1. Implement the Decision Tree

Step 1: calculate the entropy

計算傳入的 data['Wait']有幾個是 T，並算出是 T 的機率(p)，帶入 entropy 的公式，-p*log_2^p-(1-p)* log_2^{1-p}，就可以算出 entropy。

Step 2: search for the best split

用(i+0.5)去切 data，並計算出 combination entropy，取最小的 entropy，代表是最好的切法。

Step 3: build the decision tree

用 dfs 的方式去遞迴，直到目標深度或者是全部的 case 都被分類完，並記錄每一層的 decision 跟 threshlod。

以下是我的 tree:

['Patrons 1.5', ['Hungry 0.5', 'F', ['Friday 0.5', 'F', ['Price 0.5', 'F', 'T']]], 'T']

可以看到在第四層就已經完全分類完了。

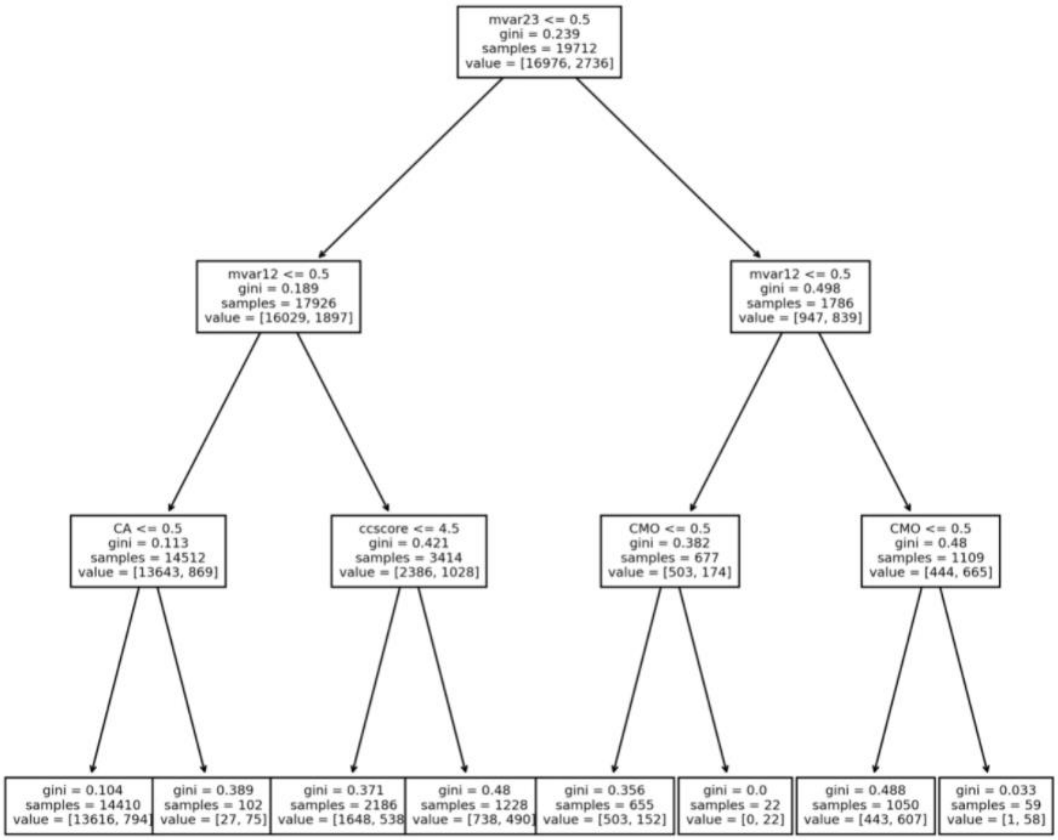
2. Classification with the MIMIC Dataset

我首先把['indextime']跟['subject_id']從 data 中剃除，因為這兩個資料跟結果沒關係。

我用的方法是 decision tree，我是從 sklearn.tree import DecisionTreeClassifier 這個 model 做使用，輸入 x_train 跟 y_train 做訓練。此外，decision tree 的深度並不是越深越好，太深的話可能會出現 overfit 的狀況，所以我將深度設定為 15。

至於評估訓練成果的 score function，我是從 sklearn.metrics import f1_score，拿 x_validation 輸入訓練好的 model，將出來的 prediction 跟 y_validation 輸入 f1_score，我得到的值約為 0.48。

以下是我前三層 features 跟 threshold:



3. Bonus

我從 sklearn import tree，並使用以下 function 印出 tree 的前五層:

```
tree.plot_tree(model,feature_names=x_train.columns, fontsize=7,max_depth=4);
```