一、First 3 features and thresholds in MIMIC dataset

1. (mvar12, 0.204)
2. (CMO, 0.24)
3. (CA,0.788)

二、Build Tree and improve

在 MIMIC 的部分，我用的是 DecisionTreeClassifier 去建樹，並且透過 f1.score 去判斷 validation data 的預測結果，微調 Classifier 中的各個數值，最後以

train_test_split(random_state=5, train_size=0.65),

DecisionTreeClassifier(criterion='gini', max_depth = 5, random_state=15, splitter='random')

找到較佳的 f1.score（0.514772158237356），因此以這個條件進行預測。

三、Preprocess data

在讀入的資料中，我刪去 subject_id, indextime, insurance, ethnicity。這些包含了病患編號、登記時間與保險這三項與患者本身身體狀況沒有關係的 feature，另外我也認為種族的差異並不會影響眾多疾病對病患的影響。

在 Preprocess 的部分還有進行 split data 將資料分成 train 與 validation 兩組，透過 train_test_split() 將資料隨機分為 train/validation（比例：0.65/0.35）。

四、Summarize

1. entropy()：計算這個 data 內，wait 為 T, F 的數量，並透過 entropy = plog2(p) + nlog2(n)（p=T/T+F, n=F/T+F）得到 entropy。
2. bestsplit()：先將第一個 column 值由小到大排列，如果第 j 與 j-1 個值不相同，取他們的中間值作為 threshold，然後計算左右 entropy 總和。一個 column 做完就換下一個 column sort 然後做一樣的事。每當算出來的 entropy < 當前 min entropy，就記下當時的 column 及 thresholds。
3. buildtree()：透過 bestsplit 可以找到最佳的 column 與 thresholds，然後將資料分成兩堆，再將兩邊的資料放入 buildtree 中，即以 recursive 的作法做 buildtree，下方附上部分程式碼。

```
col, val = findBestSplit(df)
left, right = partition(df, col, val)
l_tree, l_feature, l_thresholds = buildTree(left, depth-1)
r_tree, r_feature, r_thresholds = buildTree(right ,depth-1)
```

4. MIMIC dataset part

   如同前面敘述過的，用 DecisionTreeClassifier 建樹並預測。

5. Visualize Tree

   透過 StringIO, Image, pydotplus 函式庫將 Classifier 的 feature、threshold、smaple 數即分類狀況顯示於圖上。