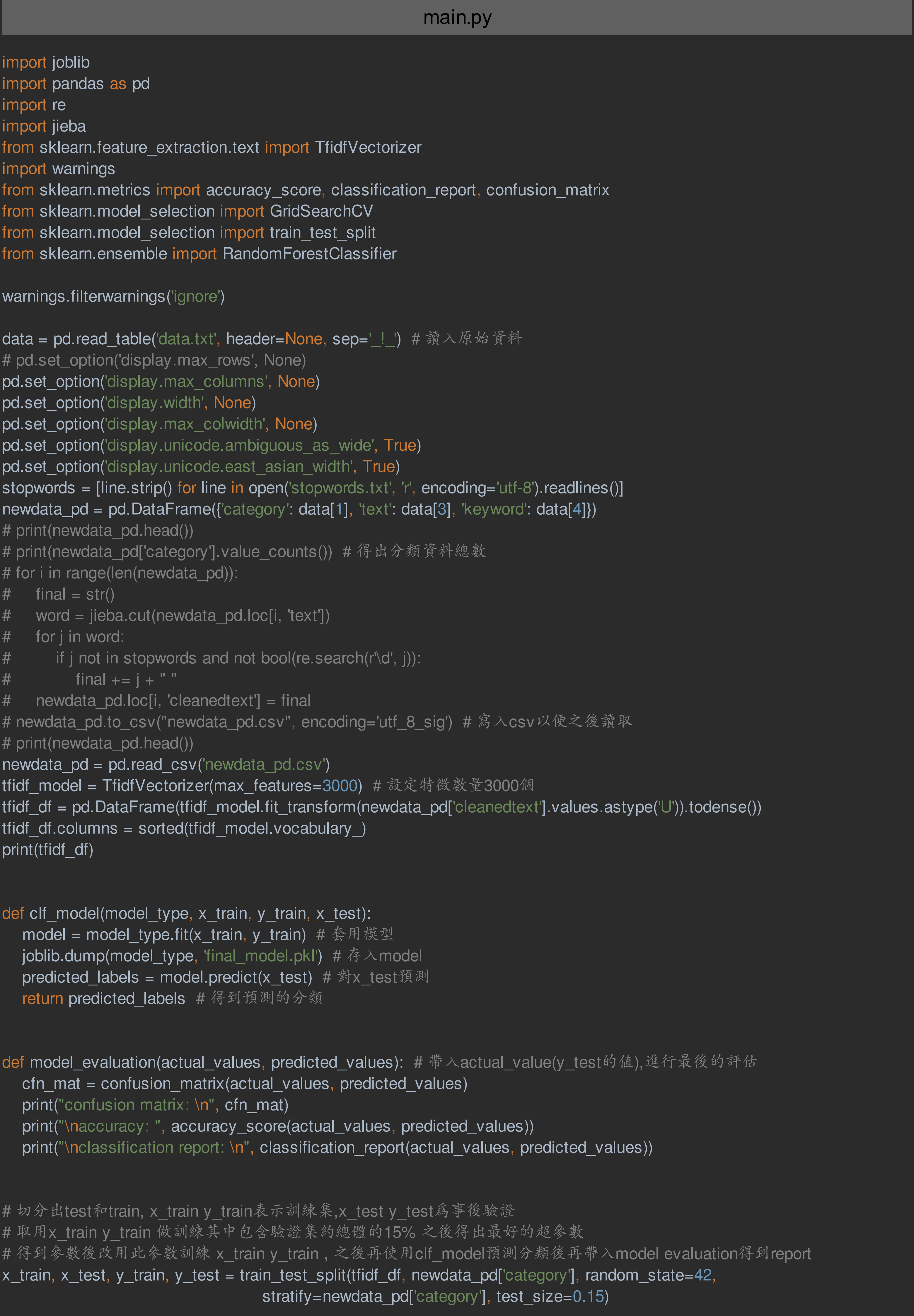
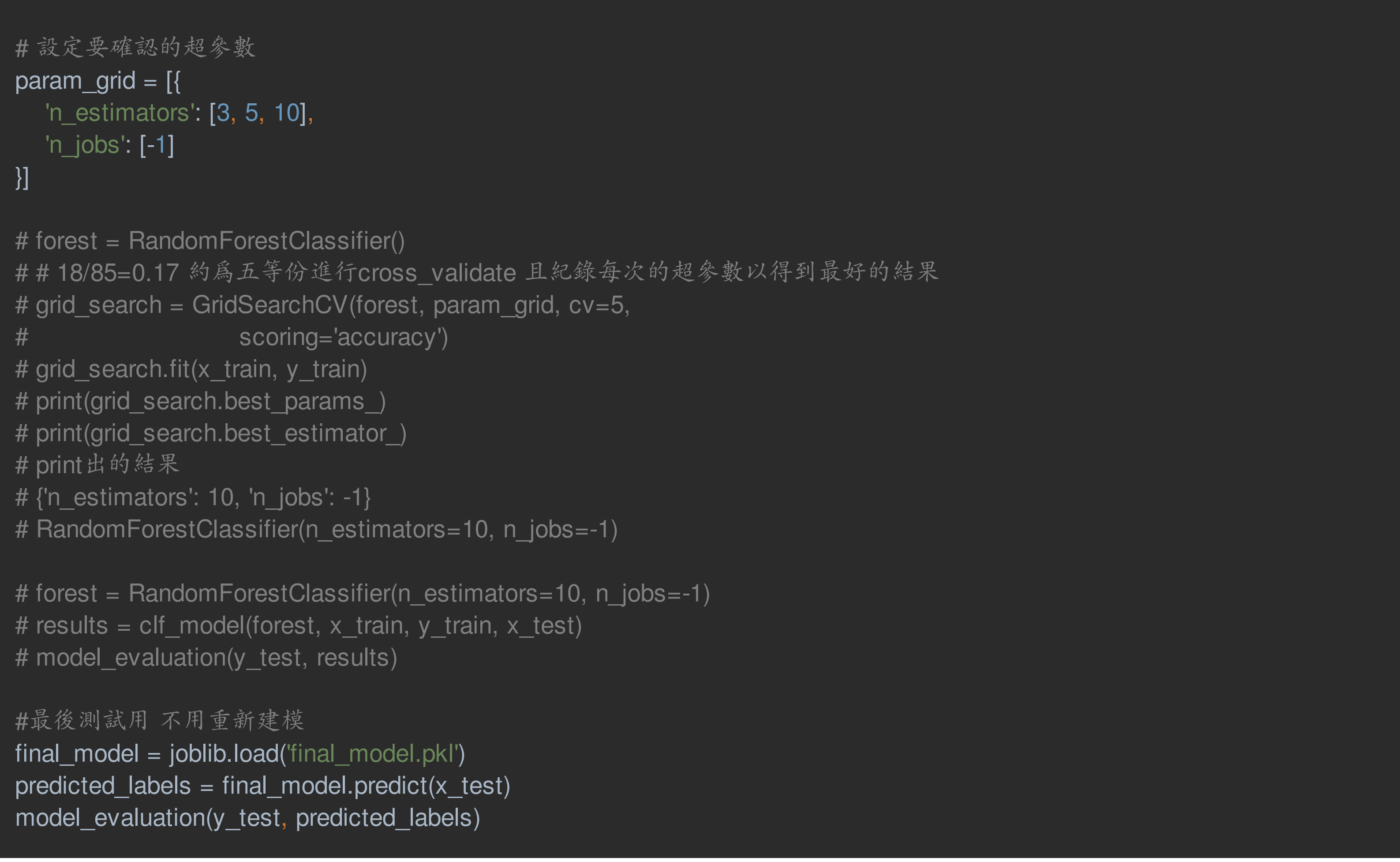


main.py

```python
import joblib
import pandas as pd
import re
import jieba
from sklearn.feature_extraction.text import TfidfVectorizer
import warnings
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
from sklearn.model_selection import GridSearchCV
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier

warnings.filterwarnings('ignore')

data = pd.read_table('data.txt', header=None, sep='_!_')  # 讀入原始資料
# pd.set_option('display.max_rows', None)
pd.set_option('display.max_columns', None)
pd.set_option('display.width', None)
pd.set_option('display.max_colwidth', None)
pd.set_option('display.unicode.ambiguous_as_wide', True)
pd.set_option('display.unicode.east_asian_width', True)
stopwords = [line.strip() for line in open('stopwords.txt', 'r', encoding='utf-8').readlines()]
newdata_pd = pd.DataFrame({'category': data[1], 'text': data[3], 'keyword': data[4]})
# print(newdata_pd.head())
# print(newdata_pd['category'].value_counts())  # 得出分類資料總數
# for i in range(len(newdata_pd)):
#     final = str()
#     word = jieba.cut(newdata_pd.loc[i, 'text'])
#     for j in word:
#         if j not in stopwords and not bool(re.search(r'\d', j)):
#             final += j + " "
#     newdata_pd.loc[i, 'cleanedtext'] = final
# newdata_pd.to_csv("newdata_pd.csv", encoding='utf_8_sig')  # 寫入csv以便之後讀取
# print(newdata_pd.head())
newdata_pd = pd.read_csv('newdata_pd.csv')
tfidf_model = TfidfVectorizer(max_features=3000)  # 設定特徵數量3000個
tfidf_df = pd.DataFrame(tfidf_model.fit_transform(newdata_pd['cleanedtext'].values.astype('U')).todense())
tfidf_df.columns = sorted(tfidf_model.vocabulary_)
print(tfidf_df)


def clf_model(model_type, x_train, y_train, x_test):
    model = model_type.fit(x_train, y_train)  # 套用模型
    joblib.dump(model_type, 'final_model.pkl')  # 存入model
    predicted_labels = model.predict(x_test)  # 對x_test預測
    return predicted_labels  # 得到預測的分類


def model_evaluation(actual_values, predicted_values):  # 帶入actual_value(y_test的值),進行最後的評估
    cfn_mat = confusion_matrix(actual_values, predicted_values)
    print("confusion matrix: \n", cfn_mat)
    print("\naccuracy: ", accuracy_score(actual_values, predicted_values))
    print("\nclassification report: \n", classification_report(actual_values, predicted_values))


# 切分出test和train, x_train y_train表示訓練集,x_test y_test爲事後驗證
# 取用x_train y_train 做訓練其中包含驗證集約總體的15% 之後得出最好的超參數
# 得到參數後改用此參數訓練 x_train y_train , 之後再使用clf_model預測分類後再帶入model evaluation得到report
x_train, x_test, y_train, y_test = train_test_split(tfidf_df, newdata_pd['category'], random_state=42,
                                                    stratify=newdata_pd['category'], test_size=0.15)
```
]



```
# 設定要確認的超參數
param_grid = [{
    'n_estimators': [3, 5, 10],
    'n_jobs': [-1]
}]

# forest = RandomForestClassifier()
# # 18/85=0.17 約爲五等份進行cross_validate 且紀錄每次的超參數以得到最好的結果
# grid_search = GridSearchCV(forest, param_grid, cv=5,
#                             scoring='accuracy')
# grid_search.fit(x_train, y_train)
# print(grid_search.best_params_)
# print(grid_search.best_estimator_)
# print出的結果
# {'n_estimators': 10, 'n_jobs': -1}
# RandomForestClassifier(n_estimators=10, n_jobs=-1)

# forest = RandomForestClassifier(n_estimators=10, n_jobs=-1)
# results = clf_model(forest, x_train, y_train, x_test)
# model_evaluation(y_test, results)

#最後測試用 不用重新建模
final_model = joblib.load('final_model.pkl')
predicted_labels = final_model.predict(x_test)
model_evaluation(y_test, predicted_labels)
```