餐饮智能推荐服务项目分析报告

**任务一：数据获取与探索分析**

1、思路

首先，读取meal\_order\_detail.csv文件，并提取列名为disher\_name的数据。然后，处理该列菜品名称字符数据。接着，构建热销度评分指标，热销度指在一定时间内产品销售的程度，公式如下

其中，指菜品被下单的次数，指菜品被下单最少的次数，指菜单被下单最多的次数。最后，绘制条形图展示热销Top10。

2、操作步骤

（1）使用pandas库读取meal\_order\_detail.csv文件。

（2） 通过列名索引，提取dishes\_name的数据，可以发现部分菜品名称后有字符数据，如“番茄炖牛腩\r\n”，这里使用df.str.strip()去除字符数据，完成菜品名称的清洗。

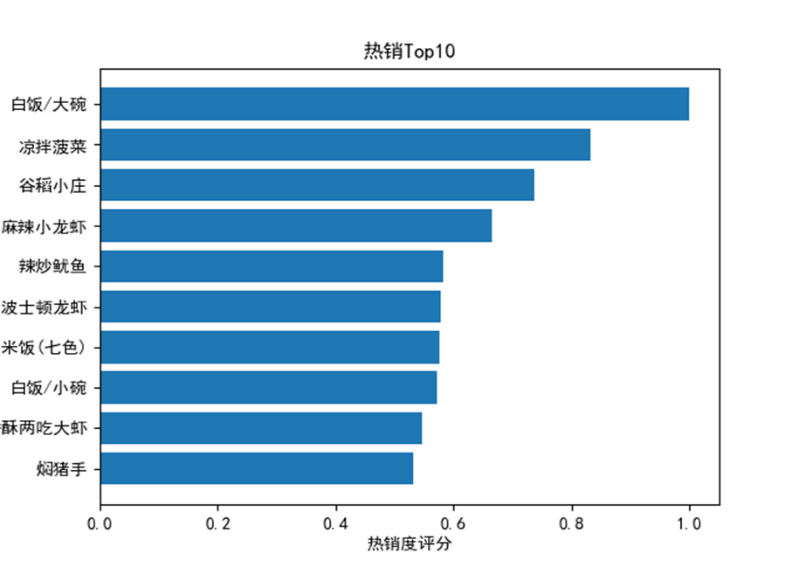
（3）计算各菜品下单的频数。创建空字典dishes\_count，使用for循环和if语句，得到键为菜品名称，值为下单频数的字典数据。

（4） 计算热销度评分指标。首先将步骤（3）得到的字典dishes\_count转换成DataFrame类型的数据。然后，建立一个行索引为菜品名称，列为“热销度评分”的DataFrame数据，使用for循环，逐一计算热销度评分。

（5）使用matplotlib库绘制条形图，展示排在前十名的菜品。

3、结果及分析

完成步骤(1)-(5)，得到一个条形图，如下所示：



**任务二：数据预处理**

1、思路

首先，读取meal\_order\_info.csv文件，提取列名为order\_status的数据，统计每个订单状态的占比，其中，“0”表示未结算，“1”表示结算，“2”表示已锁单。然后，选取有效的订单数据，这里可以认为，order\_status字段中的0/2为无效数据，dishes\_name字段中的“白饭/小碗”和“白饭/大碗”为无效数据。最后，选取主要特征，保存dishes\_name字段和emp\_id字段即可。

2、操作步骤

操作步骤主要分以下两步：

第一步，统计每个订单状态的步骤，绘制饼图。

（1）使用pandas库读取meal\_order\_info.csv文件，并提取列名为order\_status的数据。

（2）建立一个键为订单状态、值为数量的空字典order\_per，通过for循环，统计各订单状态的数量。

（3）使用matplotlib库的pie方法，绘制每个订单状态占比的饼图。

第二步，清洗无效数据，并将数据存储于csv文件。

（1）从meal\_order\_info.csv文件中提取列名为订单id（info\_id）的数据，建立一个空列表no\_id，使用for循环，将无效数据的订单id（info\_id）存入列表no\_id中。

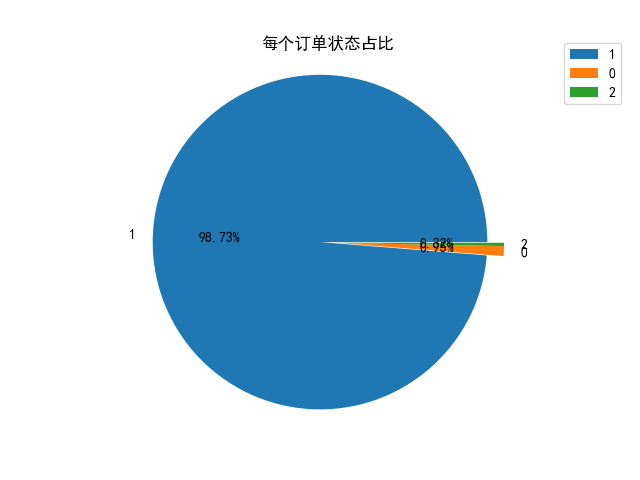
（2）使用pandas库读取meal\_order\_details.csv文件。

（3）通过isin()函数过滤订单状态为0和2或者菜品名称为“白饭/小碗”和“白饭/大碗”的数据。

（4）清洗菜品名称字符数据，通过to\_csv()将数据存入data.csv文件。

3、结果及分析

完成第一步，得到饼图，如下所示：



其中，已结算的订单（即order\_status字段值为“1”）占比为98.73%，未结算的订单（即order\_status字段为“0”）占比为0.95%，已锁单的订单（即order\_status字段为“2”）占比为0.32%。

完成第二步，清洗后的数据存入data.csv文件，一共有三个字段，即订单id（order\_id）、菜品名称（new\_dishes\_name）和客户id（emp\_id）。部分数据如下图苏轼：



data.csv文件部分数据

**任务三：模型结果准备**

1、思路

首先，需要将data.csv文件里的数据划分为训练集和测试集，比例为7：3。从客户id（emp\_id）入手，为保证有足够的信息量，只保留点了三个菜以上的客户。然后，构建训练集数据客户-菜品二元矩阵，索引为客户id（emp\_id），字段名为菜品名称（new\_dishes\_name）。

2、操作步骤

第一步，需要先删除只点了三个菜以下的客户id（emp\_id），再对数据进行训练集和测试集的划分，并且分别存入train\_data.csv文件和test\_data.csv文件。

（1）使用pandas库读取data.csv文件，提取满足只点了三个菜以下的索引，并且通过isin()将相应的数据删除。

（2）使用sklearn.model\_selection库的train\_test\_split()函数划分训练集和测试集，为避免在划分数据集的过程中把同一用户的订单分开，先把客户id提取出来放在一个列表lit2中，然后进行划分。为方便后续操作，保证每次划分的训练集和测试集不变，设置train\_test\_split()函数的参数random\_state=2。

（3）通过to\_csv()函数将训练集和测试集的数据分别存入train\_data.csv文件和test\_data.csv文件。

第二步，构建训练集数据”客户-菜品”二元矩阵。

（1）使用pandas库读取data.csv、train\_data.csv文件。

（2）使用unique()函数，将唯一菜品名称和训练集客户id分别放入dishes\_name列表、train\_emp\_id列表中。

（3）创建一个索引为train\_emp\_id、列为dishes\_name的DataFrame，使用for循环，将客户下过单的菜品赋值为1，并将空值填充为0。通过to\_csv()函数将数据存入客户-菜品二元矩阵.csv文件。

3、结果及分析

完成第一步，可知客户id（emp\_id）为1343和1594的客户只点了两个菜，所以将这两位客户的订单信息删除，剩下总共有467个客户，9432条有用数据，并且将数据划分成训练集和测试集。其中，训练集有326个客户，测试集有141个客户。部分数据如下图所示：



train\_emp\_id.csv部分数据



test\_emp\_id.csv部分数据

完成第二步，得到的部分数据如下图所示：



客户-菜品二元矩阵.csv部分数据

**任务四：模型构建**

1、思路

首先，读取客户-菜品二元矩阵.csv文件，计算菜品之间的相似度矩阵。然后，计算推荐算法中目标客户对所有菜品的感兴趣程度，并根据感兴趣程度给目标客户生成推荐列表。感兴趣程度公式如下：

其中，SIM为菜品相似度，R为客户对物品的兴趣，这里只存在0/1，即未用餐/已用餐。

2、操作步骤

第一步，需要计算菜品之间的相似度矩阵。

（1）使用pandas库读取客户-菜品二元矩阵.csv文件。

（2）创建一个索引和列都为菜品名的dataframe，用来存储菜品相似系数。

（3）通过for循环和scipy.spatial.distance库的pdist()函数，计算菜品之间的Jaccard相似度，再通过两个for循环填充下三角数据，并写入菜品相似度.csv文件。

第二步，计算客户的兴趣度，生成每个客户的推荐列表。

（1）使用pandas库读取data.csv、test\_data.csv和菜品相似度.csv文件。

（2）创建一个键为测试集的客户id，值为已下单菜品列表的字典，通过set()函数删除字典值中的重复项。

（3）生成推荐列表。读取列名列表（col），测试集客户id列表（test\_id），创建一个空字典dic，是用来存储测试集客户id-菜品-感兴趣程度的嵌套字典。

（4）使用第一个for循环，对每个客户进行循环。提取客户u已下单的菜品名称列表（lit1），获取已下单菜品在col中的索引位置，放入列表idx中。创建列表row为全部菜品的索引。

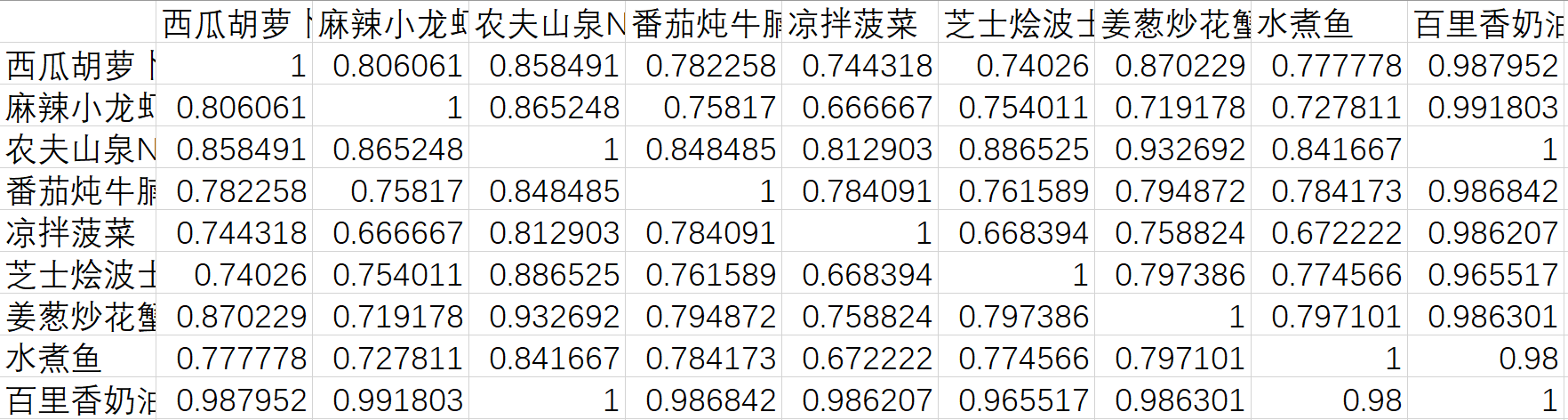
（5）使用第二个for循环，对每个客户的全部菜品进行循环，计算客户u对菜品j的感兴趣程度，每个客户u的推荐表放入字典dic2中。

3、中间过程

关于相似度计算方法的选择：当数据集为二元变量时，只有两种状态，即0或者1。此时，欧氏距离、皮尔逊相关系数等计算相似度的方法无法派上用场，于是引出Jaccard系数，这是一个能够表示两个数据集都是二元变量（也可以多元）的相似度的指标，Jaccard系数越大则相似度越高。公式如下所示：

4、结果及分析

完成第一步，得到菜品之间的相似系数。如果当集合A、B都为空时，J(A,B)定义为1，例如，J(西瓜胡萝卜沙拉，西瓜胡萝卜沙拉)=1、J(农夫山泉NFC果汁100%橙汁，百里香奶油烤红酒牛肉)=1。部分数据如下图所示：



菜品相似度.csv部分数据

完成第二步，得到一个嵌套字典dic，其中字典的键为客户id，值为以菜品名称为键、感兴趣程度为值的字典。部分数据如下图所示：

****

字典dic的部分数据

**任务五：模型评价**

1、思路

根据任务4\_2所得到的测试集客户-菜品推荐列表中感兴趣程度，对菜品进行从大到小的排序，选择推荐与客户下单数量匹配的前K个菜品。因为客户只有二元选择，即已下单与未下单，所以要用到分类准确度，这里构建评价指标准确率P，公式如下：

其中，TP为正确推荐菜品数，FP为错误推荐菜品数。最后，分析推荐效果。

2、操作步骤

（1）调用Task4\_2.py的变量test\_id、变量dic1和变量dic，这里Task4\_2.py的代码对应的是code文件夹里的任务四代码(2).txt。

（2）创建列表tp\_l和列表fp\_l分别存储测试集每个客户的推荐正确菜品数量和推荐错误菜品数量。

（3）使用第一个for循环，对每个客户进行循环。首先，提取客户真实下单的菜品放入列表dishes\_T；然后，计算需要向客户推荐的菜品数量，这里推荐原则为客户下单k个菜，就推荐k个菜。

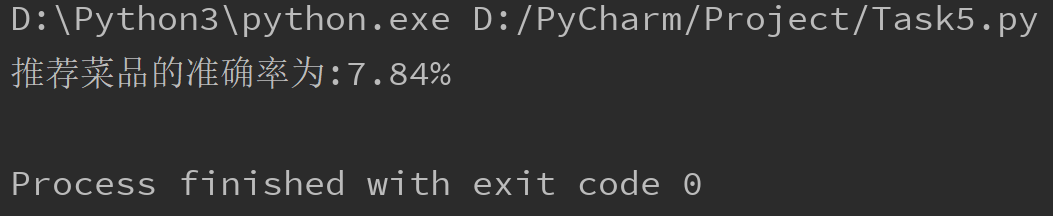
（4）使用第二个for循环，根据感兴趣程度向客户推荐k个菜品。

（5）使用第三个for循环，计算每个客户的tp值和fp值。

（6）使用两个循环，分别算出TP和FP，这里TP指所有客户推荐正确的菜品数量，FP指所有客户推荐错误的菜品数量，最后计算准确率p。

3、结果及分析

完成任务五的步骤，得到准确率如下图所示：



由于本项目采用的是最基本的协同过滤算法进行建模，因此得到的模型结果也是一个初步的效果，实际应用过程中还需结合业务进行分析，对模型进一步改造。