ncomms2022——多模态dementia诊断

一、任务目标

通过图像数据（核磁共振MRI影像）和非影像数据（主要是临床数据）对认知状态进行分类。

NC——正常认知

MCI——轻度认知障碍

AD——阿兹海默痴呆

nADD——非阿兹海默痴呆

ncomms2022项目主要将任务分为两块：一是COG任务（三分类），分类NC、MRI、DE；二是ADD任务（二分类），是对COG任务中的DE进行细分，AD和nADD。

二、环境配置

Pytorch>=1.10

Numpy>=1.19

TQDM>=4.31

Nibabel>=3.2

matplotlib>=3.3

scikit-learn>=0.23

scipy>=1.5.4

SHAP>=0.37

XGBoosh>=1.3.3

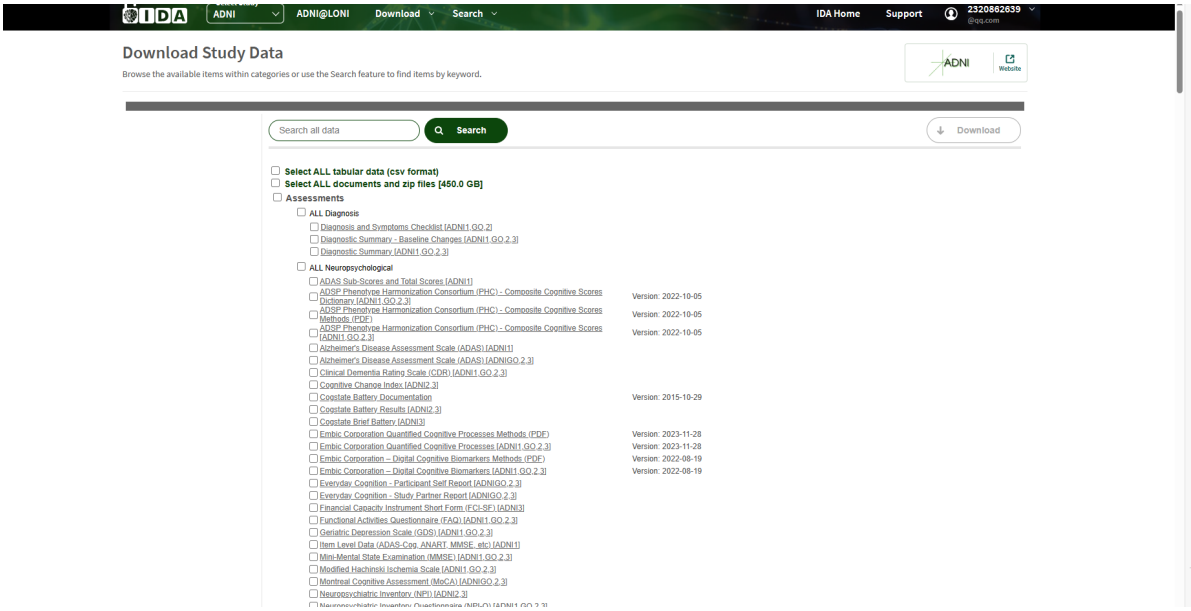
catboost>=0.24

三、数据集下载

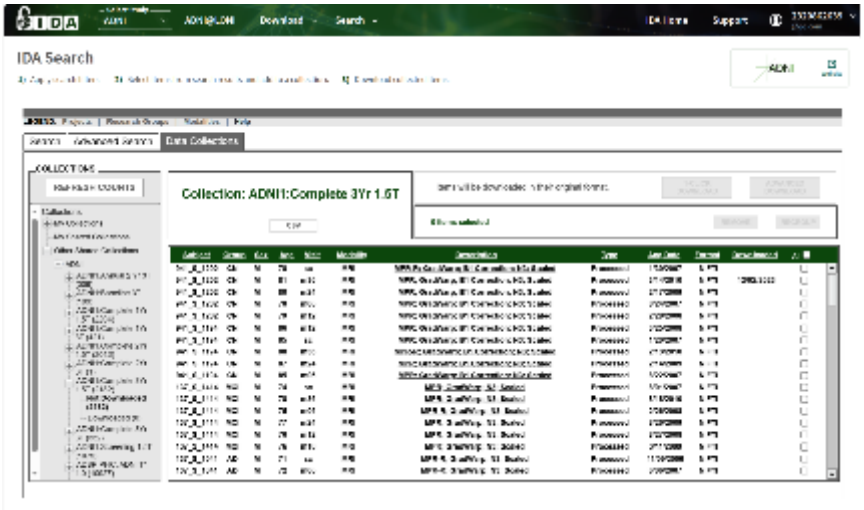
1、ADNI

非影像数据：需要慢慢筛选。（网站上的名称和实际下载的文件名不一致，需要慢慢筛选）

```
The following files need to be downloaded and saved in this folder
ADAS_ADNIGO23.csv                APOERES.csv   FHQ.csv       MRIMETA.csv
ADASSCORES.csv                   BLCHANGE.csv  GDSCALE.csv   NEUROBAT.csv
ADNI_DXSUM_PDXCONV.csv           CDR.csv       ITEM.csv      NPI.csv
ADNIMERGE.csv                    EXCLUSIO.csv  MEDHIST.csv   NPIQ.csv
ADNI_MMSE.csv                    FAMXHPAR.csv  MOCA.csv      PTDEMOG.csv
ADNI_NEUROPATH_04_12_18.csv      FAMXHSIB.csv  MODHACH.csv
ADNI_NEUROPATH_MRI_04_12_18.csv  FAQ.csv       MRI3META.csv
```

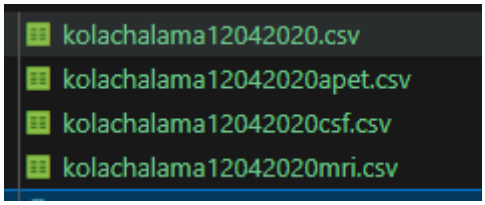


影像数据：原论文及项目中未提及具体如何选取。在ADNI官网中随机挑选了一个共享集合进行下载。数据量大且不知道是否与非图像数据匹配。

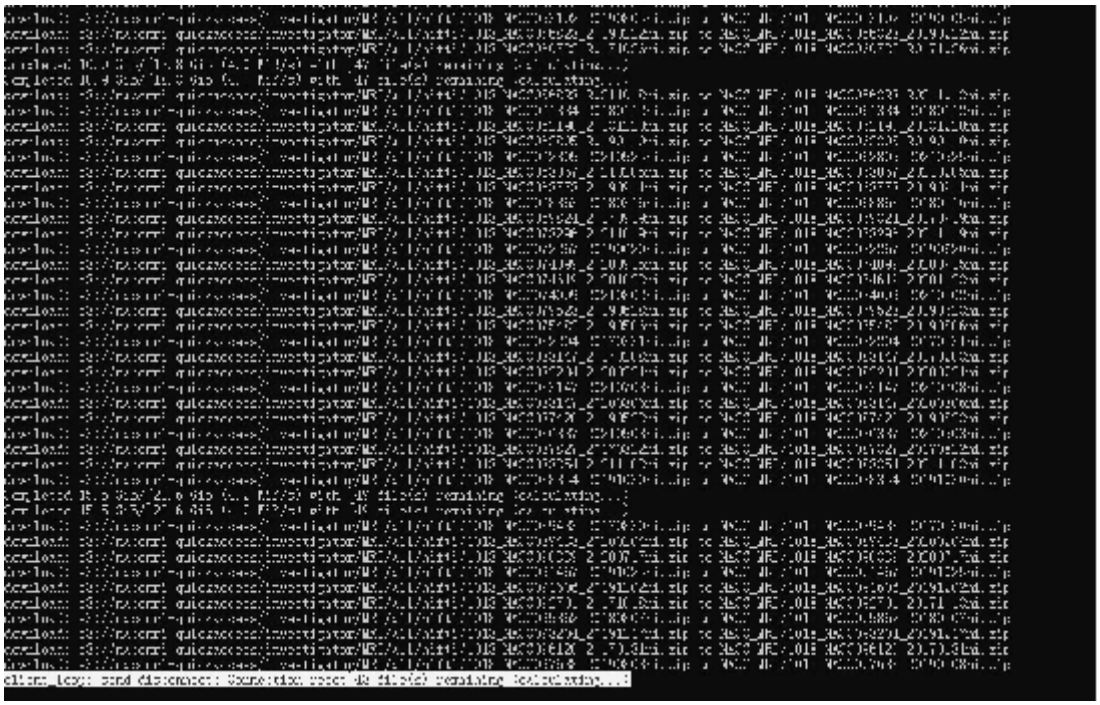


2、NACC

非影像数据：需要向机构申请



影像数据：数据量大（预计25G以上）且不知道是否与非图像数据匹配。原论文及项目中未提及具体如何选取。（目前下载失败）



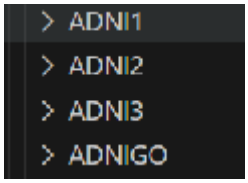
四、数据预处理及训练集划分

1、进入derived_tables文件夹，根据不同数据集生成元数据。

```
run the 3 python scripts to generate derived meta information
for MOCA, MMSE and diagnosis
```

2、进入datasets_tables文件夹，为不同数据集准备最终的元表。

```
run the ADNI1.py to get the meta table for the ADNI1 data
```



```
class TableData:
    def __init__(self):
        self.datasetName = 'ADNI1'
        self.imageDir = 'data/ADNI1/'
        self.imageFileNameList, self.RIDList = self.get_filenames_and_IDs(self.imageDir)
        print(self.imageFileNameList)
        self.columnNames = []
        self.content = defaultdict(dict) # dictionary of dictionary; {RID: {colname1: val1, colname2: val2, ...}

    def get_filenames_and_IDs(self, path):
        fullpathList = glob(path + '*.nii')
        print(fullpathList)
        fileNameList = [fullpath.split('/')[-1] for fullpath in fullpathList]
        IDList = [filename[5:15] for filename in fileNameList]
        RIDList = [id[-4:].lstrip("0") for id in IDList]
        return fileNameList, RIDList
```

其中需要以图片路径作为索引，选取记录元素，并得到列名

3、进入CrossVaild文件夹，划分训练集、测试集、验证集。

```
this CrossValid folder will contain 5 fold cross validation split information

step1, run the combine.py to combine all subjects from all cohorts and output results in all.csv

step2, run the split.py to split the NACC into train, valid, test, and consider all other cohorts
as external testing, the data split results will be stored in folder cross0/ cross1/ etc

step3, run appendNonImage.py to fill up non-imaging information by table joining
```

combine.py将所有数据集整合到一个all.csv文件中。

split.py将all.csv划分为训练集、测试集、验证集。

appendNonImage.py将所有非图像元素加入到训练集、测试集、验证集中。（列名）

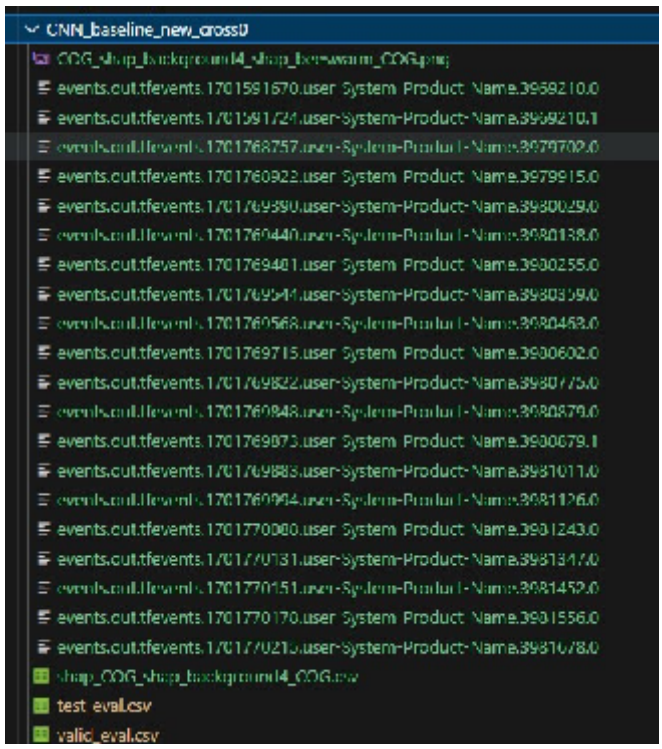
```
import csv

# tables = ['NACC', 'ADNI1', 'ADNI2', 'ADNI3', 'ADNIGO', 'NIFD', 'PPMI', 'AIBL', 'OASIS', 'FHS', 'Stanford']
tables = ['ADNI1', 'ADNI2', 'ADNI3']
column_names = ['path', 'filename', 'NC', 'MCI', 'DE', 'COG', 'AD', 'PD', 'FTD', 'VD', 'DLB', 'PDD', 'ADD', 'ALL', 'OTHER']

def read_csv_dict(content, csv_table):
    with open(csv_table, 'r') as f:
        reader = csv.DictReader(f)
        for row in reader:
            content.append(row)

content = []

for table in tables:
    if table == 'NACC':
        csv_path = '../dataset_table/NACC_ALL/' + table + '.csv'
    else:
        csv_path = '../dataset_table/' + table + '/' + table + '.csv'
    read_csv_dict(content, csv_path)

with open('all.csv', 'w', newline='') as csvfile:
    spamwriter = csv.writer(csvfile)
    spamwriter.writerow(column_names)
    for row in content:
        if row['NC'] == '1':
            row['ALL'] = '0'
        elif row['MCI'] == '1':
            row['ALL'] = '1'
        elif row['AD'] == '1':
            row['ALL'] = '2'
        elif row['ADD'] == '0':
            row['ALL'] = '3'
        spamwriter.writerow([row[col_name] if col_name in row else '' for col_name in column_names])
```

五、训练模型(NoImg)

准备好训练集、测试集、验证集后（目前仅有ADNI1中的1/10），训练CNN模型（由于没有NACC，只能COG任务）（test.ipynb）

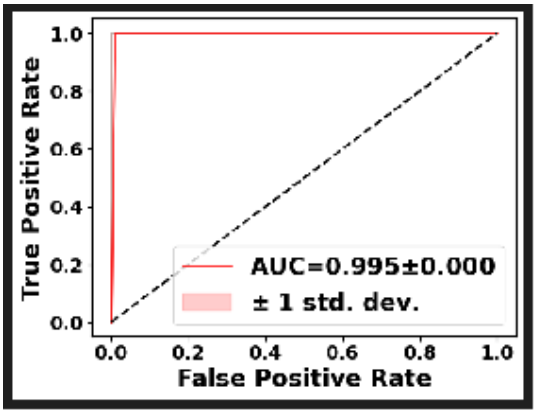
```python
from nonImg_model_wrappers import NonImg_Model_Wrapper, Fusion_Model_Wrapper
from utils import read_json

model = NonImg_Model_Wrapper(
    tasks=['COG'],                          # a list of tasks to predict
    main_config={"save_dir": "./output/cv/CrossValid/cross0/", "model_name": "CNN_baseline_new_cross0"},          # general configuration for the experiment
    task_config=read_json('nonimg_task_config.json'),       # task specific configurations
    seed=1000
)
model.train()
thres = model.get_optimal_thres()                   # get optimal threshold using validation dataset
model.gen_score(['test'], thres)                    # apply optimal threshold on test dataset and cache predictions
```

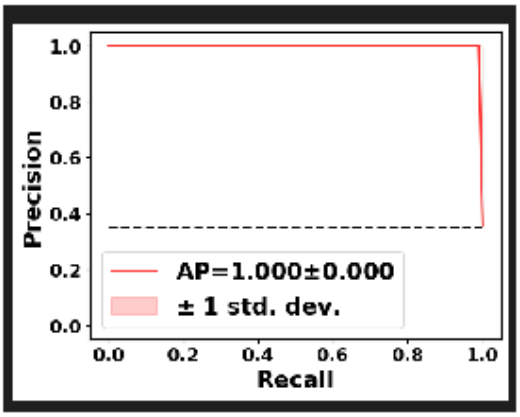
nonImg_task_config.json选取合适特征，保证特征维度匹配

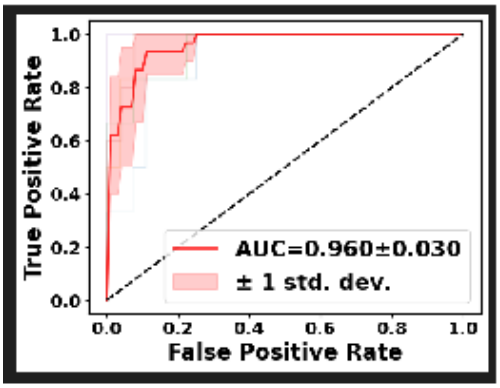


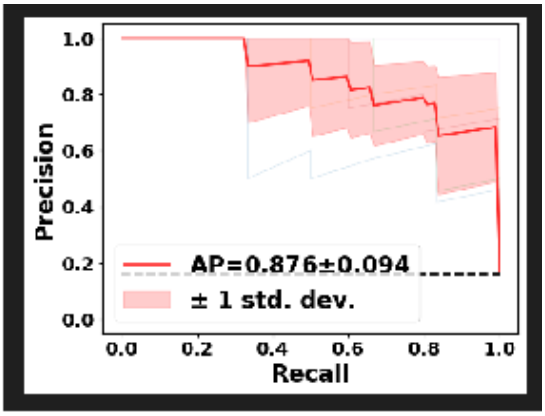
绘制ROC曲线及PR曲线

```python
from performance_eval import generate_roc, generate_pr
generate_roc(
    ['tb_log/CNN_baseline_new_cross0/test_eval.csv','tb_log/CNN_baseline_new_cross1/test_eval.csv','tb_log/CNN_baseline_new_cross2/test_eval.csv','t
    "DL",
    'red',
    'ADNI test roc'
)
generate_pr(
    ['tb_log/CNN_baseline_new_cross0/test_eval.csv','tb_log/CNN_baseline_new_cross1/test_eval.csv','tb_log/CNN_baseline_new_cross2/test_eval.csv','t
    "DL",
    'red',
    'ADNI test pr'
)
```









Shap可解释分析

```
shap_values, _ = model.shap("COG_shap_background4")
```