山东大学 软件 学院

推荐系统整体课程实验报告

|  |  |  |  |
| --- | --- | --- | --- |
| 学号：201700301022 | 姓名：唐佳文 | | 班级： AI17 |
| 实验题目：美食推荐 | | | |
| 实验学时： | | 实验日期： 2020年四月至五月 | |
| 实验目的：通过相关外卖数据进行外卖推荐，后来因为隐私问题变为美食推荐 | | | |
| 硬件环境：  Win10  Dell xps15 | | | |
| 软件环境：  Vs  VMware 虚拟机  Ubuntu16.0.4 64位  LIBFM  Pycharm | | | |
| 目录  [1任务概述 3](#_Toc41222891)  [1.1背景描述 3](#_Toc41222892)  [1.2 项目目的 3](#_Toc41222893)  [2需求分析 4](#_Toc41222894)  [2.1功能分析 4](#_Toc41222895)  [2.1.1可视化提供推荐界面 4](#_Toc41222896)  [2.1.2明确显示推荐店铺的结果 5](#_Toc41222897)  [2.2 非功能性需求分析 5](#_Toc41222898)  [2.2.1易用性 5](#_Toc41222899)  [2.2.2效率 6](#_Toc41222900)  [2.2.3可扩展性 6](#_Toc41222901)  [3项目制作流程即算法 7](#_Toc41222902)  [3.1数据 7](#_Toc41222903)  [3.1.1数据来源 7](#_Toc41222904)  [3.1.2 获取数据的策略 8](#_Toc41222905)  [3.1.3 数据清洗和整理 15](#_Toc41222906)  [3.1.3 建立评分矩阵 21](#_Toc41222907)  [3.2 项目流程 24](#_Toc41222908)  [3.2.1用户分级制度 24](#_Toc41222909)  [3.2.2用户分级算法 25](#_Toc41222910)  [3.2.2分级的推荐算法 26](#_Toc41222911)  [3.2.3流程图 40](#_Toc41222912)  [3.3 项目成果展示 42](#_Toc41222913)  [4项目评价和改进 44](#_Toc41222914)  [4.1优点 44](#_Toc41222915)  [4.2缺点 45](#_Toc41222916)  [4.3改进 46](#_Toc41222917)  [5分析和体会 50](#_Toc41222918)  [5.1体会 50](#_Toc41222919)  [代码 51](#_Toc41222920) 1任务概述1.1背景描述 收获一个推荐系统任务，结合时下美食外卖消费的兴旺，于是想制作一个简单的美食推荐系统。 1.2 项目目的 利用爬虫等爬虫策略去获取‘美团网’用户行为数据，和对应的店铺信息。经过相应的推荐算法设计和相应的推荐系统界面UI设计，当输入用户id，展示前十条推荐店铺即店铺信息。 2需求分析2.1功能分析2.1.1可视化提供推荐界面 使用Django框架利用Python编程语言进行视图，控制器风格的Web应用设计，在Pycharm中利用Django框架进行推荐系统界面显示设计 2.1.2明确显示推荐店铺的结果 当输入用户id时，能明确按照排序输出推荐店铺结果，店铺结果中应该包含店铺名称，平均得分，人均消费，店铺地点信息 2.2 非功能性需求分析2.2.1易用性 使用Django进行简洁网页开发，可以方便、快捷地创建高品质、易维护、数据库驱动的应用程序，同时要针对不存在id进行冷启动推荐。 2.2.2效率 明确项目流程路径和算法，优化流程和算法从而提高推荐效率。 2.2.3可扩展性 预留其他算法接口，项目可以在后续快速接入其他推荐算法，统一推荐算法的输入数据格式和输出格式内容，便于多重算法兼容推荐系统项目。 3项目制作流程即算法3.1数据3.1.1数据来源 用户行为数据和店铺信息数据来自美团网。一开始想爬行获取美团外卖用户行为数据，获取一座城市的外卖订单信息和用户评价信息来进行外卖推荐，但是在实施时发现一个很大的硬伤，就是无法爬取用户的定位信息，或者说可能自己正在涉及用户隐私等敏感信息，可能在违法的边缘试探，可能实验没做完，人已在派出所。遂放弃了对用户地址信息和电话等个人隐私的爬取。查询网上资料发现订单信息的获取好像需要注册个店铺，也不知道获取这种隐私违不违法。而从一些数据买卖中介看到的外卖信息样本大多主要侧重用户的电话号码等私人信息。  于是作为五讲四美的新青年，选择自主通过爬虫策略和美团网上的美食信息和评论，对用户进行推荐，但不可避免的丢失了一些人口学统计信息。 3.1.2 获取数据的策略 美团网不愧是一个有钱的网站，反爬虫策略很严格，采用了用户验证，动态jax网页刷新等反爬虫策略，网上能找到的爬虫策略比较落后，跟不上美团网的反爬虫策略的更新，一些较新的爬虫策略往往在获取几百条数据后就识别并查封。  一些常见的反爬策略如下，被识别出来后原创关闭连接或者不再返回指定格式的json文件。      第一种封锁的破解比较通用，分析原因，是因为使用urlopen方法太过频繁，引起远程主机的怀疑，被网站认定为是攻击行为。导致urlopen()后，request.read()一直卡死在那里。最后抛出10054异常。于是又去网上学习已经遗忘的爬虫技巧。   * 配置socket信息，不定时开关request，同时是不是让爬虫进程sleep一小会。 * 使用ajax构造url * 设置异常处理，当被服务器拒绝连接时，睡眠一定时间再继续爬取。      * 使用代理ip池   但前四种方法还是收效较小，面对美团的token策略作用不大。   * 也看了些博客，看看其他人登录网页url信息与自己的差别，发现uuid上存在不同，     我突然发现可能在这方面存在突破口，也就是收集不同的uuid，被服务器反爬虫发现一个之后就再换一个进行爬取，积小成多。于是建立uuid容器，里面收集不同的uuid，在循环爬行获取信息时采取不同的uuid，通过这种转换uuid的方法避免了从技术攻关层面间接攻克美团反爬策略。思想有点像ip代理池的思想。我将我的思想分享给同学，而同学又发现uuid还可以自动生成，解决了uuid来源问题。    经过一段时间的爬取。目前爬取共72家店铺，每家店铺评论人为设置小于150条，所以现在拥有不超过一万条数据。    上图为为收集到的店铺评论。  下图为一个店铺的评论   3.1.3 数据清洗和整理  1. 删除匿名用户，没有使用价值。将不同的店铺评论融合成一个评分矩阵的操作，这里设计到对dataframe的处理，由于dataframe的合并和插入行和列的技术限制较多，不能便捷进行合并生成评分表的工作。需要自己编写代码进行逻辑处理。采用numpy和矩阵格式来进行评分矩阵的生成。建立了一个评分矩阵，但是太稀疏了，与经典的电影评分矩阵无法相比。     稀疏地让人感觉失望，在筛选完大量匿名用户后，获得具有id表示的评价用户五千多名，但是评分矩阵太稀疏了，于是需要在爬取数据时在进行处理。  取消获取评论的数目限制，之前的工作中人为的设置每一家店铺最多的评论数量限制到150.现在取消该限制，因为我想着评论越多的店铺，去吃的人多，同时知名度广，所以一个食客如果在其他店铺就餐，如果对美食有追求，大概率也会去评论较多的餐厅去尝试，这样就能尽可能多地得到参与多次餐厅评价的用户。  同时还发现一个较为有趣的爬虫现象，对于美团网，在不同时间段，网站的反爬策略是不同的，当我使用相同的程序在不同的时间段进行爬取，效果是不同的，如果在工作日的白天上班时间爬取，美团网对于数据获取数量和可疑爬虫行为十分敏感，动不动就将我的爬虫抓住，可能一个uuid获取几百条信息就被发现逮捕。而在周五夜晚比如晚九点到零点左右，美团网对于大量的数据获取和可疑爬虫行为可能放松管理。一条uuid隐藏在大量访问中能一次获取两千到五千多条数据。（下图为一个晚上23：00左右一条uuid爬取过程中的截图。    根据我的猜测和分析：可能原因是在工作日白天时间，普通用户一般是不会去浏览网站上大量的美食信息（摸鱼除外），所以在这段时间大量访问发送数据请求的很有可能就是在工作时间进行数据获取和网络攻击的互联网工作者，当然从严管理，从严执行打击爬虫策略。  而到了夜晚，大量正常的访问者则会向网站发送大量数据请求，如果从严治理，很有可能错杀误杀，从而造成对正常浏览用户的错杀，对于自身产品是不利的。  就好像一个交通检查站要做到尽可能的不被车主抱怨，就需要放松在车流高峰期的检查一样。最后获取得到十五万多条数据。并对数据进行预处理 3.1.3 建立评分矩阵   去除匿名和重复后的user list 含有42568条userid。但在跑评分矩阵时跑了一晚上也没跑完。这时候想到能否在user list 去除重复值的环节进行修改，在去除匿名用户后，只将重复过的user id加入list中，这样能有效缩小矩阵规模。也使矩阵在后面的矩阵分解中更加容易和准确。    打印去除匿名后的list大小为14949 ，也要跑挺长时间的。   3.2 项目流程3.2.1用户分级制度 通过在数据处理阶段的观察，我获得了几万条用户行为数据，但其中存在刷单者（体现在多次对同一店铺进行相同低质量评论），评论次数稀少者等，没想到实际数据相比实验样本数据更为复杂。为了使推荐系统更加完善和个性化，我采取了用户分级制度，采用三级制度分级，VIP用户级别，Common用户级别，其他用户级别。  VIP用户级别：排除匿名用户，排除低质量评论（字符重复多次，长度不够者），评论数超过20条的用户我们进行采集并标记为VIP用户  Common用户级别：排除匿名用户，排除低质量评论（字符重复多次，长度不够者），评论数超过6条的用户我们进行采集并标记为Common user用户  其他用户级别：不在上述两条的用户我们都识别为其他用户。 3.2.2用户分级算法 Common用户筛选，筛选评论次数大于6条者。筛选代码如下。保存对应的用户list    VIP用户初步筛选将上述的value值变为20，将获得的VIP用户id保存在相应文件中  List中。通过对VIP列表中的用户提取评论，进行评论筛选。然后删去不符合条件者，将用户list保存。 3.2.2分级的推荐算法 通过长时间的学习，在该项目中使用的算法有基于item和评论相似度的协同过滤算法，基于美食相似度的计算：主要实现了三种相似度的计算（对输入的两个列向量进行相似度的计算。）欧式距离，余弦相似度，欧几里德距离。    SVD++矩阵分解算法，使用surprise中SVD++算法，重新处理评分矩阵，因为surprise只要按照一定数据格式输入数据，会自动在内部组织一种评分矩阵形式。通过SVD++算法进行矩阵分解效果不是很好，可能因为矩阵过于稀疏。部分核心实现代码如下。    LIBFM算法，该算法的探索来自自己之前的学习经历。该算法有台湾一所大学公开出来的开源框架。可以有效的进行迭代进行评分预测，主要用于VIP用户，实现精准个性化评分预测，从而实现精准推荐。LibFM算法的使用可以参照我探索的csdn博客<https://blog.csdn.net/qq_41785852/article/details/105447806>  信息提取和LIBFM算法结合，对于精品VIP用户，要实现精准预测，我们必须获得用户的行为特征数据，从而才能使用LIBFM算法进行精准预测，而特征数据是没有的，需要我们自己从用户评论中进行信息提取。涉及到NLP相关知识。对于评论，我设置了三个特征进行特征评论提取，从口味，服务，环境卫生三个维度设置特征。  VIP特征提取：获取VIP用户的评论，进行中文分句处理，从中提取相关维度信息，将包含维度信息的分句们利用SnowNlp模块进行情感分析生成得分，将分句们所有包含该维度信息的得分进行平均，获得用户关于该维度的打分。对所有评论执行该操作，我们就能获得关于该VIP用户的一系列维度预期值和最终打分。从而获得VIP的偏好模型。  基于时间上下文的店铺特征得分：将所有的店铺评论按照时间排序，获得距离当前时间最近前200条评论，获得三个维度平均特征得分。将所有店铺得分整合成同一个数据集。  将VIP训练集和店铺测试集通过csvToLibFm代码转化成libfm识别的模式，放入虚拟机中训练并预测得分。获得VIP评分矩阵。  中文分句核心代码。    获得服务维度的特征的代码如下。    余下代码在getVIPuser.py代码中,不过多展示。  对于某一个VIP用户我们经过libFM格式处理后，得到如下结果，我们看到第一列是评分，后面三列是其对不同维度的打分。我们可以通过libfm的训练得到一个用户对不同维度的打分模型，则后续我们可以根据其他店铺的不同特征得分，来预测VIP用户对店铺的打分。  用户载入虚拟机后的libfm格式的训练集    载入店铺数据测试集    利用libfm进行训练。迭代一百次。    迭代过程。结果放入output.libfm文件。再将其导出到特定的VIP评分矩阵表中。    大致流程如下，省略处理过程。   3.2.3流程图 流程逻辑在untitled的views.py文件中实现。   3.3 项目成果展示   前十条推荐。 4项目评价和改进4.1优点 更加贴近实战，因为数据来源自己的爬取。  通过开创用户分级算法分配来面临复杂的实际数据，对于VIP用户进行精准推荐，对于普通用户采取svd矩阵分解算法与itemcf算法融合的普通推荐。  通过评论提取语义信息，并生成维度信息，重复利用语义特征来进行精准推荐。  精准推荐融合了时间上下文信息。  算法可扩展性强，统一数算法输入和输出格式，方便快速转变算法。  对于冷启动问题，我们采用领头羊策略，根据羊群效应和从众效应，我们对于其他用户的推荐算法取绝于VIP中平均得分最高的店铺，VIP用户都是一群积极评价美食的用户，容易想象他们是一群对美食有着较高要求的吃货。我们通过综合所有VIP用户的评分得到美食店铺总分进行排序，一方面排除了VIP用户的个人偏好，得到一个比较综合，符合大众口味的评分推荐，推荐给新用户。虽然不一定完全符合新用户的个性化口味，但也一定不会难吃。 4.2缺点 对于普通用户的推荐算法效果不是很好，毕竟矩阵比较稀疏。  因为无法获得用户的人口统计学信息比如性别，年龄，工作，收入，地点等信息，缺少人口统计学推荐的融合。 4.3改进 在信息提取的特征提取上，算法还可以优化，从而得到更加精准的评判得分。我采用的是正则表达式般的粗鲁提取，完全可以通过nlp相关知识进行完善。  在进行该部分实验时，也看到了一些基于语义的深度学习推荐算法，但是由于机子跑不起来，遂放弃，十分可惜。可以向这些方向改进。  在查阅资料时思考到我所持有得评分矩阵太过稀疏，在查询相关论文时发现基于文本评论内容和评分相融合的用户偏好预测算法。HFT模型和DTMF模型。  HFT模型，融合了评分与文本评论的推荐模型，将用户评论作为主题模型输入，得到的主题和矩阵分解中的潜在因子之一进行融合，比如HFT模型将所有商铺的所有评论文本组合成一篇文档，所有商品的评论集和作为主题发现模型的文档集，从而得到反映商品特性的主题分布。HFT的介绍论文题目如下。  Hidden Factors and Hidden Topics: Understanding Rating Dimensions with Review Text  主要的思路为    后面打算尝试用HFT模型看一下能否将评论文本中提取得隐含主题因子和评分结合起来预测用户偏好，来提高推荐效率。  ANR算法，"**ANR: Aspect-based Neural Recommender**" in CIKM 2018。文本本评论很容易在许多电子商务和评论网站（例如Amazon和Yelp）上获得，是推荐系统的宝贵信息来源。但是，并非评论的所有部分都同样重要，并且基于其上下文，相同的词语选择可能反映出不同的含义。在本文中，我们提出了一种新颖的端到端基于方面的神经推荐器（ANR），以通过基于注意力的组件为用户和项目执行基于方面的表示学习。此外，我们通过适应神经共同注意机制，通过估计方面级别的用户和项目重要性，对用户如何对项目进行评分背后的多方面过程进行建模。我们提出的模型同时解决了现有推荐系统的一些缺点。  还有DeepCoNN算法等。 5分析和体会5.1体会 这个项目真的艰辛，获取美团数据就用了两周，得到实际数据，实际数据有惨不忍睹，不像网上一些电影推荐采用的样本数据那么漂亮，需要自己进行进行数据处理，同时采用矩阵分解算法效果比较差，只能按条件分类。但也真实因为实际数据的惨不忍睹，在处理数据时对数据有很多思考，所以产生了创意性的用户分级推荐的想法，从而充分利用用户评论来分析用户的偏好 | | | |
| 代码 Untilte库  Views.py---------文件包含用户流程的处理，在这里对用户进行分级推荐算法  Vip\_test.csv--------包含从虚拟机的libfm模型中训练出来的VIP用户评分。如下图所示，第一行是ViP用户id，行索引是不同的店铺id，第i行第j列代表VIP用户j对店铺i的评分。    Foodcomments-------存取获取的数据，每一个文件代表一个店铺下的评论，可见还是蛮大的。      All\_information.csv------- 保存店铺信息    Crawlview.py----------美团数据获取文件  cleanfoodDt.py---------处理数据python文件  getVIPuser.py---------筛选VIP用户的文件  getShopID.py-------------为surprise中SVD算法准备文件的数据  getShopview.py----------为surprise中SVD算法准备文件的数据  foodbySurprise.py--------实验SVD和距离算法的文件，后续在views.py中实现。  PredictScoreCSV.py---------将VIP相关csv文件处理成libFM识别的格式文件  虚拟机中使用文件和代码就不一一列举了。 | | | |