Data Mining Project Report

- Data Preprocess

首先清除包含過多空值 (Null Value) 的特徵，因為過多的空值將會影響訓練資料集的品質，進而影響模型的訓練結果。如果該特徵的非空值 (Non-Null Value) 低於 80% 則刪除該特徵，包含：

特徵名稱	非空值比例
OP_time_minute	61.45%
OP_time_hour	61.45%
ASA	61.35%
CBC_WBC	31.46%
CBC_RBC	31.4%
CBC_HG	47.42%
CBC_HT	47.20%
CBC_MCV	31.36%
CBC_MCH	31.39%
CBC_MCHC	31.38%
CBC_RDW	31.37%
CBC_Platelet	31.74%
CBC_RDWCV	6.89%
BUN	46.21%
Crea	42.43%
GOT	43.86%
GPT	34.22%
ALB	29.27%
Na	30.47%
K	30.99%
UA	24.38%

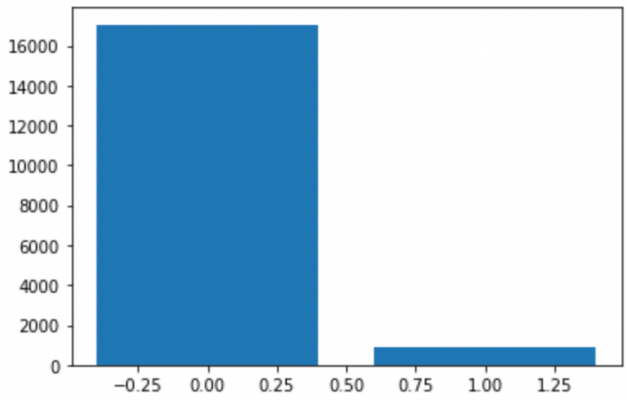
此外，也針對訓練資料集中的每一個樣本進行分析，如果該樣本有任一個特徵包含空值，則將該樣本去除。根據空值進行資料清理後，訓練資料集的維度由 (52159, 68) 變為 (52065, 47)。

接著針對「非數值特徵」編碼，在資料集中的非數值行特徵有 Joint 與 SEX。針對這兩個特徵進行 Label Encoding。確保所有特徵都是以數值表示後，針對每一個特徵進行 Rescaling。此專案中採取簡單的 Rescaling 方式，將每一個數值除以該特徵的最大值，確保所有數值皆在 0 到 1 的區間。

雖然訓練資料集已經根據空值進行資料清理，然而許多特徵所含有的資訊卻極低。為了排除包含資料量極低的特徵，將單一數值佔據超過 90% 的特徵刪除。訓練資料集的特徵數由原來的 47 變為 14，藉此達到降低維度的效果。

為了避免模型的訓練受到離群值 (Outlier) 的影響，分析每一個樣本的各個特徵。如果其中一個特徵被視為離群值，則將該樣本由訓練資料集中去除。離群值的判斷方式為，將該特徵進行標準化，得到該特徵的 Z Score，若大於 1.5 個標準差則視為離群值。訓練資料集的樣本數由 52065 減少為 17931。

在開始訓練模型之前，仍需要處理 Imbalanced Data 的問題。如下圖所示兩種類別的比例懸殊，類別 A 高達 95%，類別 B 僅有 5%。



在此專案中透過 SMOTE Oversampling 的方式，生成類別 B 的樣本。訓練資料集中的樣本數因此而增加為 32196 個。

- Model Topology (CNN)

以神經網路模型解決此問題時，將此問題視為一種 Anomaly Detection。因此，以 Autoencoder 為模型架構，並由 Convolution Layer 組成。

如下圖所示，Autoencoder 由 Encoder 與 Decoder 組成，Encoder 與 Decoder 主要包含 Conv1D、MaxPooling1D 等 Layer。在 Encoder 中，3 個 Conv1D Layer 的 Kernel Size 皆為 3，Filter 的數目分別為 64、32 與 16。在 Convolution Layer 後都會接上一個 MaxPooling1D Layer，並設定 pool_size 為 2。

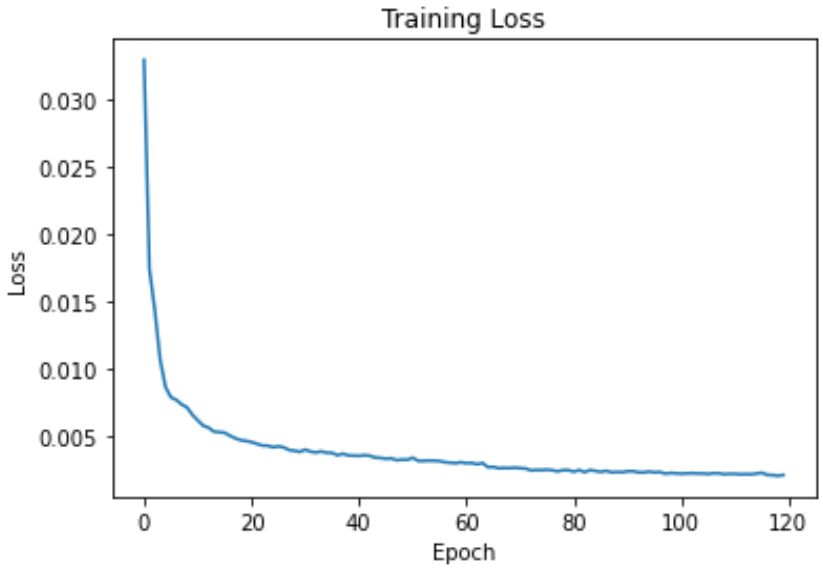
Decoder 中的 Layer 設定與 Encoder 雷同但是順序相反，且將 MaxPooling 換為 UpSampling，目的是為了讓 Autoencoder 輸入與輸出的形狀相同。

為了加速 Autoencoder 的收斂速度，在 Encoder 與 Decoder 中都有使用 BatchNormalization Layer；為了避免模型 Overfitting，加入 Dropout Layer 並設定 rate 為 0.2。

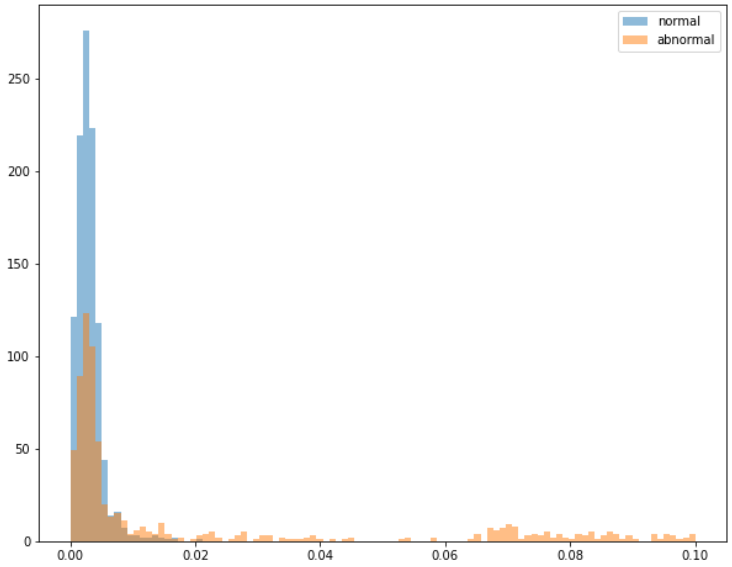
```
self.encoder = tf.keras.Sequential([
    layers.Input(shape=(16, 1)),
    layers.Conv1D(64, 3, strides=1, padding='same', activation='relu'),
    layers.BatchNormalization(),
    layers.MaxPooling1D(pool_size=2),
    layers.Conv1D(32, 3, strides=1, padding='same', activation='relu'),
    layers.Dropout(rate=0.2),
    layers.BatchNormalization(),
    layers.MaxPooling1D(pool_size=2),
    layers.Conv1D(16, 3, strides=1, padding='same', activation='relu'),
    layers.BatchNormalization(),
    layers.MaxPooling1D(pool_size=2),])

self.decoder = tf.keras.Sequential([
    layers.Conv1D(16, 3, strides=1, padding='same', activation='relu'),
    layers.BatchNormalization(),
    layers.UpSampling1D(size=2),
    layers.Conv1D(32, 3, strides=1, padding='same', activation='relu'),
    layers.Dropout(rate=0.2),
    layers.BatchNormalization(),
    layers.UpSampling1D(size=2),
    layers.Conv1D(64, 3, strides=1, padding='same', activation='relu'),
    layers.BatchNormalization(),
    layers.UpSampling1D(size=2),
    layers.Conv1D(1, 3, strides=1, padding='same', activation='sigmoid')])
```

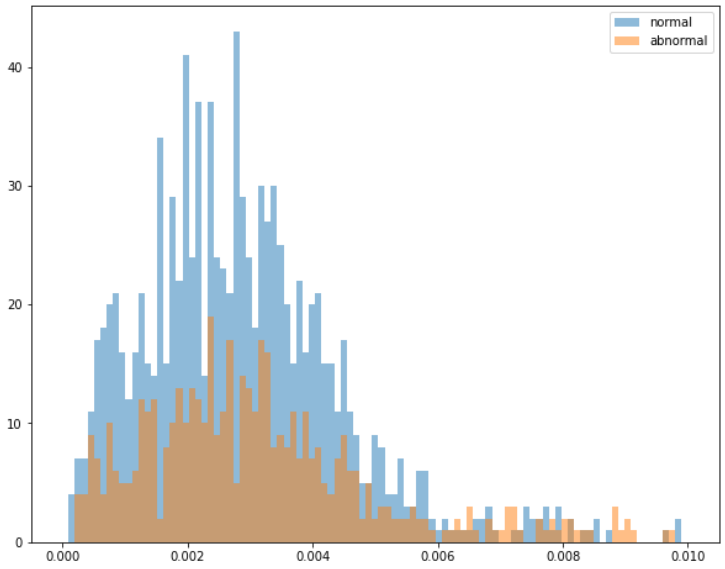
下圖為 Autoencoder 在訓練時 Loss 的變化，可以發現模型的收斂效果還算不錯。



下圖為 Autoencoder 在 Validation 時的表現。X 軸為 Mean Absolute Error (MAE)，Y 軸為數量。由下圖可以發現，訓練過後的 Autoencoder 在處理部分的 Abnormal Data (outcome = 1) 時，仍會得到很小的 MAE，導致無法分辨此樣本為 Abnormal Data。



下圖中，只聚焦在 Normal (outcome = 0) 與 Abnormal Data (outcome = 1) 重疊的區域，可以發現 Autoencoder 針對兩種 Data 的臨界點大約在 MAE = 0.006 的位置。此位置也作為模型在預測階段時，設定門檻的參考。



最後，下圖為 Autoencoder 模型的預測結果。針對第一個測試資料，Autoencoder 模型將其預測為 Normal Data (outcome = 0)。

```
array([0, 1, 0, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0,
       1, 0, 1, 1, 0, 1, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 1, 1, 0, 0, 1, 0,
       1, 1, 0, 0, 1, 0, 1, 1, 1, 1, 0, 0, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1,
       0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1,
       0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0,
       0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1, 0, 1,
       0, 0, 0, 0, 1, 1, 1, 0, 0, 1, 0, 0, 0, 1, 1, 1, 0, 1, 1, 0, 1, 1,
       0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 1, 0, 1, 0, 0, 1, 1, 0, 1, 0,
       0, 0, 0, 0, 1, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0,
       1, 1, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 1,
       1, 0, 0, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 0, 1, 0, 0, 1, 1, 0,
       0, 0, 0, 0, 1, 0, 0, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0,
       0, 1, 1, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0,
       0, 1, 0, 1, 1, 1, 0, 0, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 1, 0,
       0, 1, 0, 0, 1, 0, 1, 0, 1, 1, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0,
       1, 1, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0,
       0, 0, 1, 0, 1, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0,
       0, 0, 1, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0,
       1, 0, 1, 0, 1, 1, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1,
       1, 1, 1, 1, 1, 0, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0,
       0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1,
       1, 1, 0, 1, 0, 1])
```

Autoencoder 的預測方式為計算模型的輸入與輸出之間的 MAE，如果 MAE 小表示模型能夠有效的重新生成原來的輸入，表示此輸入與原來的訓練資料雷同，因此被判斷為 Normal Data；相反的，若 MAE 大則表示模型無法生成原來的輸入資料，表示該資料為 Anomaly，視為 Abnormal Data。

輸入資料的所有特徵會在神經網路中進行複雜的特徵工程，因此難以透過簡單的 IF-ELSE 規則，分析模型對於每一種特徵的判斷。

- Model Topology (Logistic Regression)

透過 Scikit-Learn 套件建立 Logistic Regression 模型，並設定 max_iter = 400，避免在模型收斂之前達到 Iteration 的上限。將 Logistic Regression 進行訓練，並使用 Validation Dataset 分析模型的準確度：

Training Accuracy	84%
Validation Accuracy	97%
Validation Accuracy on Anomaly	75%

如上表所示，模型在 Training Dataset 的 Accuracy 為 84%，在 Validation Dataset 的 Accuracy 為 97%。然而，若只看 Validation Dataset 整體的 Accuracy 仍然無法確定模型是否正確地將 Abnormal Data 分類出來。因此，Validation Accuracy on Anomaly 則是針對 Validation Dataset 中的 Abnormal Data 進行分類的結果。

在此專案中，Validation Dataset 的組成為 950 個 Normal Data 與 50 個 Abnormal Data。因此，75% 表示模型能夠將 37 個 Abnormal Data 分類正確。

接著，輸入測試資料至模型中進行預測。下圖為模型的預測結果：

```
array([1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1,
       0, 0, 1, 1, 0, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1,
       1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1, 0, 1, 1, 1,
       1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1,
       0, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 0, 0, 1,
       1, 0, 0, 0, 1, 0, 1, 0, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1,
       0, 0, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1,
       1, 0, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1,
       1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 1, 1, 0, 1, 0, 0, 1, 1, 1, 1, 1,
       0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 0, 1,
       1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 1, 1, 1, 1, 1,
       1, 0, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1,
       1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 0, 0, 1, 1, 1, 0, 1, 0,
       0, 1, 1, 0, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 0, 1, 1, 1,
       0, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 0,
       0, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 0, 0, 1, 1, 0,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1,
       0, 0, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 0, 0, 1, 1, 0, 0, 0, 1, 1, 0,
       1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 1, 1,
       1, 1, 1, 0, 1, 1, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1,
       0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 1,
       0, 0, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1,
       1, 1, 0, 1, 0, 1])
```

可以發現模型將第一筆測試資料預測為 Abnormal Data (outcome = 1)。可以藉由觀察 Logistic Regression 中的 Coefficient，推測模型對於每一個參數的詮釋。

下圖為 Logistic Regression 的 Coefficient 與 Intercept：

```
array([[-2.56441488,  0.6130094 ,  9.50517098,  0.48400466, -6.415574  ,
        -0.31315842,  0.29837615,  2.8989094 ,  2.56088325,  6.18207648,
         5.59454268,  6.6369577 ,  5.11228026]])
```

```
array([5.56593086])
```

觀察 Coefficient 與 Intercept 可以發現模型較容易輸出 1 的結果，也就是說模型容易將樣本預測為 Abnormal，使得預測結果中以 Abnormal 居多。

- Model Topology (Support Vector Machine)

透過 Scikit-Learn 套件建立 Support Vector Machine，並使用 Polynomial 作為 Kernel Function，設定 Regularization Parameter 為 1.0 (預設)，設定 gamma 為 0.3。

將 Support Vector Machine 進行訓練，並使用 Validation Dataset 分析模型的準確度：

Training Accuracy	84%

Validation Accuracy	98%
Validation Accuracy on Anomaly	70%

接著進行模型的預測：

```
array([1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1,
       0, 0, 1, 1, 0, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1,
       0, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1, 0, 1, 1, 1,
       1, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 0, 1, 1, 1, 1, 0, 0, 1,
       0, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 0, 0, 1,
       1, 0, 0, 0, 1, 0, 1, 0, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1,
       0, 0, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1,
       1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1,
       1, 0, 0, 0, 1, 1, 0, 0, 1, 1, 0, 1, 1, 0, 1, 0, 0, 1, 1, 1, 1, 1,
       0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 0, 1,
       1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 1, 1, 1, 1,
       1, 0, 0, 1, 1, 0, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 0, 0, 1, 1, 1,
       1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 0, 0, 1, 1, 1, 0, 1, 0,
       0, 1, 1, 0, 1, 1, 1, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 1, 1,
       0, 1, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 0,
       0, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 0, 0, 1, 1, 0,
       0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1,
       0, 0, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 0, 0, 1, 1, 0, 0, 0, 1, 1, 0,
       1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 1, 1,
       1, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1,
       0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 1,
       0, 0, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1,
       1, 1, 0, 1, 0, 1])
```

可以發現模型將第一筆測試資料預測為 Abnormal Data (outcome = 1)。在此專案中，使用 Polynomial 作為 SVM 的 Kernel，因為是 Non-Linear，所以難以透過與 Logistic Regression 類似的方式來詮釋模型的輸出。然而，可以觀察到 SVM 也將大部分測試資料預測為 Abnormal，與 Logistic Regression 有類似的結果。

- Model Topology (Decision Tree, C4.5)

透過 Scikit-Learn 套件建立 Decision Tree，並設定 criterion 為 entropy，設定 max_depth 為 14。

將 Decision Tree 進行訓練，並使用 Validation Dataset 分析模型的準確度：

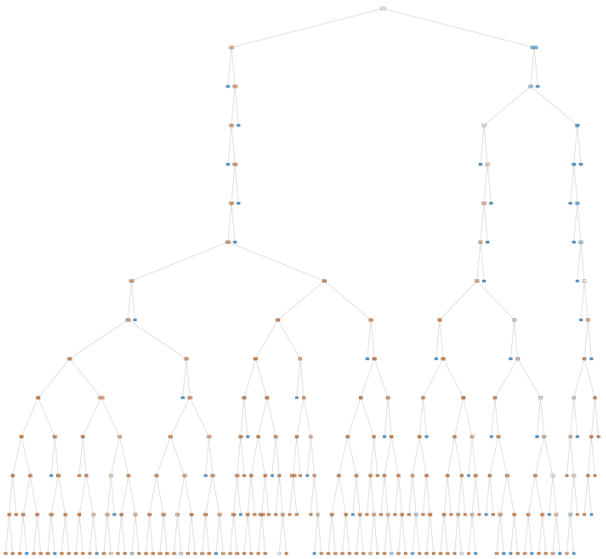
Training Accuracy	94%
Validation Accuracy	98%
Validation Accuracy on Anomaly	80%

相較於 Logistic Regression 與 Support Vector Machine，Tree-Based 的演算法有比較好的表現。

接著進行模型的預測：

```
array([1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1,
       0, 0, 1, 1, 0, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1,
       0, 1, 0, 0, 1, 0, 1, 1, 1, 1, 0, 0, 1, 0, 0, 1, 0, 1, 0, 1, 1, 1,
       1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1,
       0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 0, 0, 1,
       1, 0, 0, 0, 1, 0, 1, 0, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1,
       0, 0, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1,
       1, 0, 1, 0, 0, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1,
       1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1,
       0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 0, 1,
       1, 1, 1, 0, 1, 0, 0, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 1, 1, 1, 1,
       1, 0, 0, 1, 1, 0, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 0, 0, 1, 1, 1,
       1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 0, 0, 1, 1, 1, 1, 1, 0,
       0, 1, 1, 0, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 0, 1, 1, 1,
       0, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 0,
       0, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 0, 0, 1, 1, 0,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1,
       0, 0, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 0, 0, 1, 1, 0, 0, 0, 1, 1, 0,
       1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 1, 1,
       1, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1,
       0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 1,
       0, 0, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1,
       1, 1, 0, 1, 0, 1])
```

可以發現模型將第一筆測試資料預測為 Abnormal Data (outcome = 1)。可以透過 Decision Tree 的視覺化工具了解模型如何預測樣本。下圖為此模型的視覺化。



藉此可以將模型的預測結果以 IF-ELSE 規則的方式表示。

- Model Topology (Random Forest)

透過 Scikit-Learn 套件建立 Random Forest，並設定 n_estimators 為 10，設定 max_depth 為 25。

將 Random Forest 進行訓練，並使用 Validation Dataset 分析模型的準確度：

Training Accuracy	98%
Validation Accuracy	98%
Validation Accuracy on Anomaly	75%

相較於原來的 Decision Tree，Training Accuracy 稍微上升，Validation Accuracy 不變，但是 Validation Accuracy on Anomaly 下降。可能是因為 Random Forest 是由更多 Decision Tree 組成的模型，複雜的模型也容易發生 Overfitting。

接著進行模型的預測：

```
array([1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1,
       0, 0, 1, 1, 0, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1,
       0, 1, 0, 0, 1, 0, 1, 1, 1, 1, 0, 0, 1, 0, 0, 1, 0, 1, 0, 1, 1, 1,
       1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1,
       0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 0, 0, 1,
       1, 0, 0, 0, 1, 0, 1, 0, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1,
       0, 0, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1,
       1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1,
       1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1,
       0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 0, 1,
       1, 1, 1, 0, 1, 0, 0, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 1, 1, 1, 1,
       1, 0, 0, 1, 1, 0, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 0, 0, 1, 1, 1,
       1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 0, 0, 1, 1, 1, 1, 1, 0,
       0, 1, 1, 0, 1, 1, 1, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 1, 1,
       0, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 0,
       0, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 0, 0, 1, 1, 0,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1,
       0, 0, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 0, 0, 1, 1, 0, 0, 0, 1, 1, 0,
       1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 1, 1,
       1, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1,
       0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 1,
       0, 0, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1,
       1, 1, 0, 1, 0, 1])
```

可以發現模型將第一筆測試資料預測為 Abnormal Data (outcome = 1)。Random Forest 是一種 Ensamble Learning 的演算法，也就是說它是許多小模型 (Decision Tree) 的組成，最終的預測結果是所有小模型預測結果的整合。

因此我們可以視覺化 Random Forest 中的每一個 Decision Tree，將每一個 Decision Tree 化為 IF-ELSE 的規則，最終 Random Forest 的預測結果就能透過 IF-ELSE 的規則來描述。

- Conclusion

在初期的實作過程中，將重點放在 Imbalanced 問題上，花了許多時間研究 Resampling 的方法，然而卻無法提升模型在 Abnormal Data 上的準確度。在排除 Imbalanced 問題後，嘗試不同的模型進行預測，仍然無法在 Abnormal Data 上得到好的表現。

最終，將重點放在資料集的品質上。透過空值與資訊量的想法去除過多的特徵，大幅降低每一個樣本的維度。為了避免 Minority Class Resampling 的數量太多，因此嘗試刪減 Majority Class 的樣本數量。以離群值的角度切入，大幅減少 Majority Class 的樣本數量。

將處理過後的資料集進行模型的訓練後，發現模型在 Validation 時的表現顯著提升，較不會將所有的樣本都預測為 Normal Data (Majority Class)。