```
### START CODE HERE ###
# Define the attributes
max_depth = 2
depth = 0
min_samples_split = 2

# total number of trees in a random forest
n_trees = 10

# number of features to train a decision tree
n_features = 7

# the ratio to select the number of instances
sample_size = 0.8
n_samples = int(training_data.shape[0] * sample_size)
### END CODE HERE ###
```

如附圖，

1. 共使用了 10 個 trees。
2. 使用了 7 個 features。(`selected_features = ['wbc_apache', 'bmi', 'diabetes_mellitus', 'weight', 'map_apache', 'heart_rate_apache', 'hospital_death'] {'map_apache <= 50.5': [1, 0]}`)
3. 使用所有 data 裡面 80%的 data 作為 instance。
4. 在前面處理 data 的時候，發現常常會多或少 feature 或重複 feature，在中間的時候，也很常 data 的 frame 不符合要求，系統報錯，還有在最後的時候，發現自己測試的 f-score 都是 0.74、0.75，但丟到 kaggle 上卻變成了 0.3、0.4。
5. 最後有稍微借助 chatgpt 的力量，他幫我偵測到我沒有處理好 hospital_death 的部分，導致可能這個 feature 重複出現或是根本沒出現，致使系統報錯，然後在 f-score 的部分，回去翻老師的 ppt，覺得是有點 overfit，所以把原本設定 max_depth 從 6 改成 4，把 sample_size 從 0.6 改成 0.8，tesing_data 的跟 validation_data 的 f-score 有很明顯變準，最後從 4 改成 2，得到了現在的成績，validtion_data 跟 testing_data 也幾乎變成一樣，學會了理論對實作果然很重要。