

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。

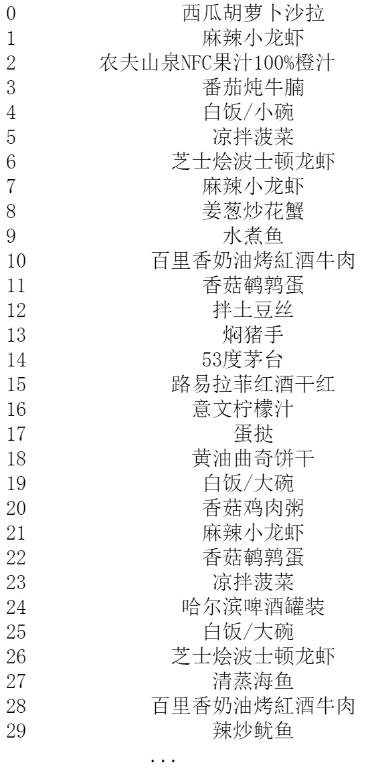
# 三、项目实现步骤

## 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。

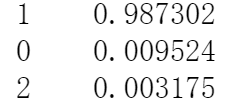
from pylab import mpl

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isin函数和取反去除订单详情表中有，订单表中没有的订单

## 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，找到客户吃过的所有菜品并根据相似度表找出相似度高的前几个，保存在一个字典下。

步骤：

在类下，定义函数recommodation\_dic对测试集顾客的ID找到他们吃过的菜并进行推荐使用排序函数对顾客点过的菜进行排序，并且根据传入的N值，找到每个菜相似度靠前的N个

最后，实例化模型

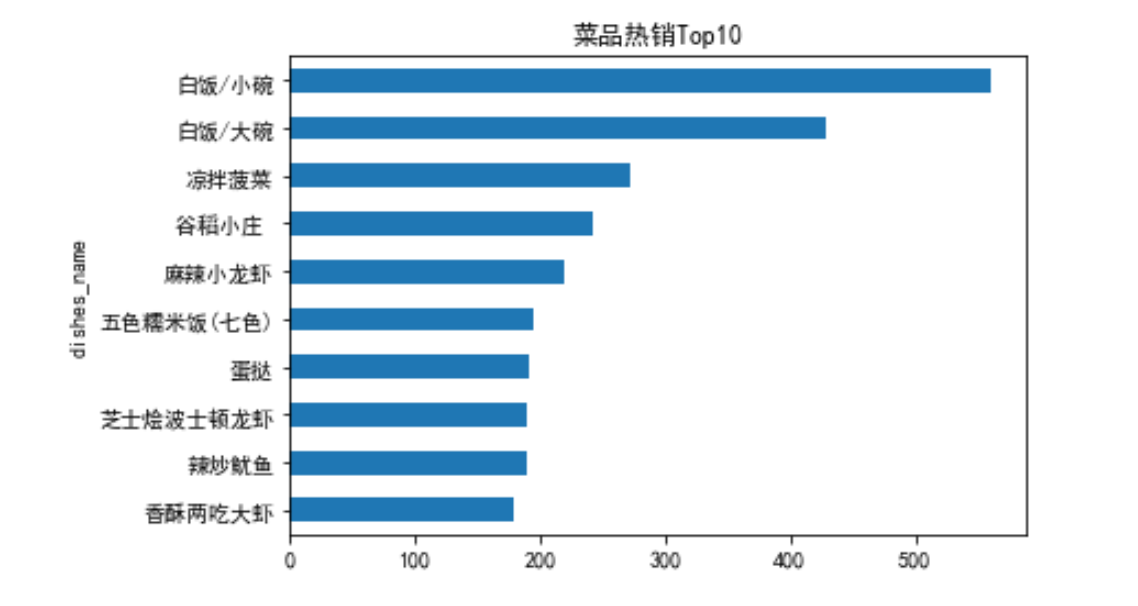
## 3.5.1 构建测试用户IP字典:

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。

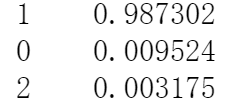
# 四、小结:



在所有的订单中，白饭是顾客最热销的菜品，因为大多数的来餐厅的顾客都会点米饭，除开米饭来看，凉拌菠菜是菜品当中最畅销的菜，谷稻小庄，然后是麻辣小龙虾。五色糯米饭、蛋挞、芝士烩波士顿龙虾、辣炒鱿鱼的销量相差不多、最后是两吃大虾。这就是排名靠前的前十个热销菜品。

在订单中，显示了三种订单状态，1为结算，2为锁单，0为未结算。根据统计，如图所示，我们可以发现大部分的订单都已经结算。有少部分的订单状态显示为锁单或者未结算。在使用模型的时候，我们需要删去一些订单。

同时在订单详情表中也需要去删除这些不符合的订单。



然后再进行划分测试集和训练集，最后对模型的进行评估。如图所示，当我们对顾客的菜品进行分析后，按照顾客每吃一个菜就推荐一个菜进行分析。当每次推荐一个的时候，模型的准确率达到了20%左右的准确率。

