|  |
| --- |
| 数据预处理 |
| import pandas as pd  import numpy as np  from matplotlib import pyplot as plt  part = 2  if part == 1:  dt1 = pd.read\_excel("附件一-电视剧评估信息.xlsx",'Sheet1')  dt2 = pd.read\_excel("附件二-电视剧基本信息.xlsx",'Sheet1')  writer = pd.ExcelWriter("result.xlsx")  #print(dt1.merge(dt2,left\_on='TV Drama',right\_on='TV Drama'))  dtret = dt1.merge(dt2,*left\_on*='TV Drama',*right\_on*='TV Drama')  dtret.to\_excel(writer,'Sheet1')  writer.save()  writer.close()  classes = {}  for i in dt1.groupby('Theme'):  classes.update({i[0]:i[1]})  print(str([i for i in classes.keys()]))  cnt\_drama = [0 for i in range(12)]  cnt\_drama\_comments = [0 for i in range(12)]  for i,j in zip(dtret.sort\_values(*by*='Date of Issuance License')['Date of Issuance License'].reindex(),  dtret.sort\_values(*by*='Date of Issuance License')['Number of Comments'].reindex()):  cnt\_drama[int(i.split("/")[1])-1]+=1  cnt\_drama\_comments[int(i.split("/")[1])-1]+=j  #print(dtret[dtret['Score'] == 0])  month = [i for i in range(1,13)]  plt.bar(month,cnt\_drama)  plt.show()  dt = pd.read\_excel('result.xlsx','Sheet1')  comment\_max = dt['Number of Comments'].max()  Score\_max = dt['Score'].max()  Episode\_max = dt['Episode'].max()  Info\_total = []  for i in dt.index:  Info\_new = []  for j in dt.columns:  if j == 'TV Drama':  Info\_new.append(dt[j][i])  if j == 'Score':  try:  a = float(dt[j][i])  except:  a = 0  Info\_new.append(a)  elif j == 'Number of Comments':  try:  a = float(dt[j][i]/comment\_max)  except:  a = 0  Info\_new.append(dt[j][i])  Info\_new.append(a)  elif j == 'Episode':  try:  a = float(dt[j][i]/Episode\_max)  except:  a = 0  Info\_new.append(dt[j][i])  Info\_new.append(a)  Info\_total.append(Info\_new)  for i in Info\_total:  Rscore = i[1]\*i[2] - 0.2 \* i[3]  i.append(Rscore)  Info\_dt = pd.DataFrame(Info\_total)  Info\_dt.columns = ['TV Drama','Score','Number of Comments','Chs','Episode','Ehs','RScore']  print(Info\_dt.sort\_values(*by*='RScore',*ascending*=False).head(10)) |

|  |
| --- |
| 明星筛选 |
| import os  import shutil  import re  from bs4 import BeautifulSoup  listdir = os.listdir(os.getcwd())  if not os.path.exists("hm"):  if not os.path.isdir("hm"):  os.makedirs(os.path.join(os.getcwd(),"hm"),*mode*=0o777)  print("Success create the html folder")  else:  print("Warnning: Html dir is already existed!")  else:  print("Warnning: Html is already existed!")  try:  for name in listdir:  if re.match(r'.\*\_hm.txt',name) is not None:  shutil.move(name,"hm/")  print (name + "-->" + os.getcwd()+"/hm/"+name,end="\n")  print("Move files successfully")  except:  print("Already done!")  pass  stars\_bd\_file\_list = os.listdir("html")  for file in stars\_bd\_file\_list:  try:  with open(os.path.join(os.getcwd(),"html",file),"r") as f:  line = 0  try:  for i in f.readlines():  line+=1  except:  pass  print("length:"+str(line))  if line <= 10:  f.close()  os.remove(os.path.join(os.getcwd(),"html",file))  continue  bsdt = BeautifulSoup(f.read(),"html.parser",*from\_encoding*='utf-8')  try:  for i in bsdt.find(*attrs*={"class":re.compile("hint\_[a-zA-Z0-9]{0,9} c\_font\_[a-zA-Z0-9]{0,9}")}):  print(i)  except:  print(file)  pass  except:  print("utf-8"+file) |

|  |
| --- |
| 百度信息搜集 |
| import requests  import time  import pandas as pd  import numpy as np  import os  import json  name = "白雪"  url = r'http://index.baidu.com/api/FeedSearchApi/getFeedIndex?word=[[{"name":"'+name+'","wordType":1}]]&area=0&days=30'  print(url)  headers = {  "Host": 'index.baidu.com',  'User-Agent': 'Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:94.0) Gecko/20100101 Firefox/94.0',  'Accept': 'application/json, text/plain, \*/\*',  'Accept-Language': 'zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2',  'Accept-Encoding': 'gzip, deflate, br',  'Referer': 'https://index.baidu.com/v2/main/index.html',  'Connection': 'keep-alive',  'Cookie': 'BAIDUID=6EE4B14CF0C5FFBD8BC51FE2BD5CC1A2:FG=1; BIDUPSID=6EE4B14CF0C5FFBDCDDAA3478440FB86; PSTM=1628165068; \_\_yjs\_duid=1\_6bfdb0ac0e3760480bbcac4da6e828871628176157265; BDUSS=X5NT1F4RjA2UmN4WnBOaVNkdX5QOC1iM2Nad3kyMmEzVUFmMmgxM1AzRUw4Mk5oRVFBQUFBJCQAAAAAAAAAAAEAAACPk8sQt67U88jwAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAtmPGELZjxhV; BDORZ=FFFB88E999055A3F8A630C64834BD6D0; H\_PS\_PSSID=35295\_35104\_31254\_35435\_35456\_34584\_34504\_34606\_35329\_35315\_26350; BDSFRCVID=hFtOJexroG04sxcH4wwF-tse4ec\_23JTDYLEL5mveYl6oVFVJeC6EG0Ptq6Ex2P-EHtdogKKy2OTH9DF\_2uxOjjg8UtVJeC6EG0Ptf8g0M5; H\_BDCLCKID\_SF=tbAOVCD2JKL3fP36q4OHht\_tMq0X5-RLf273ah7F5l8-hCb6b-IKXUv3etoMeR3e5RnB3qrCBD3xOKQphUIhyjO-04FLBf0qbCJLLnbN3KJm8tP9bT3vLfuyjtcz2-biW5KH2MbdflbP\_IoG2Mn8M4bb3qOpBtQmJeTxoUJ25DnJhbLGe6L3-RJH-xQ0KnLXKKOLVKDyWl7ketn4hUt2jbtb-PAJ-bbjH6ne0CbHKMAKhpc2QhrdQf4WWb3ebTJr32Qr-qcH0KQpsIJM5bL-QJvX02c4Bn3baKvia-I-BMb1bqbDBT5h2M4qMxtOLR3pWDTm\_q5TtUJMeCnTDMFhe6oM-Uvy-40XKD600PK8Kb7VbnnD5MnkbJkXhPtjtnJOK2-JL-3IH4b5eDbN-pPKDUI7Qbrr0xAta6c-hR4KaMQPSlcNLTjpQT8r5hLq0RkHLPj4aPbeab3vOPI4XpO1ef0zBN5thURB2DkO-4bCWJ5TMl5jDh3Mb6ksD-FtqtJHKbDH\_KPMtMK; ab\_sr=1.0.1\_Mjk1MjAzY2E5NTZkODAxY2NiNDQ5YmY4N2U5NzExOWNmZmQ0OWE3YTM4ZGUxMmI1MzNjMTYxMDIxMWQxZWU3NmU5MTFlYzc2NTQ0MmQzYjZmZDVjNzQwMzE5NjQ4Yjk2ZjFmMmI3NGJhMGQ4NTlmN2FiZmMyOWIwYzgxZTI0NjAwMjkyYTBlODA3ODU2NjNkN2QyNTRhMDc1M2E0M2FhMA==; BA\_HECTOR=a021248020058524cd1gr6nok0r; delPer=0; PSINO=6; BDRCVFR[gltLrB7qNCt]=mk3SLVN4HKm; Hm\_lvt\_d101ea4d2a5c67dab98251f0b5de24dc=1639145239; Hm\_lpvt\_d101ea4d2a5c67dab98251f0b5de24dc=1639145471; RT="z=1&dm=baidu.com&si=6rc7piqj5be&ss=kx0gr626&sl=d&tt=6o8&bcn=https%3A%2F%2Ffclog.baidu.com%2Flog%2Fweirwood%3Ftype%3Dperf&ld=5109"; \_\_yjs\_st=2\_ZmE0MWQ4MjVkYmQwMzcyYWZlNGYzNDY0MGFlMGNlMTdiZTBmZDIyNzNhZTQyNWM0NWQ1OTRlZDE0MjFlNzhhMWU5OWUyZDdmZjE3MTgxNjBhOWY5YTNiMGQxYjIzYTU5ZDRmOWRhMTgyOWFkODFkOGMzOTQ0NTE5N2IxZGFmODY1YzUzM2NiZDUwOGE2YTVkMzQ5YzljMDk2Y2RkZmQwY2QwNDVhNTNmNmYyZWQ1Zjk4YWMzYTRiODM4NDAzMjNhNDdlZmUwMjE4N2M4YjY3Y2JkMzMzMjBlOWZhMDQ1Yzk0YmNmYzlkM2EwZWU3NjA0Zjg5YTgzNGMwZGQ2NWE3MmJkMzYxNGRiNjZlZDJiOGM2NGMwNzAwNzgxZWZlMzMzXzdfNDNkM2FjNmQ=; bdindexid=vo1c73ivop6an83hkaavraogl3',  'Sec-Fetch-Dest': 'empty',  'Sec-Fetch-Mode': 'no-cors',  'Sec-Fetch-Site': 'same-origin',  'Pragma': 'no-cache',  'Cache-Control': 'no-cache'  }  dt = pd.read\_excel('result.xlsx','Sheet1')  stars = set()  for names in dt['Starring']:  if names is not np.nan:  for name in str(names).replace('、',',').replace('，',',').split(','):  stars.add(name)  total\_file = open("summary\_hm.txt","a+")  total\_file.seek(0,0)  for name in stars:  time.sleep(1)  try:  if os.path.exists(name+'\_hm.txt'):  continue  with open(name+'\_hm.txt','w') as f:  time.sleep(1)  url = r'http://index.baidu.com/api/FeedSearchApi/getFeedIndex?word=[[{"name":"'+name+'","wordType":1}]]&area=0&startDate=2021-11-10&endDate=2021-12-09'  print(url)  r = requests.get(*url* = url,*headers* = headers)  r.raise\_for\_status()  r.encoding = r.apparent\_encoding  ret = str(r.text)  ret = json.loads(ret)  print(name+','+str(dict(ret['data'])['index'][0]['generalRatio'])+'Success')  f.write(name+','+str(dict(ret['data'])['index'][0]['generalRatio'])+'\n')  total\_file.write(name+','+str(dict(ret['data'])['index'][0]['generalRatio'])+'\n')  except:  print(name + "Failure")  continue  # for name in stars:  # time.sleep(1)  # try:  # if os.path.exists(name+'\_hm.html'):  # print(name,"Success")  # continue  # with open(name+'\_hm.html','w') as f:  # time.sleep(1)  # url = r'http://index.baidu.com/api/FeedSearchApi/getFeedIndex?word=[[{"name":"'+name+'","wordType":1}]]&area=0&days=30'  # r = requests.get(url = url,headers = headers)  # r.raise\_for\_status()  # r.encoding = r.apparent\_encoding  # ret = str(r.text)  # print(ret)  # f.write(ret)  # print(name,"Success")  # except:  # print(name,"Failure")  # continue |

|  |
| --- |
| 新浪微博信息收集 |
| import pandas as pd  import numpy as np  import os  import time  import matplotlib.pyplot as plt  from bs4 import BeautifulSoup  import requests  from selenium.webdriver import Chrome  from selenium.webdriver.chrome.options import Options  from selenium.webdriver.support.wait import WebDriverWait  from selenium.webdriver import \*  from selenium import webdriver  from selenium.common.exceptions import TimeoutException  from selenium.webdriver.support.wait import WebDriverWait  from urllib.parse import quote  from pyquery import PyQuery as pq  dt = pd.read\_excel('result.xlsx','Sheet1')  stars = set()  for names in dt['Starring']:  if names is not np.nan:  for name in str(names).replace('、',',').replace('，',',').split(','):  stars.add(name)    url = "https://s.weibo.com/weibo?q="  headers = {  "Host": "s.weibo.com",  "Accept": "text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,\*/\*;q=0.8",  "Accept-Language": "zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2",  "Accept-Encoding": "gzip, deflate, br",  "Connection": "keep-alive",  "Cookie": "www.52jingsai.com,widget.weibo.com,www.baidu.com; SUBP=0033WrSXqPxfM725Ws9jqgMF55529P9D9WWbjVd7EgcvSe75XM2kBhOX5JpX5KMhUgL.Foe4eh2pSo.7S0-2dJLoIp7LxKML1KBLBKnLxKqL1hnLBoM01K5peKq4ehMf; SINAGLOBAL=8505429497315.529.1632922843183; ULV=1639136383200:2:1:1:3763013819565.094.1639136383093:1632922843184; ALF=1670672351; SCF=AoMImLiEPADgRCx1ffu6aj6AK92GFULp5oaCrYYu3UyFQ1CaK7GE8lZnOmRTsnXxymUaky46S2SrG5WmaKwfqDc.; SUB=\_2A25Mt0wwDeRhGeVH61MQ9ifMzDmIHXVvxTr4rDV8PUNbmtAKLUGtkW9NT2jo5JhLkwNEFUwWQ\_iTWmiOS-eqHHhv; SSOLoginState=1639136352; \_s\_tentry=www.baidu.com; Apache=3763013819565.094.1639136383093",  "Upgrade-Insecure-Requests": "1",  "Sec-Fetch-Dest": "document",  "Sec-Fetch-Mode": "navigate",  "Sec-Fetch-Site": "cross-site",  "User-Agent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:94.0) Gecko/20100101 Firefox/94.0",  }  # for name in stars:  # time.sleep(1)  # try:  # if os.path.exists(name+'\_sina.html'):  # continue  # # with open(name+'\_sina.html','w') as f:  # time.sleep(1)  # r = requests.get(url = url+name,headers = headers)  # r.raise\_for\_status()  # r.encoding = r.apparent\_encoding  # ret = str(r.text)  # print(ret)  # #f.write(ret)  # except:  # continue  for name in stars:  time.sleep(1)  try:  if os.path.exists(name+'\_sina.html'):  print(name,"Success")  continue  with open(name+'\_sina.html','w') as f:  time.sleep(1)  r = requests.get(*url* = url+name,*headers* = headers)  r.raise\_for\_status()  r.encoding = r.apparent\_encoding  ret = str(r.text)  # print(ret)  f.write(ret)  print(name,"Success")  except:  print(name,"Failure")  continue |

|  |
| --- |
| 新浪微博文件处理 |
| from bs4 import BeautifulSoup  import os  import re  list\_file = os.listdir(os.getcwd())  for file\_name in list\_file:  if re.match(r'.\*.\_sina.html',file\_name) is None:  list\_file.remove(file\_name)  total\_fans\_cnt = open("total\_fans\_cnt.txt","a+")  total\_fans\_cnt.seek(0,0)  for file in list\_file:  try:  with open(file,"r") as f:  name = file.split('\_')[0]  bsdt = BeautifulSoup(f.read(),"html.parser",*from\_encoding*="utf-8")  if bsdt.find("span",*attrs*={'class':'s-nobr'}) is not None:  print(name+" "+str(bsdt.find("span",*attrs*={'class':'s-nobr'}).contents[0]))  total\_fans\_cnt.write(name+" "+str(bsdt.find("span",*attrs*={'class':'s-nobr'}).contents[0])+'\n')  except:  continue  print(list\_file) |

|  |
| --- |
| 明星排位 |
| import os  import re  import json  import ast  filelist = os.listdir(os.getcwd())  for i in filelist:  if re.match(r'.\*\_hm.txt',i) is None:  filelist.remove(i)    total = []  for file in filelist:  with open(file,"r") as f:  for line in f.readlines():  splited = line.split(",")  str1 = "".join(','+splited[i] for i in range(1,len(splited)))  ret = ast.literal\_eval(str(str1[1:]).replace(" ","").replace('\n',''))  ret.update({'name':splited[0]})  print(ret)  total.append(ret)    for i in sorted(total,*key*=lambda *x*:x['avg'],*reverse*=True)[:34]:  print(i['name']) |