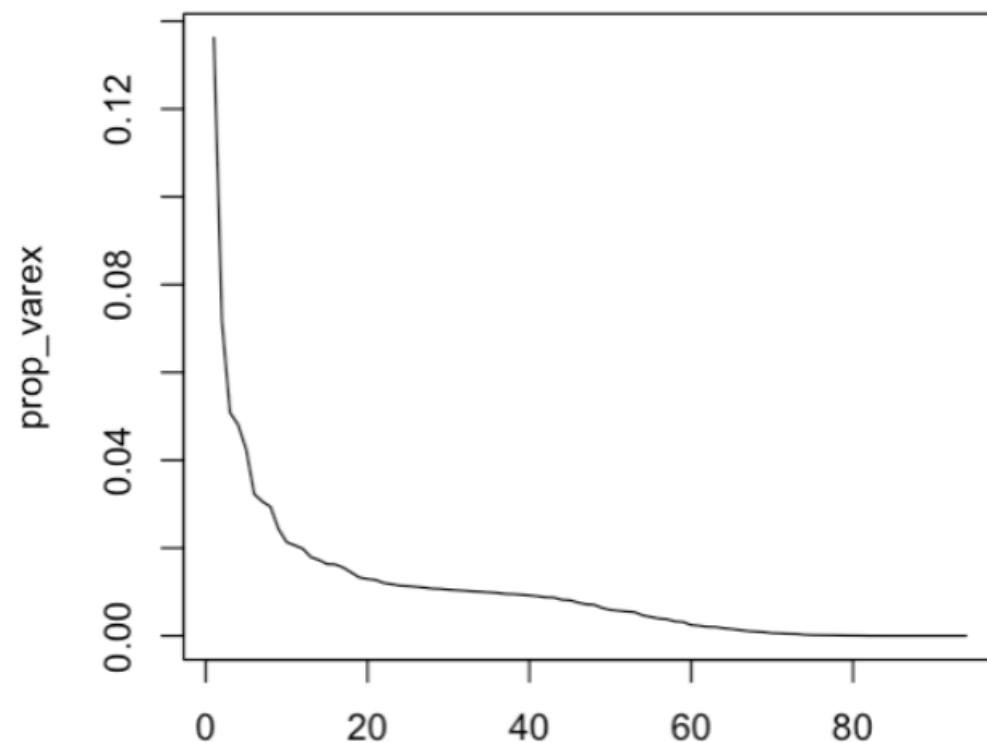
# 模型評估標準

- 95% 以上的資料中的公司都沒有破產(Bankruptcy == 0)  
所以全部猜 1 就可以有超級高的 Accuracy (NULL model)
- 我們將目標設定成要盡可能增加 recall 。  
嘗試預測出更多可能會倒的公司去對他們做關切或提早做應對措施，並去檢視可能面臨的問題，是這次專題的主要目標。

# Models

## pca 分析 -> 取前 40 個

```
#--- watch pc  
std_dev <- pca$sdev  
pr_var <- std_dev^2  
prop_varex <- pr_var/sum(pr_var)  
plot(prop_varex, type = 'lines')
```





## model #1: rpart decision tree

decision tree

```
prediction
truth      0      1
0 1303      15
1   39       7
```

decision tree with pca

```
pred
truth      0      1
0 1306      15
1   32      11
```

```
> |
```

## model #1: rpart decision tree

decision tree	decision tree with pca
<pre>&gt; print( accuracy ) [1] 0.9604106 &gt; print( recall ) [1] 0.1521739 &gt; print( precision ) [1] 0.3181818 &gt; print( NegativePrecision ) [1] 0.9709389</pre>	<pre>&gt; print( accuracy ) [1] 0.9655425 &gt; print( recall ) [1] 0.255814 &gt; print( precision ) [1] 0.4230769 &gt; print( NegativePrecision ) [1] 0.9760837</pre>

## model #2: random forest

random forest	decision tree with pca
<pre>prediction truth    0    1 0 1301    16 1   30    17</pre>	<pre>prediction truth    0    1 0 1297    20 1   40     7</pre>

## model #2: random forest

### random forest

```
> print( accuracy )  
[1] 0.9662757  
> print( recall )  
[1] 0.3617021  
> print( precision )  
[1] 0.5151515  
> print( NegativePrecision )  
[1] 0.9774606
```

### decision tree with pca

```
> print( accuracy )  
[1] 0.9560117  
> print( recall )  
[1] 0.1489362  
> print( precision )  
[1] 0.2592593  
> print( NegativePrecision )  
[1] 0.9700823
```

## model #3: logistic regression

logistic regression

```
prediction
truth    0    1
0 1306   17
1   28   13
```

logistic regression with pca

```
pred
truth    0    1
0 1237   86
1   15   26
```

## model #3: logistic regression

### logistic regression

```
> print( accuracy )  
[1] 0.9670088  
> print( recall )  
[1] 0.3170732  
> print( precision )  
[1] 0.4333333  
> print( NegativePrecision )  
[1] 0.9790105
```

### logistic regression with pca

```
> print( accuracy )  
[1] 0.9259531  
> print( recall )  
[1] 0.6341463  
> print( precision )  
[1] 0.2321429  
> print( NegativePrecision )  
[1] 0.9880192
```

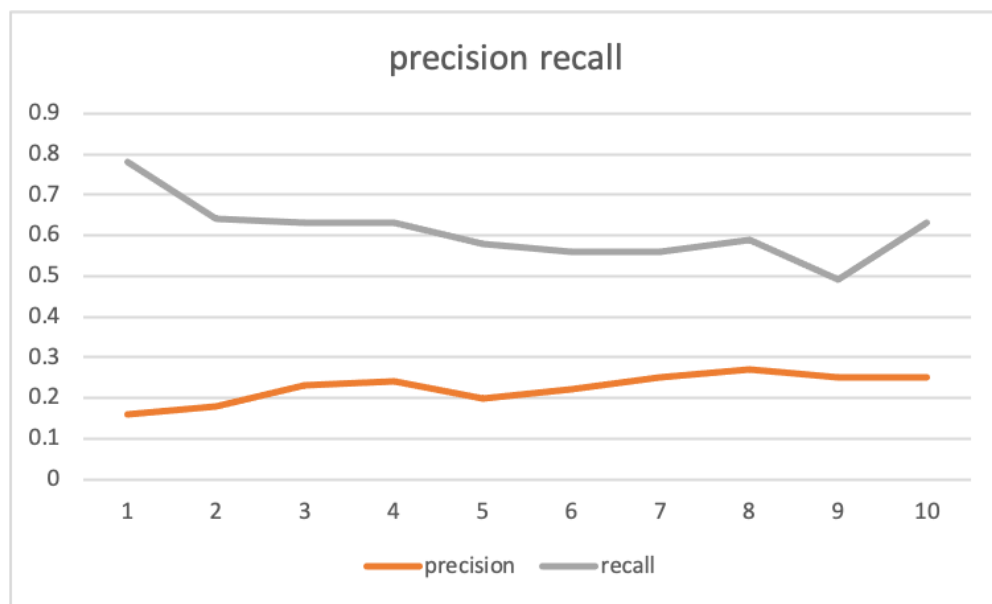
# SMOTE 生成資料

# Pytorch

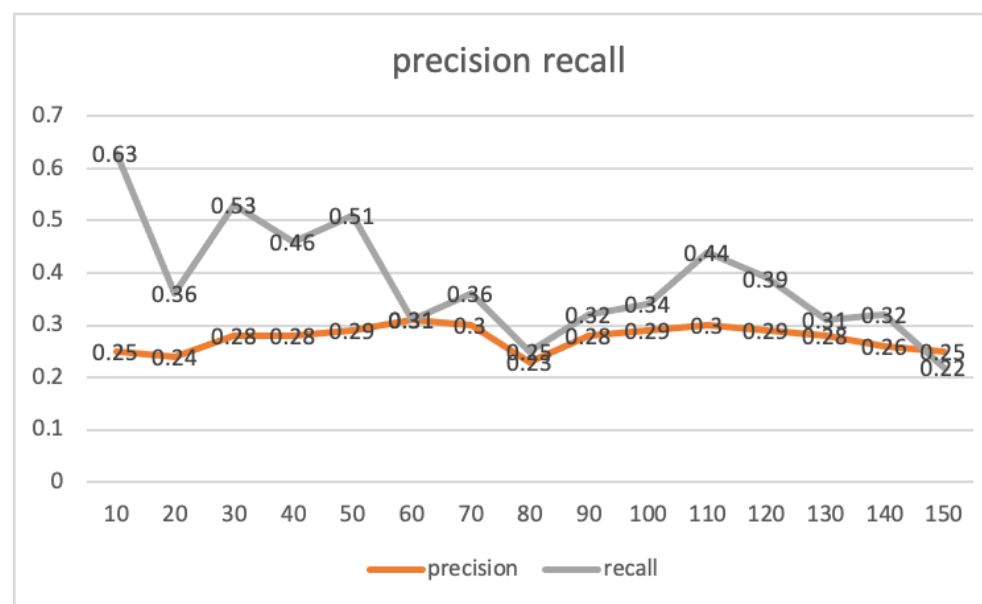


# epoch 次數比較

epoch 1-10



epoch 10-150



## 總結

recall	原始資料	透過 PCA 降維處理
Decision Tree		
Random Forest		
Logistic Regression		
Convolution Neural Network		

# 問題與討論

- 資料面？
  - 資料的收集
  - 資料的真實性
- 模型面？
  - 資料數量
  - 對未來預測的時效性

# Reference

<https://www.kaggle.com/jerryfang5/bankrutcy-prediciton-by-r/notebook>

<https://www.kaggle.com/seongwonr/bankruptcy-prediction-with-smote>

[https://colab.research.google.com/drive/12wXAyrbX8Ji5J6CNAEIQwtDOaxy8BCIO?  
usp=sharing](https://colab.research.google.com/drive/12wXAyrbX8Ji5J6CNAEIQwtDOaxy8BCIO?usp=sharing)