

# 自行评测通过的样例数目截图



```
7 data_path = "./data"
8 data = pd.read_json(data_path+"/测试集.json", lines=True)
9 MODEL_NAME = "distilbert-base-uncased"
10 MAX_SEQ_LENGTH = 256
11 NUM_CLASSES = 42
12
13 label_dict = {'ARTS': 0, 'ARTS & CULTURE': 1, 'BLACK VOICES': 2, 'BUSINESS': 3, 'COLLEGE': 4, 'COMEDY': 5, 'CRIME': 6, 'CULTURE & ARTS': 7, 'DIVORCE': 8, 'EDUCATION': 9, 'ENTERTAINMENT': 10, 'ENVIRONMENT': 11, 'FIFTY': 12, 'FOOD & DRINK': 13, 'GOOD NEWS': 14, 'GREEN': 15, 'HEALTHY LIVING': 16, 'HOME & LIVING': 17, 'IMPACT': 18, 'LATINO VOICES': 19, 'MEDIA': 20, 'MONEY': 21, 'PARENTING': 22, 'PARENTS': 23, 'POLITICS': 24}
14 label_dict_inverted = {
15     0: 'ARTS', 1: 'ARTS & CULTURE', 2: 'BLACK VOICES', 3: 'BUSINESS', 4: 'COLLEGE', 5: 'COMEDY', 6: 'CRIME',
16     7: 'CULTURE & ARTS', 8: 'DIVORCE', 9: 'EDUCATION', 10: 'ENTERTAINMENT', 11: 'ENVIRONMENT', 12: 'FIFTY',
17     13: 'FOOD & DRINK', 14: 'GOOD NEWS', 15: 'GREEN', 16: 'HEALTHY LIVING', 17: 'HOME & LIVING', 18: 'IMPACT',
18     19: 'LATINO VOICES', 20: 'MEDIA', 21: 'MONEY', 22: 'PARENTING', 23: 'PARENTS', 24: 'POLITICS',
19 }
20 if __name__ == "__main__":
21     # Your code here
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

```
D:\Develop\anaconda3\envs\newsClassification\python.exe D:\Study\Codes\BigData\newsClassification\test.py
correct_samples: 521, total_samples: 619
Test Accuracy: 0.8416801292407108, Test Precision: 0.7666002858619236, Test F1 Score: 0.654271591257634
进程已结束，退出代码为 0
```

评测通过的样例数为521，总样例数为619

## 运行说明

我的复现问题是：**没有模型文件**

模型文件在**该链接**中可以找到：<https://box.nju.edu.cn/d/7abb838012ff4c5d8338/>

进行测试时选择下载名为 `trained_model_best.pth` 的模型文件（约268M）

将该模型放在results文件夹下，并在测试文件中设置为对应的路径，即可开始测试

# 替换成你想要加载的模型文件名

```
loaded_model.load_state_dict(torch.load('results/trained_model_best.pth'))
```

原本的测试代码不会输出正确的样例数，只会输出准确度和精确度，添加以下代码即可（已经在压缩文件中附上修改后的代码）

# 直接比较预测标签和真实标签

```
correct_samples = np.sum(np.array(predicted_labels) == np.array(true_labels))
```

```
print(f"correct_samples: {correct_samples}, total_samples: {len(true_labels)}")
```

- 注意

- 测试文件的运行需要导入 `some_classes.py` 的类，请确保它们在同一工作目录下
- 在main中调用`cal_acc_test()`方法测试整个数据集
- 加载模型到内存需要**花费一段时间**，当模型加载完毕后，能够**很快**地继续预测