Stroke Prediction Deep Learning Project

Shao Yan Chen 2021/12

Outline

- Introduction
- Related Work
- Method Description
- Experimental Results

Introduction

Stroke is the 2nd leading cause of death globally

- Stroke(台灣稱中風)，根據WHO於2019年所提供的統計報告，它是全球最致命的疾病之一。
- 其成因與突發性腦血管阻塞、破裂有關連。
- 全球每年有約11%(620萬人)死於中風。

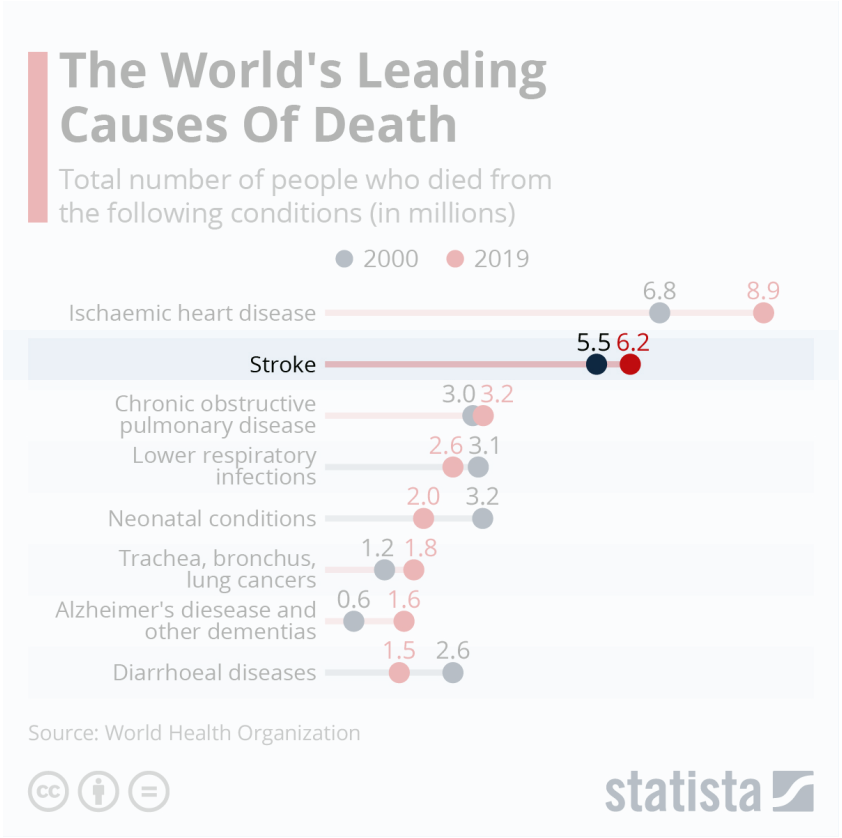
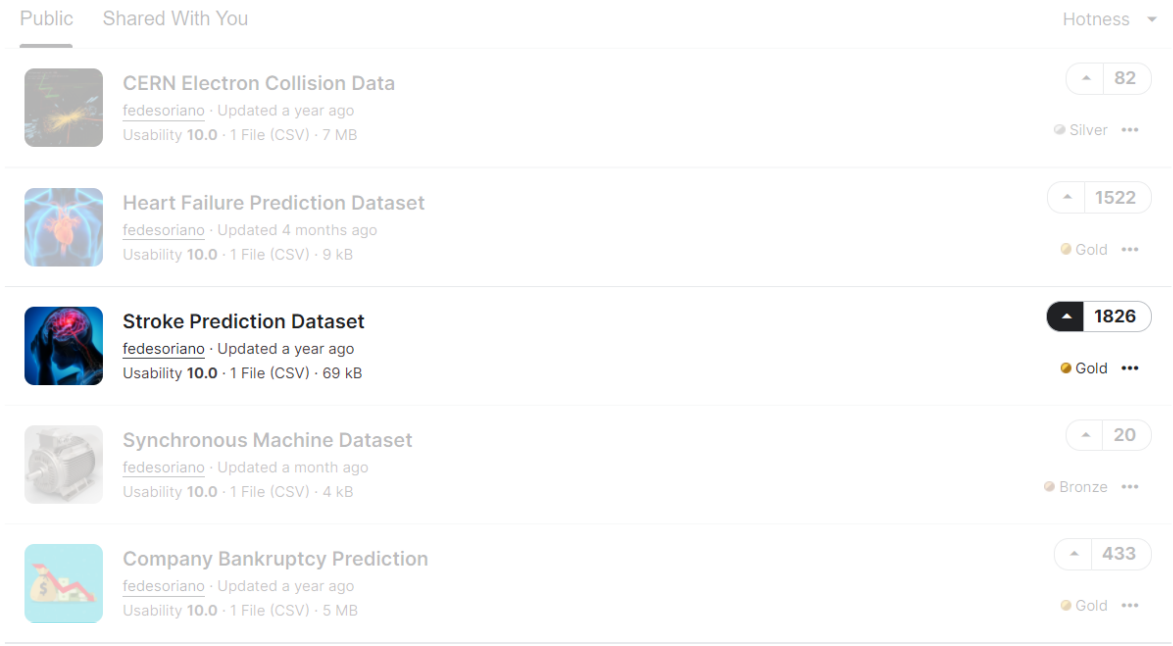


image source: https://www.statista.com/chart/23755/total-number-of-people-who-died-from-the-following-conditions/

So we want to know whether a patient is probably to get stroke based on some features

- 本次從Kaggle上找了一筆資料集，進行建模預測病患是否會得中風。
- 資料提供者為**fedesoriano**，其在Kaggle上提供多筆實用性高的資料。
- 本次的Stroke Prediction Dataset資料集為其提供的多筆醫療相關資料之一。



screenshot source：https://www.kaggle.com/fedesoriano/datasets

And these are some information about the dataset

- 此組資料有5110筆樣本，12個變數。
- 12個變數：
  1. id
  2. gender
  3. age
  4. hypertension
  5. heart_disease
  6. ever_married
  7. work_type
  8. residence_type
  9. avg_glucose_level
  10. bmi
  11. smoking_status
  12. stroke (target)

Related Work

An integrated machine learning approach to stroke prediction(2010)

- 使用Cardiovascular Health Study (CHS)公開的資料集做訓練。
- 使用SVM建模
- L1 norm & Cox (統計方法)

Interpretable classifiers using rules and Bayesian analysis: Building a better stroke prediction model(2015)

- 透過關聯模型：Bayesian Rule Lists建立預測模型。
- 最後得出當時最佳準確率模型。

Comparing Deep Neural Network and Other Machine Learning Algorithms for Stroke Prediction in a Large-Scale Population-Based Electronic Medical Claims Database(2017)

- 比較DNN、GBDT、LR、SVM在預測stoke上的差異。
- **DNN模型設定：每層neurons和input dim一樣多，使用tanh作為Activation function，初始值隨機，使用SGD訓練，每層做batch normalize。**
- 結果：DNN和GBDT並列最佳預測率，但**DNN用較少資料就能達到最佳預測率**。

Method Description

Before Analysis：Data preprocessing

- 資料中有缺失值，從統計上的考量，以MICE方法進行補值。
- 透過EDA檢查變數關聯、找出Insights：變數間關聯小，初步判斷無顯著變數影響Stroke。
- 多數為類別資料且無順序關係，以one-hot encoding處理。
- 將數值資料標準化。
- 切割資料0.8訓練集0.2測試集，其中訓練集切分0.2為驗證集。

These are some models we used in this project

- Statistics model：Logistic Regression
- Machine Learning：KNN、SVM、Decision Tree、Random Forest
- Deep Learning：DNN

How we set the DNN model

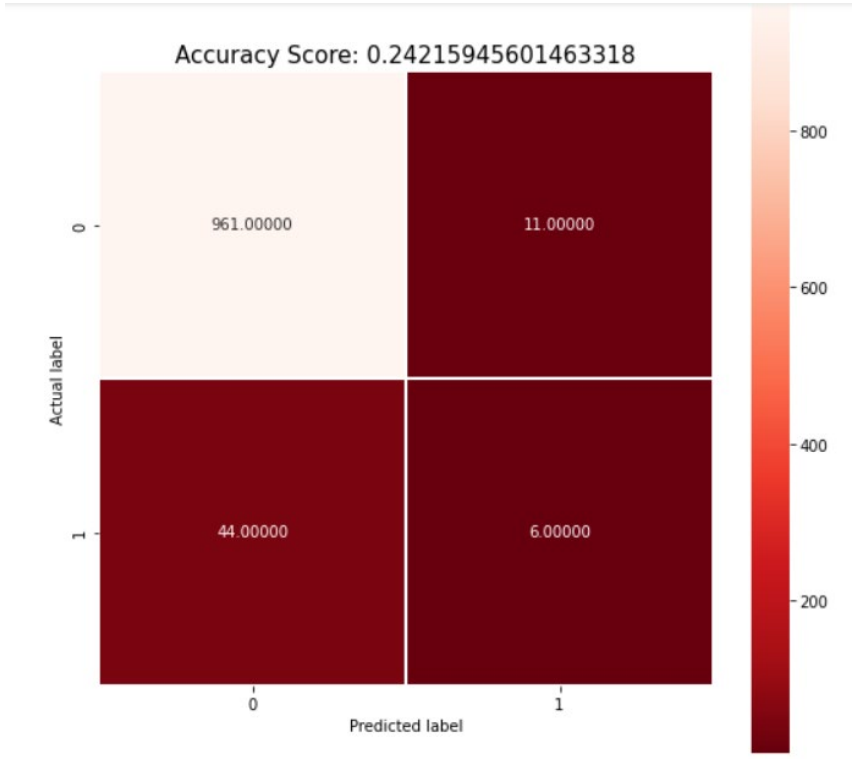
- 一開始先以最簡單的模型開始：
- Hidden Layers：5層，神經元數量從256遞減。
- Activation function: Relu
- Output: sigmoid
- Optimizer = SGD, lr = 0.0005
- Loss: binary crossentropy
- Metrics: accuracy
- Batch size = 256
- Epochs: = 500
- Validation: 0.2

And we tried different model settings

- [5 Denses, Relu, sigmoid, SGD, lr = 0.0005, binary crossentropy]
- [5 Denses, Relu, sigmoid, SGD, lr = 0.0005, binary crossentropy, **dropout = 0.4**]
- [6 Denses, Relu, sigmoid, SGD, lr = 0.0005, binary crossentropy, **Batch normalization**]
- [6 Denses, **tanh**, sigmoid, SGD, lr = 0.0005, binary crossentropy, **Batch normalization**]
- [6 Denses, **tanh**, sigmoid, **Adam**, binary crossentropy, **Batch normalization**]

We got 0.946 accuracy and 0.242 score on test data

- 這個模型的結果看起來不錯，但透過混淆矩陣發現了一個大問題。
- 或許是資料不平衡的關係(沒中風的患者佔絕大多數)，導致模型難以對中風患者進行預測➔使用**SMOTE**調整資料。

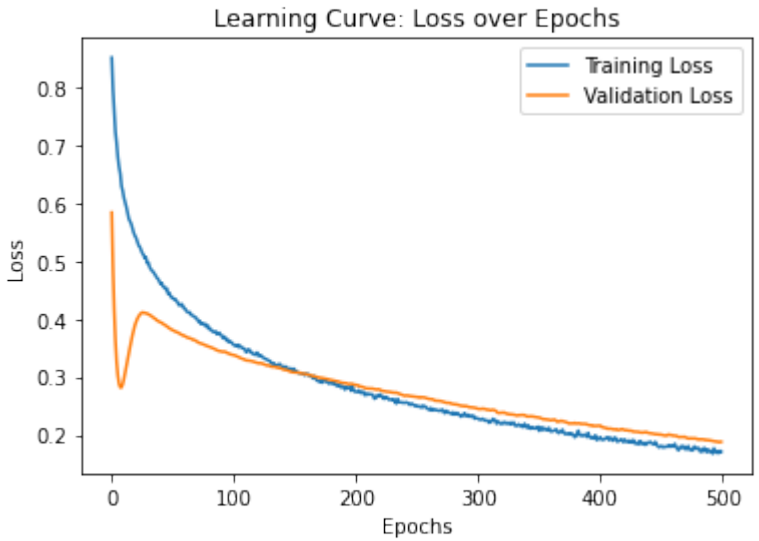


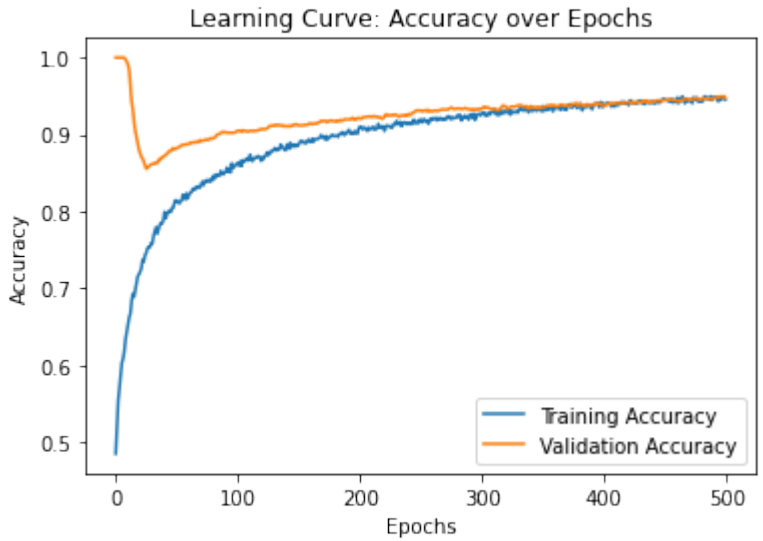
DNN with SMOTE data

- 資料已經過SMOTE處理。
- Hidden layers: 6 Dense, each with bathnormalization.
- Activation function: Relu
- Output layer: sigmoid
- Optimizer: SGD, lr = 0.0005
- Loss: binary crossentropy
- Metrics: accuracy
- Batch size = 256
- Epochs = 500
- Validation = 0.2

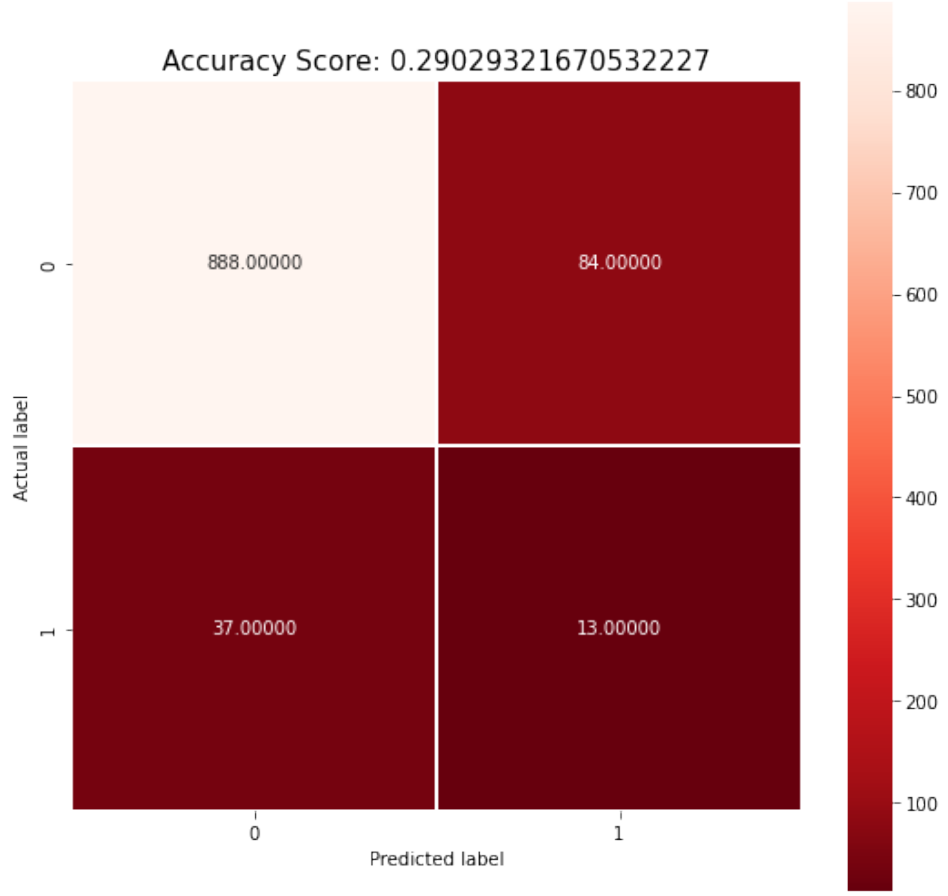
This time we got accuracy 0.881, score 0.290

- [6 Denses, Relu, sigmoid, SGD, lr = 0.0005, binary crossentropy, Batch normalization]
- 本次學習曲線平滑，後期穩定，但預測率比上個模型稍低。



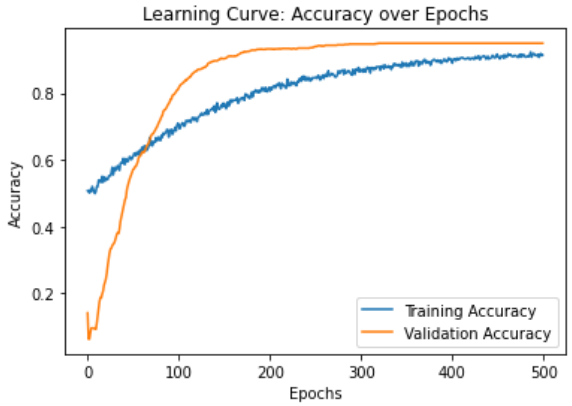


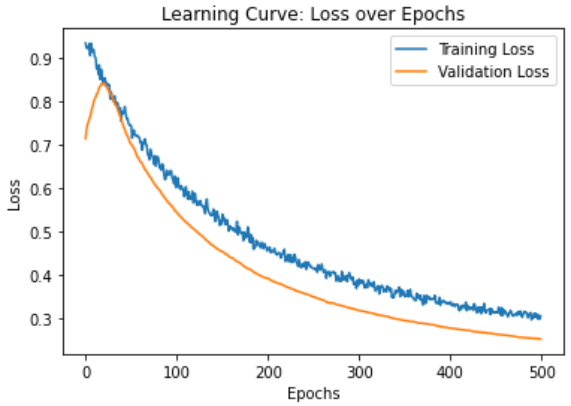
We have 0.26 True Positive, it is much greater than previous one.

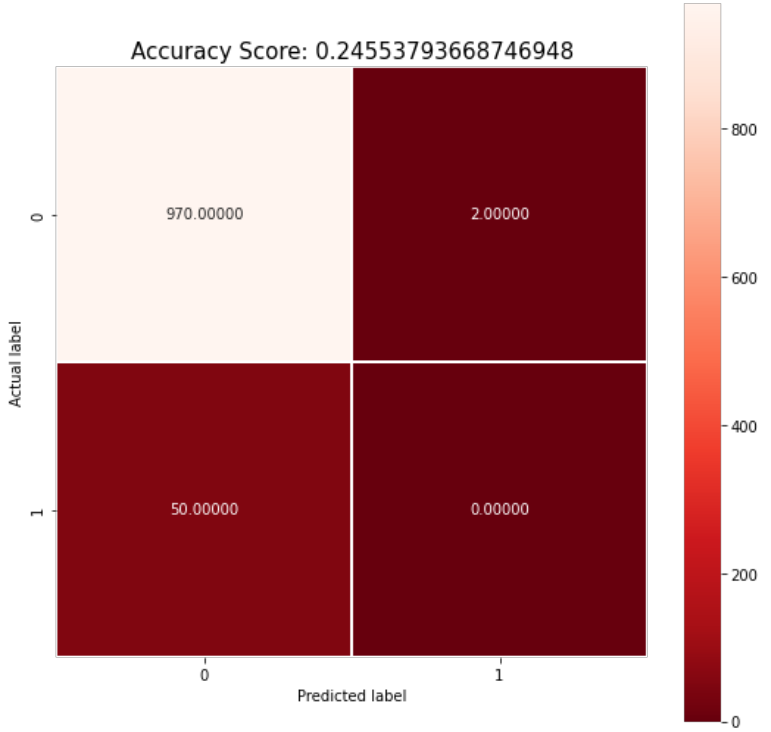


[5 Denses, Relu, sigmoid, SGD, lr = 0.0005, binary crossentropy, dropout = 0.4]

- 完全無法預測stroke

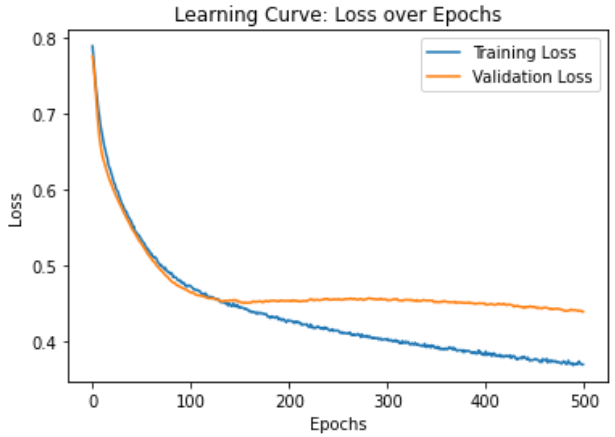


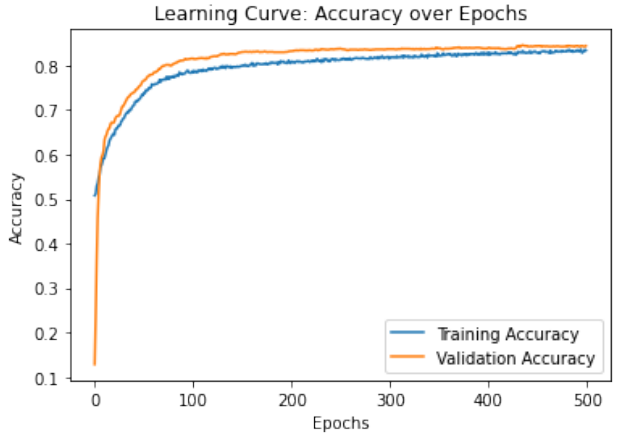


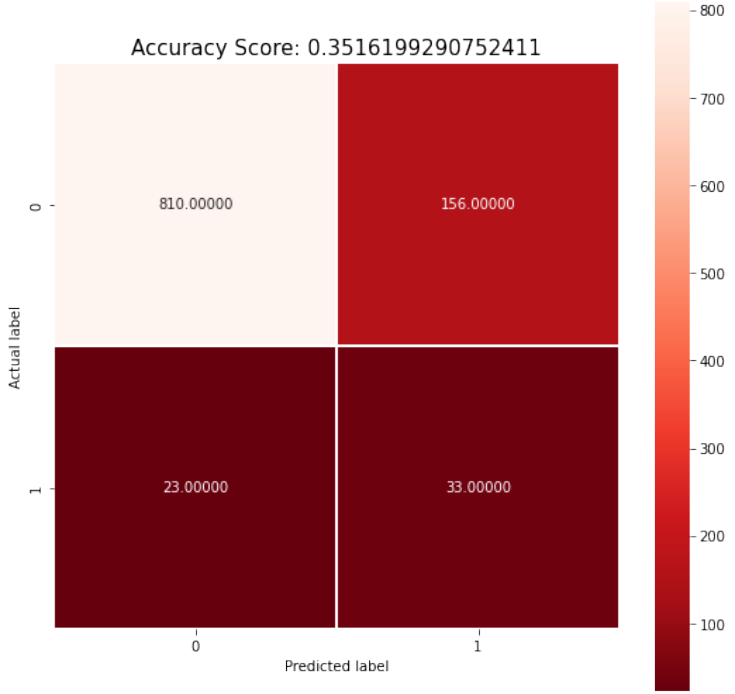


[6 Denses, tanh, sigmoid, SGD, lr = 0.0005, binary crossentropy, Batch normalization]

- Score: 0.35 / Accuracy: 0.82
- 雖然準確率相對不高，但從混淆矩陣來看，此模型是最佳模型。
- 因對於判斷病人是否生病來說，本該發現中風卻未發現較為嚴重，故認為此模型雖預測率降低，但仍有8成2的準確率，換來0.58的recall為好的結果。

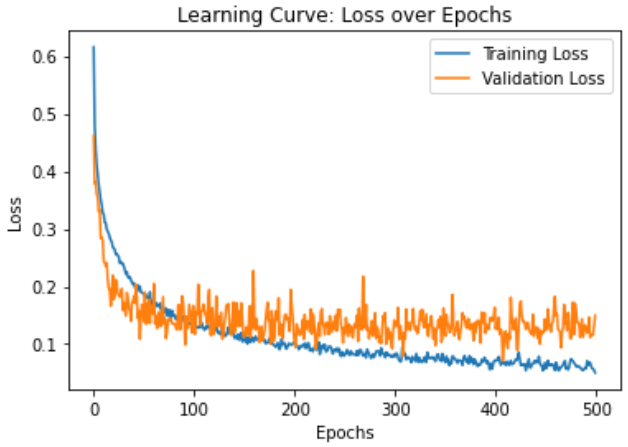


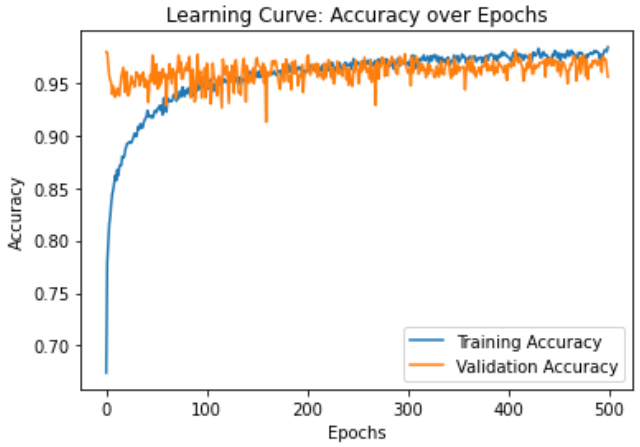


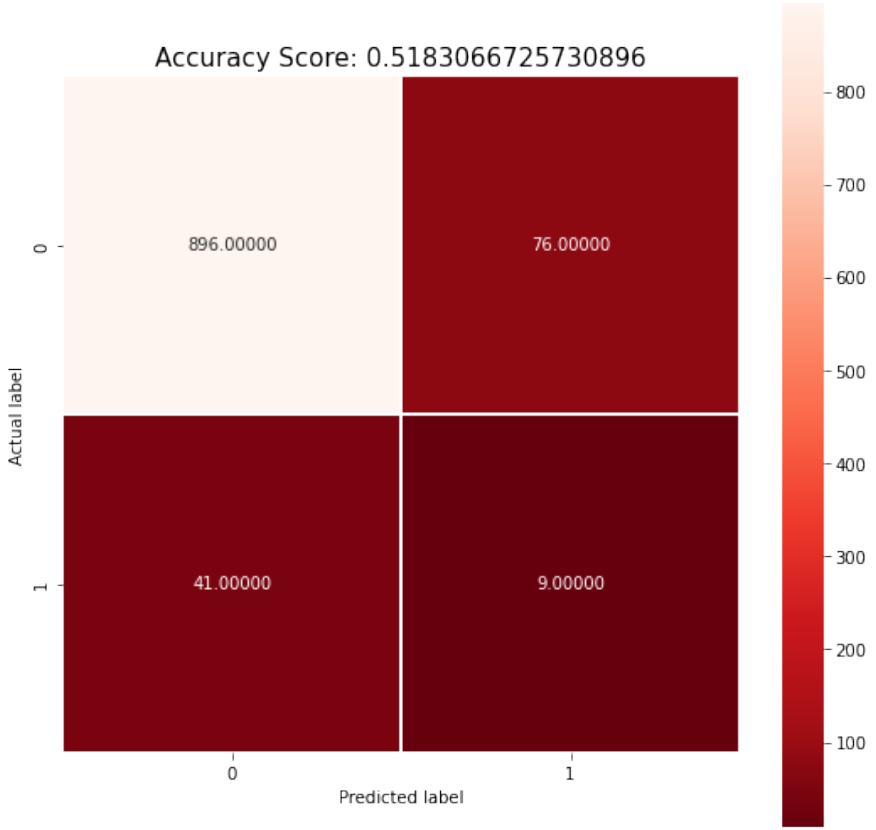


[6 Denses, tanh, sigmoid, Adam, binary crossentropy, Batch normalization]

- 學習相對SGD不穩定。
- Score較高：0.51
- Accuracy高一點點：0.885
- TP僅不到20%



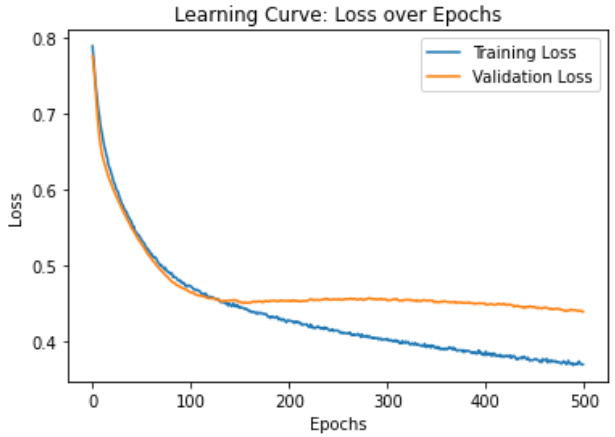


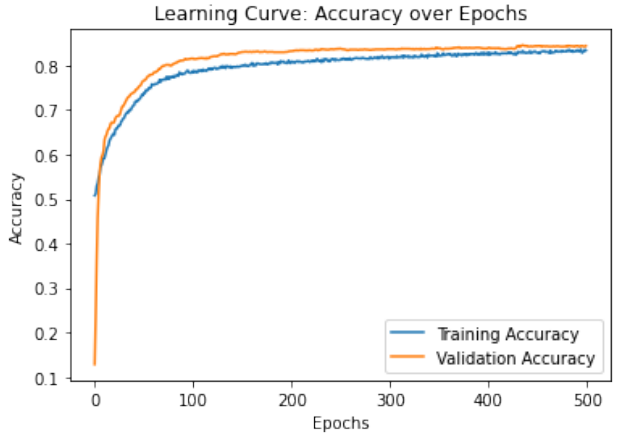


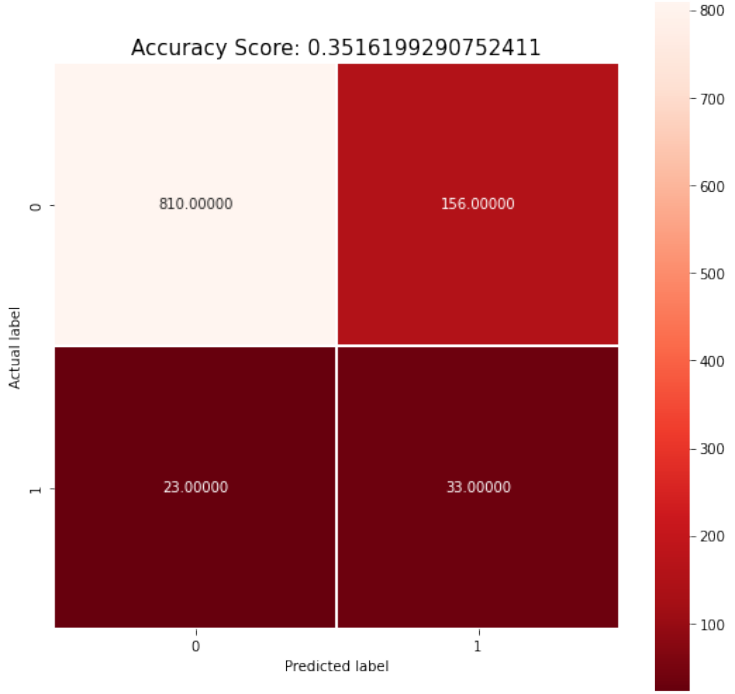
Experimental Results

[6 Denses, tanh, sigmoid, SGD, lr = 0.0005, binary crossentropy, Batch normalization]

- Score: 0.35 / Accuracy: 0.82
- 雖然準確率相對不高，但從混淆矩陣來看，此模型是最佳模型。
- 因對於判斷病人是否生病來說，本該發現中風卻未發現較為嚴重，故認為此模型雖預測率降低，但仍有8成2的準確率，換來0.58的recall為好的結果。







I think DNN perform a greater result

- Logistic Regression: acc 0.78, ROC 0.78 ,Precision: 0.16, Recall 0.80, F1 0.26
- KNN: acc 0.91, ROC 0.63 ,Precision: 0.13, Recall 0.43, F1 0.21
- SVM: acc 0.88, ROC 0.65 ,Precision: 0.14, Recall 0.48, F1 0.22
- Decision Tree: acc 0.92, ROC 0.54 ,Precision: 0.11, Recall 0.18, F1 0.13
- Random Forest: acc 0.97, ROC 0.55 ,Precision: 0.21, Recall 0.14, F1 0.17

- **DNN best: acc 0.82, ROC 0.71 ,Precision: 0.17, Recall 0.58, F1 0.26**

Thank you.

Have a good day.