```
import numpy as np
import jieba
import json
```

```
with open("movie_all_info.json") as movie:
    movie_info=json.load(movie)
with open("stopword.txt","r",encoding="utf8") as f:
    stopword=f.read()
```

電影類別

```
my_label_list=[]
for i in range(len(movie_info)):
    labels=movie_info[i]["label"]
    for j in range(len(labels)):
        if labels[j] not in my_label_list:
            my_label_list.append(labels[j])
```

```
print(len(my_label_list))
```

```
19
```

訓練集

```
train_data=[]
for i in range(len(movie_info)):
    if len(movie_info[i]["label"])!=0:
        train_data.append(movie_info[i])
```

```
print(len(train_data))
```

```
5924
```

處理訓練資料

```
clean_data=[]
for j in range(len(train_data)):
    terms =[t for t in jieba.cut(train_data[j]["intro"],cut_all=False) if t not in stopword ]
    clean_data.append({"id":train_data[j]["id"],"label":train_data[j]["label"][0],"terms":terms})
```

```
label_list=[]
for i in range(len(clean_data)):
    label_list.append(clean_data[i]["label"])
print(label_list)
```

```
['劇情', '劇情', '恐怖', '冒險', '紀錄片', '劇情', '劇情', '劇情', '劇情', '喜劇', '恐怖', '喜劇', '喜劇', '劇情', '劇情', '紀錄片', '喜劇', '劇
```

把電影類型進行編碼

```
clean=[]
for i in range(len(clean_data)):
    context=""
    for h in range(len(clean_data[i]["terms"])):
        context+=clean_data[i]["terms"][h]
    clean.append(context)
```

```
print(clean[1])
```

```
影展介紹2016開春之際可樂電影往年策劃招牌強檔單元日韓巨星映畫祭藝術人文系列著電影旅行—世界旅程影展於新年度亦展開新嘗試將帶領觀眾看見電影更可能性
```

```
# class_data=[[]]*19
# for i in range(len(clean_data)):
#     for j in range(len(my_label_list)):
#         if(clean_data[i]["label"]==my_label_list[j]):
#             if len(class_data[j])==0:
#                 class_data[j]=clean_data[i]["terms"]
#             else:
#                 class_data[j]+=clean_data[i]["terms"]
#             break
```

```
from sklearn.preprocessing import MultiLabelBinarizer
multilabel_binary=MultiLabelBinarizer()
multilabel_binary.fit(label_list)
y=multilabel_binary.transform(label_list)
```

清理數據的特徵

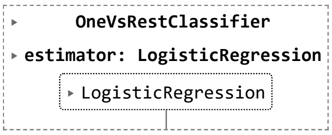
```
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split
tfidf_vec=TfidfVectorizer(max_df=0.8,max_features=10000)
xtrain,xval,ytrain,yval=train_test_split(clean,y,random_state=87, test_size=0.1)
len(xval)
```

```
593
```

```
from itertools import chain
xtrain_tfidf=tfidf_vec.fit_transform(xtrain)
xval_tfidf=tfidf_vec.transform(xval)
```

```
from sklearn.linear_model import LogisticRegression
from sklearn.multiclass import OneVsRestClassifier
from sklearn.metrics import f1_score
lr=LogisticRegression()
clf=OneVsRestClassifier(lr)
```

```
clf.fit(xtrain_tfidf,ytrain)
```



```
▸ OneVsRestClassifier
▸ estimator: LogisticRegression
    ▸ LogisticRegression
```

```
t = 0.4
y_pred = clf.predict(xval_tfidf)
y_pred_prob = clf.predict_proba(xval_tfidf)

y_pred_new = (y_pred_prob >= t).astype(int)
value=f1_score(yval, y_pred_new, average="micro")*2
```

```
print("precision;",value)
```

```
precision; 0.616012238653748
```

✓ 0 秒 完成時間：晚上11:31 ● ✕