

餐饮智能推荐服务项目分析报告

洪志鹏



目录

[一、项目背景 3](#_Toc34253418)

[二、项目目标 3](#_Toc34253419)

[三、项目实现步骤 3](#_Toc34253420)

[3.1.1 读取订单: 3](#_Toc34253421)

[3.1.2 处理菜品名称字符数据: 4](#_Toc34253422)

[3.1.3 构建热销度评分指标: 4](#_Toc34253423)

[3.1.4 绘制条形图展示热销Top10: 4](#_Toc34253424)

[3.2.1 统计每个订单状态占比: 5](#_Toc34253425)

[3.2.2 选取有效的订单数据: 5](#_Toc34253426)

[3.2.3 选取主要特征: 6](#_Toc34253427)

[3.3.1 将数据集划分为训练集和测试集: 6](#_Toc34253428)

[3.3.2 构建客户-菜品二元矩阵: 6](#_Toc34253429)

[3.4.1 自定义函数求菜品相似度: 7](#_Toc34253430)

[3.4.2 自定义函数进行推荐，并生产推荐列表: 7](#_Toc34253431)

[3.5.1 构建测试用户IP字典: 8](#_Toc34253432)

[3.5.2 构建评价指标 8](#_Toc34253433)

[四、模型评估: 8](#_Toc34253434)

# 一、项目背景

餐饮行业作为我国的第三产业中的一个传统服务型行业，一直有这突飞猛进的增长速度。然而于此同时，餐饮业发展的质量和内涵也有这重大的变化，根据国家统计局的数据。餐饮行业餐费收入处于增长趋势，但是同比增长率却有着很大的变动。

如今餐饮企业面临着许许多多的问题以及压力，例如同行竞争压力大，餐厅对消费者的需求认识不到位，客户的忠诚度低，菜品和服务没有深入客户的需求、缺乏个性化。为了解决这样的问题提升餐厅的问题，我们用推荐菜品这样的方式去满足客户的需求，同时给商家带来销售量。

# 二、项目目标

对数据进行分析，对餐饮企业的订单数据进行分析，并利用协同过滤算法对就餐的客户进行菜品的智能推荐服务。需要完成，从数据的清理分析，到模型构建，再实现对餐饮行业的分析然后生成顾客的推荐列表，最后对推荐的结果进行分析记录。

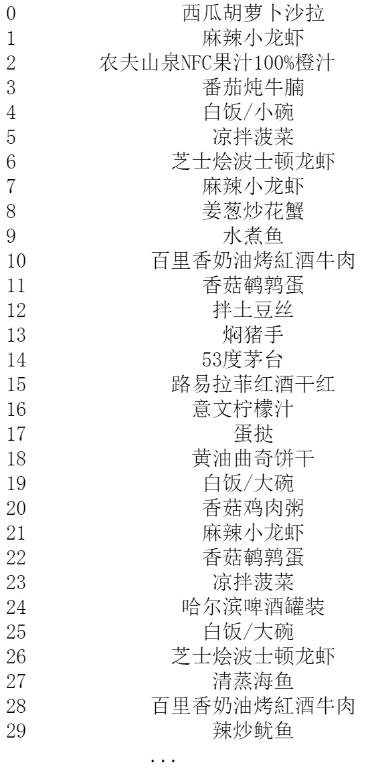
# 三、项目实现步骤

## 3.1.1 读取订单:

读取pandas包，并用read\_csv进行读取，对于meal\_order\_info.csv需要用encoding=“GBK”进行读取。两个csv分别存储在info 和detail 变量名下。

## 3.1.2 处理菜品名称字符数据:

思路：用replace函数替换

需要处理掉菜品名里面的“/n”和“/r”。先创建一个空列表newdishname，用循环对dishes\_name里的值一一取出，再用replace对取出的值替换掉特殊字符，再添加到newdishname中，最后将newdishname传入到dishes\_name。

## 3.1.3 构建热销度评分指标:

构建函数hot\_ranting(data),返回return (data-data.min())/(data.max()-data.min())。

## 3.1.4 绘制条形图展示热销Top10:

思路：菜品的订单有数量的不同，要用groupby函数，sum函数。此外还需要排序，需要sort\_value函数，最后需要plot函数画图。

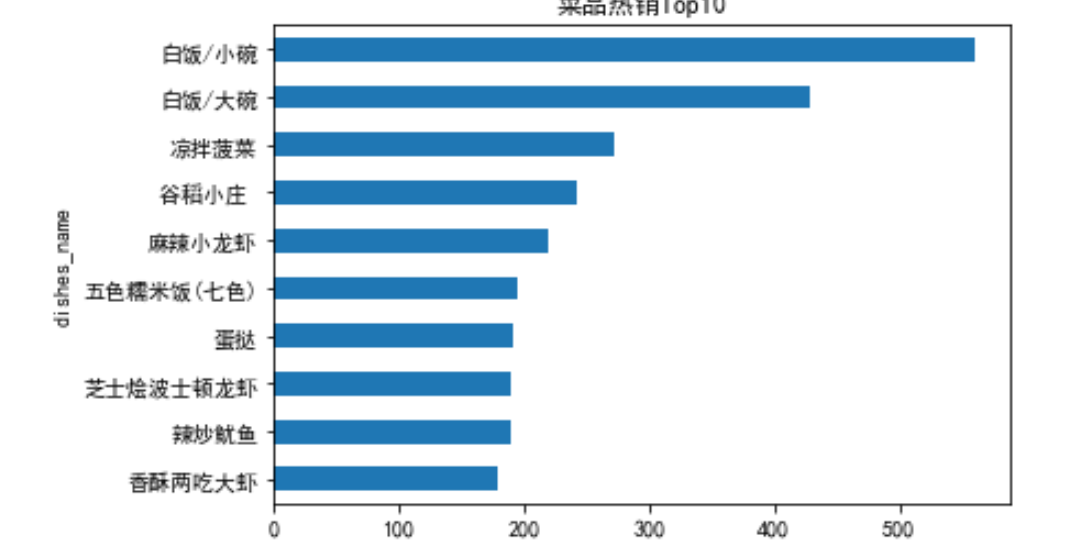
先获取detail里面的'dishes\_name','counts','amounts' 传入data中。用groupby根据dishes\_name 求和并根据数量降序排序，取最后10个进行画图。

data.groupby('dishes\_name').sum().sort\_values(by='counts').tail(10)['counts'].plot.barh(title="菜品热销Top10")。

画图中文显示不正常 引入pylab的mpl 引入中文。

from pylab import mpl

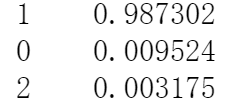
mpl.rcParams['font.sans-serif']=['SimHei']



## 3.2.1 统计每个订单状态占比:

思路，需要对某一列进行统计，用value\_counts函数

用value\_counts对order\_status统计个数，并在value\_counts函数中设置normalize=True 让其显示为百分比。



## 3.2.2 选取有效的订单数据:

思路：用drop函数进行删除，并且可以用set找到两个不一样的订单

选取order\_status=1 的全部订单 info=info[info.order\_status==1]。

删除掉白饭/小碗和大碗，用loc定位再取出index 并用drop函数删去这些数据，并返回参数给detail。

detail=detail.drop(index=(detail.loc[(detail['dishes\_name']=='白饭/小碗')].index))

detail=detail.drop(index=(detail.loc[(detail['dishes\_name']=='白饭/大碗')].index))

去除在detail中有但是在info里面没有的订单，使用差集找到不同的订单，再用drop去除。

## 3.2.3 选取主要特征:

对data2传入dishes\_name 和emp\_id的数据。

data2=detail[['dishes\_name','emp\_id']]

## 3.3.1 将数据集划分为训练集和测试集:

思路：不能让train\_test\_split随机划分，所以需要对顾客名单进行分割

由于不能随机分，所以导出点了三个菜以上的顾客名单，并去除重复数据用drop\_duplicate函数。

customer\_list=dish\_order.emp\_id.value\_counts()[dish\_order.emp\_id.value\_counts()>3].index.tolist()

再用train\_test\_split对顾客名单进行划分，训练集占比30%，test\_size=0.3 保存为train\_data 和 test\_data

## 3.3.2 构建客户-菜品二元矩阵:

思路：构建一个dataframe框架，再往里面填充数据，最后再用fillna填充

定义函数two\_dimension(customer\_list,dishes\_list,df),对传入的 customer\_list 和dishes\_list构建dataframe，在根据客户的列表，在df中找出这个顾客吃过的菜，并写入这个顾客对应的菜，值为1，最后用fillna 用0来填充没有数据的地方。最后返回dataframe。

引用函数，构成训练集和测试集的二元矩阵。

train\_customer\_dish=two\_dimension(train\_data,dishes\_name,dish\_order)

test\_customer\_dish=two\_dimension(test\_data,dishes\_name,dish\_order)

## 3.4.1 自定义函数求菜品相似度:

思路：用numpy进行数组的计算，同时为了模型的通用，建立一个class，保存所有的相似性的dataframe

构建类 ItemCF 并在里面构建相似度和推荐函数。

在类下面构建函数sim\_two 使用欧几里得距离求两个菜品的相似度。

1/(1+np.sqrt(((np.array(dish1)-np.array(dish2))\*\*2).sum()))

在类下面，再构建sim\_all函数，创建一个columns和index为菜品名的数据框命名为sim\_matrix,储存两两个菜品的相似度，使用循环和sim\_two函数 填充sim\_matrix并返回。

## 3.4.2 自定义函数进行推荐，并生产推荐列表:

思路：

在得到所有的相似性之后，对测试集的顾客开始推荐，得到每个顾客各个菜品最高的分数，并排序，保存在一个字典下。同时根据推荐的个数再返回相应推荐菜品的个数。

步骤：

在类下，定义函数recommodation\_dic 对传入的客户名字，生成键队对列表。用客户吃过的全部菜品，和其他菜品对某一菜品的相似度相乘计算P值。并对其进行排序在返回

再在类下构建全顾客的推荐列表。

定义函数all\_customer\_recommodation\_list，生成全部推荐列表传入test\_all\_dic和train\_all\_dic。

定义函数select\_recommodation\_dic， 传入推荐个数n 返回测试集推荐列表。

实例化模型

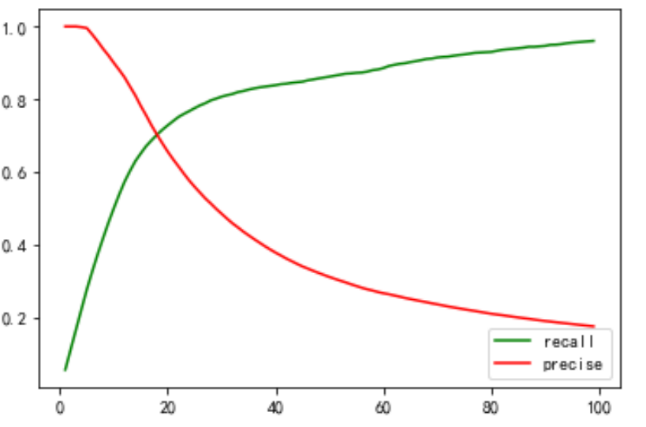
## 3.5.1 构建测试用户IP字典:

创建字典，循环客户IP，找到客户吃过的菜品，生成IP字典 保存在customerIP里面

## 3.5.2 构建评价指标

构建指标准确率函数 precise(test\_predict\_dic,dic) 计算出预测准确的值，总体的值和顾客吃过的菜，分别计算出准确率和召回率。

# 四、模型评估:

当对客户推荐1-3个菜品的时候，模型的准确率很高，但是只能少部分的覆盖顾客喜欢的菜，当推荐菜品在四个即四个以上的时候，菜品的推荐的准确率会略微下降，并且准确率下降的速度会逐渐减慢。但是相对于召回率来说，在推荐菜品在10个一下的菜，菜品的推荐的增加能够包含更多客户感兴趣的菜。但是超过一定范围之后，菜品推荐的召回率增长率会下降。因此可以通过此模型的运用对顾客进行推荐。同时在定义类的时候需要大量的计算，在模型实例化的时候会出现一段时间的等待，但在模型计算完成后，只需要取出并保存即可。

测试集和训练集的推荐列表分别保存在两个Excel下，每一列从上到下代表一个顾客推荐菜品的顺序，越是在上面推荐度就越高。可以根据需求使用select\_recommodation\_dic函数，传入每个顾客推荐的菜品个数N，返回一个字典。