一、代码

爬虫

import requests

import json

import pandas as pd

import time

import random

newdata=[]

headers={

'user-agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/79.0.3945.88 Safari/537.36'}

for i in range(1,20):

dat={'access\_play\_control\_platform': '14',

'channel\_id': '2',

'data\_type': '1',

'from': 'pcw\_list',

'is\_album\_finished':'' ,

'is\_purchase': '',

'key': '',

'market\_release\_date\_level': '',

'mode': '24',

'pageNum': i,

'pageSize': '48',

'site': 'iqiyi',

'source\_type': '',

'three\_category\_id': '',

'without\_qipu': '1'}

url = "https://pcw-api.iqiyi.com/search/video/videolists?access\_play\_control\_platform=14&channel\_id=2&data\_type=1&from=pcw\_list&is\_album\_finished=&is\_purchase=&key=&market\_release\_date\_level=&mode=24&pageNum="+str(i)+"&pageSize=48&site=iqiyi&source\_type=&three\_category\_id=&without\_qipu=1"

html=requests.get(url=url,data=dat,headers=headers)

newdata.append(html.json()['data']['list'])

i=i+1

time.sleep(random.random()\*10)

from itertools import chain

data=list(chain.from\_iterable(newdata))

nda=pd.DataFrame(data)

nda.to\_csv(r'C:\Users\tong.shao\Desktop\bylw\aqycontent.csv',encoding='utf\_8\_sig')

二、计算机相关知识

关联分析是一种简单、实用的分析技术，就是发现存在于大量数据集中的关联性或相关性，从而描述了一个事物中某些属性同时出现的规律和模式。

关联分析是从大量数据中发现项集之间有趣的关联和相关联系。关联分析的一个典型例子是购物篮分析。该过程通过发现顾客放入其购物篮中的不同商品之间的联系，分析顾客的购买习惯。通过了解哪些商品频繁地被顾客同时购买，这种关联的发现可以帮助零售商制定营销策略。其他的应用还包括价目表设计、商品促销、商品的排放和基于购买模式的顾客划分。

关联规则其实是两个项集之间的蕴涵表达式。如果我们有两个不相交的项集X和Y，就可以有规则X→Y, 例如｛Bread，Milk｝→{Diaper}。项集和项集之间组合可以产生很多规则，但不是每个规则都是有用的，我们需要一些限定条件来帮助我们找到强度高的规则。

支持度(s)

关联规则的支持度定义为：也就是同时包含X和Y这两个项集的事务占所有事务的比例。我们看｛Bread，Milk｝→{Diaper}这个例子，同时包含｛Bread，Milk，Diaper}这个项集的事务一共有2项，因此这个规则的支持度是。

置信度(c)

关联规则的置信度定义为：这个定义确定的是Y在包含X的事务中出现的频繁程度。还是看｛Bread，Milk｝→{Diaper}这个例子，包含｛Bread，Milk｝项的事务出现了2次，包含｛Bread，Milk，Diaper}的事务也出现了2次，那么这个规则的置信度就是1。

对于关联规则定义这两个度量很有意义的。首先，通过对规则支持度支持度的限定滤去没有意义的规则。我们从商家的角度出发，数据挖掘意义是通过挖掘做出相应的战略决策产生价值。如果一个规则支持度很低，说明顾客同时购买这些商品的次数很少，商家针对这个规则做决策几乎没有意义。其次，置信度越大说明这个规则越可靠。

**Apriori定律1**：如果一个集合是频繁项集，则它的所有子集都是频繁项集。

例如：假设一个集合{A,B}是频繁项集，即A、B同时出现在一条记录的次数大于等于最小支持度min\_support，则它的子集{A},{B}出现次数必定大于等于min\_support，即它的子集都是频繁项集。

**Apriori定律2：**如果一个集合不是频繁项集，则它的所有超集都不是频繁项集。

举例：假设集合{A}不是频繁项集，即A出现的次数小于 min\_support，则它的任何超集如{A,B}出现的次数必定小于min\_support，因此其超集必定也不是频繁项集。

三、开源项目

1、上传每日的作业和leetcode题解

2、提交作业的pr并merge