使用 NER 模型產生的風險詞，並加上風險詞的上下文，以句點作為結尾。將這個資料作為訓練集，訓練隨機森林分類模型

分別對每個案件類型建立分類模型

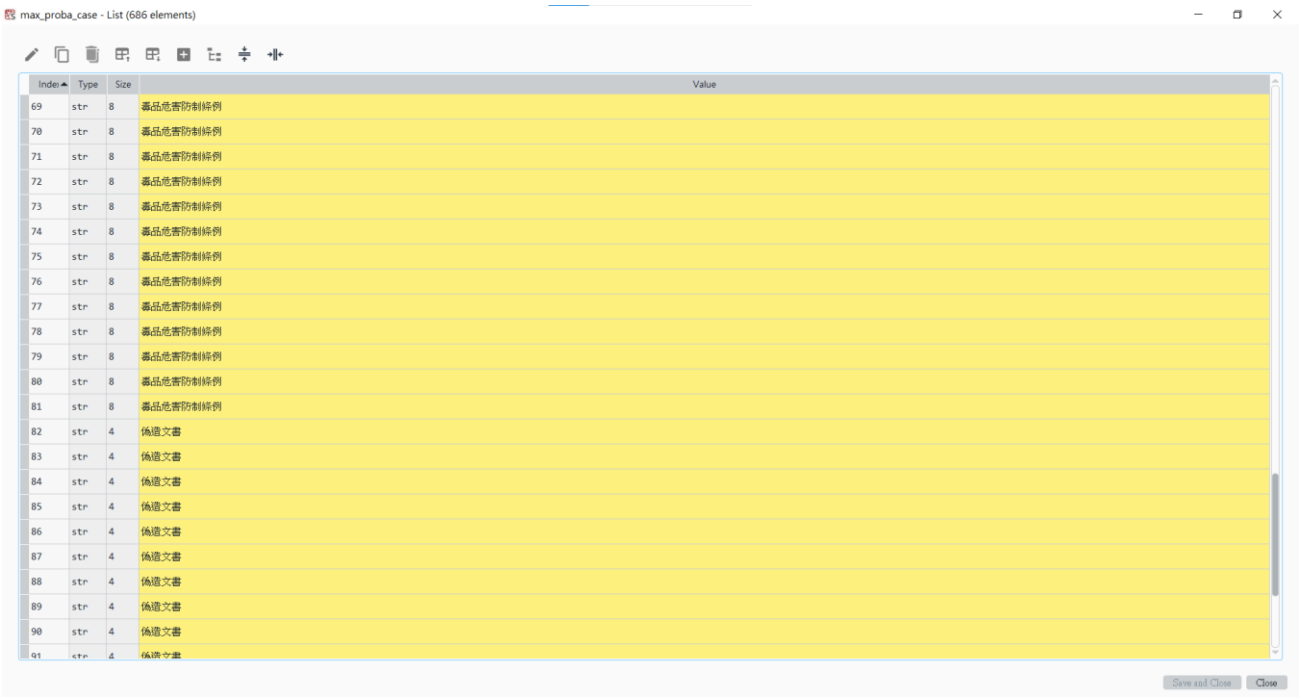
```
#Use Randomforest to train classifier model
rf_classifier_seg_drug = RandomForestClassifier()
rf_classifier_seg_forged_documents = RandomForestClassifier()
rf_classifier_seg_gamble = RandomForestClassifier()
rf_classifier_seg_negligent_injury = RandomForestClassifier()
rf_classifier_seg_public = RandomForestClassifier()
rf_classifier_seg_theft = RandomForestClassifier()

rf_classifier_seg_drug.fit(train_embedding, merged_df['label_case1']==1)
rf_classifier_seg_forged_documents.fit(train_embedding, merged_df['label_case2']==1)
rf_classifier_seg_gamble.fit(train_embedding, merged_df['label_case3']==1)
rf_classifier_seg_negligent_injury.fit(train_embedding, merged_df['label_case4']==1)
rf_classifier_seg_public.fit(train_embedding, merged_df['label_case5']==1)
rf_classifier_seg_theft.fit(train_embedding, merged_df['label_case6']==1)
```

將上述的資料轉換成詞嵌入向量，作為輸入屬性，而目標屬性分別為「是案件（1）/非案件（0）」，各別對應六種案件，例如：是竊盜（1）/非竊盜（0）。將測試資料輸入六個訓練好的分類模型，就會得到六組是非的機率值（predict_proba），再取出最高「是」的機率轉換成對應的案件，以此作為分類預測結果

```
probs = case[j].predict_proba(test)
probs_all_list.append(probs)
```

預測結果如下：



另外使用同一種方式，但不經過 NER 模型，直接將原文轉換成詞嵌入向量，訓

練隨機森林模型，將這兩個模型進行比較

上下文：

詞嵌入(分類)	Period
F1_micro	0.9723
F1_macro	0.9685

原文：

詞嵌入(分類)	Original
F1_micro	0.8921
F1_macro	0.8701