一、研究目的

利用財務特徵(X1-X95)預測公司是否破產(Y)，比較不同的機器學習模型，若Precision,Recall,Accuracy若未達0.9，則發展衍生變數使模式具有良好的預測能力。

二、使用技術方法

改善目標類別不均衡問題：向下取樣(RandomUnderSampler)。
機器學習模型：決策樹、隨機森林、羅吉斯回歸、KNN。
評估模型正確率：混淆矩陣。

三、分析結果

- **原始資料的混淆矩陣**

  自變數X：X1~X95，共95個自變數。
  依變數y：Bankrupt?，共1個依變數。
  訓練集：測試集=75：25 & 90：10。

- **向下取樣的混淆矩陣**

  自變數X：X1~X95，共95個自變數。
  依變數y：Bankrupt?，共1個依變數。
  改善目標類別不均衡問題：向下取樣(RandomUnderSampler)。
  訓練集：測試集=75：25 & 90：10。
  備註：只有將測試集向下取樣，訓練集沒有向下取樣(採原資料集)。

  此資料集在「Bankrupt?」欄位中比例懸殊太大，會造成混淆矩陣有過度擬合(overfitting)的問題，故為了降低此問題，使用向下取樣來解決此問題。

Bankrupt?	Frequence_是否破產	Percentage_是否破產
0.0	6599	0.968
1.0	220	0.032

備註：Bankrupt?=0.0→沒破產，Bankrupt?=1.0→有破產。

- **全部的混淆矩陣**
  - 決策樹（max_depth=2）

    訓練集：測試集=75：25和90：10在經過資料處理和向下取樣後的混淆矩陣recall值皆有到1.0的準度。

    若要使用決策樹模型，會建議使用向下取樣後的模型，且訓練集：測試集=75：25或90：10。

<table>
<tr><td colspan="2">混淆矩陣</td></tr>
<tr><td>訓練</td><td>測試</td></tr>
<tr><td colspan="2">未處理</td></tr>
<tr><td colspan="2">訓練集：測試集=75：25</td></tr>
<tr><td>

```
[[4941    0]
 [ 173    0]]
```

	precision	recall	f1-score	support
0.0	0.97	1.00	0.98	4941
1.0	0.00	0.00	0.00	173
accuracy			0.97	5114
macro avg	0.48	0.50	0.49	5114
weighted avg	0.93	0.97	0.95	5114

</td><td>

```
[[1658    0]
 [  47    0]]
```

	precision	recall	f1-score	support
0.0	0.97	1.00	0.99	1658
1.0	0.00	0.00	0.00	47
accuracy			0.97	1705
macro avg	0.49	0.50	0.49	1705
weighted avg	0.95	0.97	0.96	1705

</td></tr>
<tr><td colspan="2">訓練集：測試集=90：10</td></tr>
<tr><td>

```
[[5923   14]
 [ 171   29]]
```

	precision	recall	f1-score	support
0.0	0.97	1.00	0.98	5937
1.0	0.67	0.14	0.24	200
accuracy			0.97	6137
macro avg	0.82	0.57	0.61	6137
weighted avg	0.96	0.97	0.96	6137

</td><td>

```
[[659   3]
 [ 17   3]]
```

	precision	recall	f1-score	support
0.0	0.97	1.00	0.99	662
1.0	0.50	0.15	0.23	20
accuracy			0.97	682
macro avg	0.74	0.57	0.61	682
weighted avg	0.96	0.97	0.96	682

</td></tr>
<tr><td colspan="2">已處理</td></tr>
<tr><td colspan="2">訓練集：測試集=75：25</td></tr>
</table>

[[576 0] [0 173]] ========== precision recall f1-score support 0.0 1.00 1.00 1.00 576 1.0 1.00 1.00 1.00 173 accuracy 1.00 749 macro avg 1.00 1.00 1.00 749 weighted avg 1.00 1.00 1.00 749	[[1658 0] [0 47]] ========== precision recall f1-score support 0.0 1.00 1.00 1.00 1658 1.0 1.00 1.00 1.00 47 accuracy 1.00 1705 macro avg 1.00 1.00 1.00 1705 weighted avg 1.00 1.00 1.00 1705
訓練集：測試集=90：10	
[[666 0] [0 200]] ========== precision recall f1-score support 0.0 1.00 1.00 1.00 666 1.0 1.00 1.00 1.00 200 accuracy 1.00 866 macro avg 1.00 1.00 1.00 866 weighted avg 1.00 1.00 1.00 866	[[662 0] [0 20]] ========== precision recall f1-score support 0.0 1.00 1.00 1.00 662 1.0 1.00 1.00 1.00 20 accuracy 1.00 682 macro avg 1.00 1.00 1.00 682 weighted avg 1.00 1.00 1.00 682

- 隨機森林（n_estimators=25）

  訓練集：測試集=75：25和90：10在經過向下取樣後的混淆矩陣recall值皆有到大於0.9的準度。

  若要使用決策樹模型，會建議使用向下取樣後的模型，且訓練集：測試集=75：25或90：10。

混淆矩陣	
訓練	測試
未處理	
訓練集：測試集=75：25	
[[4941 0] [5 168]] ========== precision recall f1-score support 0.0 1.00 1.00 1.00 4941 1.0 1.00 0.97 0.99 173 accuracy 1.00 5114 macro avg 1.00 0.99 0.99 5114 weighted avg 1.00 1.00 1.00 5114	[[1653 5] [40 7]] ========== precision recall f1-score support 0.0 0.98 1.00 0.99 1658 1.0 0.58 0.15 0.24 47 accuracy 0.97 1705 macro avg 0.78 0.57 0.61 1705 weighted avg 0.97 0.97 0.97 1705
訓練集：測試集=90：10	

<table>
<tr><td>

```
[[5937    0]
 [   5  195]]
```

	precision	recall	f1-score	support
0.0	1.00	1.00	1.00	5937
1.0	1.00	0.97	0.99	200
accuracy			1.00	6137
macro avg	1.00	0.99	0.99	6137
weighted avg	1.00	1.00	1.00	6137

</td><td>

```
[[661   1]
 [ 19   1]]
```

	precision	recall	f1-score	support
0.0	0.97	1.00	0.99	662
1.0	0.50	0.05	0.09	20
accuracy			0.97	682
macro avg	0.74	0.52	0.54	682
weighted avg	0.96	0.97	0.96	682

</td></tr>
<tr><td colspan="2">已處理</td></tr>
<tr><td colspan="2">訓練集：測試集=75：25</td></tr>
<tr><td>

```
[[576   0]
 [  0 173]]
```

	precision	recall	f1-score	support
0.0	1.00	1.00	1.00	576
1.0	1.00	1.00	1.00	173
accuracy			1.00	749
macro avg	1.00	1.00	1.00	749
weighted avg	1.00	1.00	1.00	749

</td><td>

```
[[1656    2]
 [   0   47]]
```

	precision	recall	f1-score	support
0.0	1.00	1.00	1.00	1658
1.0	0.96	1.00	0.98	47
accuracy			1.00	1705
macro avg	0.98	1.00	0.99	1705
weighted avg	1.00	1.00	1.00	1705

</td></tr>
<tr><td colspan="2">訓練集：測試集=90：10</td></tr>
<tr><td>

```
[[666   0]
 [  0 200]]
```

	precision	recall	f1-score	support
0.0	1.00	1.00	1.00	666
1.0	1.00	1.00	1.00	200
accuracy			1.00	866
macro avg	1.00	1.00	1.00	866
weighted avg	1.00	1.00	1.00	866

</td><td>

```
[[661   1]
 [  2  18]]
```

	precision	recall	f1-score	support
0.0	1.00	1.00	1.00	662
1.0	0.95	0.90	0.92	20
accuracy			1.00	682
macro avg	0.97	0.95	0.96	682
weighted avg	1.00	1.00	1.00	682

</td></tr>
</table>

- 羅吉斯回歸（solver='newton-cg'）
  訓練集：測試集=75：25和90：10在經過向下取樣後的混淆矩陣皆有提高數值，但成果仍然不理想，所以不會建議使用羅吉斯回歸模型。

混淆矩陣	
訓練	測試
未處理	
訓練集：測試集=75：25	

```
[[4926   15]
 [ 169    4]]
```

	precision	recall	f1-score	support
0.0	0.97	1.00	0.98	4941
1.0	0.21	0.02	0.04	173
accuracy			0.96	5114
macro avg	0.59	0.51	0.51	5114
weighted avg	0.94	0.96	0.95	5114

```
[[1647   11]
 [  47    0]]
```

	precision	recall	f1-score	support
0.0	0.97	0.99	0.98	1658
1.0	0.00	0.00	0.00	47
accuracy			0.97	1705
macro avg	0.49	0.50	0.49	1705
weighted avg	0.95	0.97	0.96	1705

訓練集：測試集=90：10

```
[[5921   16]
 [ 195    5]]
```

	precision	recall	f1-score	support
0.0	0.97	1.00	0.98	5937
1.0	0.24	0.03	0.05	200
accuracy			0.97	6137
macro avg	0.60	0.51	0.51	6137
weighted avg	0.94	0.97	0.95	6137

```
[[658   4]
 [ 20   0]]
```

	precision	recall	f1-score	support
0.0	0.97	0.99	0.98	662
1.0	0.00	0.00	0.00	20
accuracy			0.96	682
macro avg	0.49	0.50	0.49	682
weighted avg	0.94	0.96	0.95	682

已處理

訓練集：測試集=75：25

```
[[551  25]
 [145  28]]
```

	precision	recall	f1-score	support
0.0	0.79	0.96	0.87	576
1.0	0.53	0.16	0.25	173
accuracy			0.77	749
macro avg	0.66	0.56	0.56	749
weighted avg	0.73	0.77	0.72	749

```
[[1591   67]
 [  42    5]]
```

	precision	recall	f1-score	support
0.0	0.97	0.96	0.97	1658
1.0	0.07	0.11	0.08	47
accuracy			0.94	1705
macro avg	0.52	0.53	0.53	1705
weighted avg	0.95	0.94	0.94	1705

訓練集：測試集=90：10

```
[[638  28]
 [175  25]]
```

	precision	recall	f1-score	support
0.0	0.78	0.96	0.86	666
1.0	0.47	0.12	0.20	200
accuracy			0.77	866
macro avg	0.63	0.54	0.53	866
weighted avg	0.71	0.77	0.71	866

```
[[642  20]
 [ 19   1]]
```

	precision	recall	f1-score	support
0.0	0.97	0.97	0.97	662
1.0	0.05	0.05	0.05	20
accuracy			0.94	682
macro avg	0.51	0.51	0.51	682
weighted avg	0.94	0.94	0.94	682

- KNN（n_neighbors=3）

  訓練集：測試集=75：25和90：10在經過向下取樣後的混淆矩陣皆有提高數值，但成果仍然不理想，所以不會建議使用KNN模型。

<table>
<tr><td colspan="2">混淆矩陣</td></tr>
<tr><td>訓練</td><td>測試</td></tr>
<tr><td colspan="2">未處理</td></tr>
<tr><td colspan="2">訓練集：測試集=75：25</td></tr>
<tr><td>

```
[[4935    6]
 [ 149   24]]
```

	precision	recall	f1-score	support
0.0	0.97	1.00	0.98	4941
1.0	0.80	0.14	0.24	173
accuracy			0.97	5114
macro avg	0.89	0.57	0.61	5114
weighted avg	0.96	0.97	0.96	5114

</td><td>

```
[[1648   10]
 [  46    1]]
```

	precision	recall	f1-score	support
0.0	0.97	0.99	0.98	1658
1.0	0.09	0.02	0.03	47
accuracy			0.97	1705
macro avg	0.53	0.51	0.51	1705
weighted avg	0.95	0.97	0.96	1705

</td></tr>
<tr><td colspan="2">訓練集：測試集=90：10</td></tr>
<tr><td>

```
[[5932    5]
 [ 168   32]]
```

	precision	recall	f1-score	support
0.0	0.97	1.00	0.99	5937
1.0	0.86	0.16	0.27	200
accuracy			0.97	6137
macro avg	0.92	0.58	0.63	6137
weighted avg	0.97	0.97	0.96	6137

</td><td>

```
[[659   3]
 [ 20   0]]
```

	precision	recall	f1-score	support
0.0	0.97	1.00	0.98	662
1.0	0.00	0.00	0.00	20
accuracy			0.97	682
macro avg	0.49	0.50	0.49	682
weighted avg	0.94	0.97	0.95	682

</td></tr>
<tr><td colspan="2">已處理</td></tr>
<tr><td colspan="2">訓練集：測試集=75：25</td></tr>
<tr><td>

```
[[559  17]
 [ 77  96]]
```

	precision	recall	f1-score	support
0.0	0.88	0.97	0.92	576
1.0	0.85	0.55	0.67	173
accuracy			0.87	749
macro avg	0.86	0.76	0.80	749
weighted avg	0.87	0.87	0.86	749

</td><td>

```
[[1470  188]
 [  37   10]]
```

	precision	recall	f1-score	support
0.0	0.98	0.89	0.93	1658
1.0	0.05	0.21	0.08	47
accuracy			0.87	1705
macro avg	0.51	0.55	0.51	1705
weighted avg	0.95	0.87	0.91	1705

</td></tr>
<tr><td colspan="2">訓練集：測試集=90：10</td></tr>
</table>

```
[[634  32]
 [ 90 110]]
==================================================

              precision    recall  f1-score   support

         0.0       0.88      0.95      0.91       666
         1.0       0.77      0.55      0.64       200

    accuracy                           0.86       866
   macro avg       0.83      0.75      0.78       866
weighted avg       0.85      0.86      0.85       866
```

```
[[589  73]
 [ 15   5]]
==================================================

              precision    recall  f1-score   support

         0.0       0.98      0.89      0.93       662
         1.0       0.06      0.25      0.10        20

    accuracy                           0.87       682
   macro avg       0.52      0.57      0.52       682
weighted avg       0.95      0.87      0.91       682
```

四、結論

各項數值最高最優良的模型為有處理過資料的決策樹(max_depth=2)模型，再來為有處理過資料的隨機森林(n_estimators=25)模型。

排名	優良度比較
1	決策樹(max_depth=2,訓練集：測試集=75：25)
	||
1	決策樹(max_depth=2,訓練集：測試集=90：10)
	V
3	隨機森林(n_estimators=25,訓練集：測試集=75：25)
	V
4	隨機森林(n_estimators=25,訓練集：測試集=90：10)

其他模型在經過資料處理後大多都有提高混淆矩陣的數值，但成果仍然不理想，所以不會建議使用。