

# Introduction to Machine Learning, Final Project

## GitHub link of your code

[Github Link](#)

## Reference if you used any code from other resources

### Reference Link

如何過透 tensorflow 實作 NN model

## Brief introduction

看完這次的作業，要預測的結果是0和1的機率分佈，第一直覺是就 logistic regression 去預測結果，再來就是利用 NN model 去預測結果，所以在這次的final project我就利用這兩個 ML model 來完成這次的作業。

## Methodology

### 1. Data pre-process

首先透過 pandas 將 train dataset 讀進來

然後透過 feature name 以及數據的觀察，可以發現 ID 數值全部不一樣可以將他 drop 掉。再來 product code 是產品的編號 ID，跟failure影響沒有關係，我也將 drop掉。再來我們可以看到 feature attribute\_0 跟 attribute\_1 這兩個 feature 都是由字串所組成的，我們要將它變成數值的資料。所以我們字串前面的 material\_ 去掉，只留下他後面的數字。這樣所有數據都是數值的資料了。再來有一些資料是 missing 的。我使用 sklearn 的 simpleImputer 來填空缺失的資料。並且參數使用 median。

### 2. Model architecture

模型設計的部分，我是使用 NN model，NN model 的部分 hidden layer 有兩層，一層 input layer 跟 一層 output layer。而input layer 的 dimension 為 23，因為總共有 23 個feature，而兩個 hidden layer 使用的unit 為 94，最後的output layer unit 則是設定成 1。Activation function 設定成 relu，。然後 drop 設定成 0.3 隨機捨棄 30%的連線。

### 3. Hyperparameter

```
model = Sequential()  
model.add(Dense(92, input_dim=23))  
model.add(Activation('relu'))  
model.add(Dropout(0.3))  
model.add(Dense(92))  
model.add(Activation('relu'))  
model.add(Dropout(0.3))  
model.add(Dense(1))  
model.add(Activation('sigmoid'))  
model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
```

```
loss='binary_crossentropy'  
optimizer='adam'
```

## Model Link

### Summary

#### Comparisons of different approaches

除了利用 NN model 外，我還使用了 logistic regression 來預測這次的作業。模型的參數就只有改變 max\_iter=1000，跑出來的結果大概是 0.58903 已經快要接近 baseline 了。但相較於 NN model NN model 擁有比較高一點點的 score 但是也大概高 0.002 左右。

### Result

#### 1. For logistic regression

---

 <b>result.csv</b> Complete (after deadline) · 15m ago	<b>0.58903</b>	<b>0.58107</b>	<input type="checkbox"/>
--	----------------	----------------	--------------------------

#### 2. For NN model

 <b>109550032.csv</b> Complete (after deadline) · 42m ago	<b>0.59134</b>	<b>0.58438</b>	<input type="checkbox"/>
---	----------------	----------------	--------------------------