大数据实践A实验报告

汇报小组：10组 汇报时间：2024年7月

小组成员：龚仔航、林思涛、钟问重、刘哲、宋奇轩、胡聿彬、刘宇帆

目 录

COMPANY

01 小组分工

02 项目一

03 项目二

04 总结

小组分工

Work assignment

成员：	工作
龚仔航	PPT制作以及汇报
林思涛	项目一建模以及评估
钟问重	项目二建模、分析以及可视化
刘哲	项目二数据处理
宋奇轩	项目一（User Based Model）
胡聿彬	项目一数据处理
刘宇帆	项目一(Spark ASL)

TWO

项目一：某团大数据智能推荐系统

项目背景

1.外卖已经成为当今最常见的用餐方式之一 。在外卖平台上，平台会引导用户对于品尝过的菜品进行评价打分，并通过数据，针对老用户进行个性化的菜品推荐，包括用户的偏爱菜品和新菜品。

数据探查方式

2.原始数据使用JSON格式存储，故使用Spark SQL进行加载，生成dataframe后，进行对评分项分组、统计有无重复数据等数据探索。

数据处理策略

3.首先，处理重复评分数据，将最新评分认定为用户的最终评分，将其余记录删除；其次，以数据去重再排序后的下标值代替原始数据的值的方式对数据进行编码，进行数据标准化；最后，将数据集按8：1：1的比例划分为训练集、验证集和测试集。

算法选择——三种协同过滤算法

基于用户的协同过滤算法

即计算用户之间的相似度，随后选取与目标用户最相近的K用户，最后通过这K个用户进行推荐 。

基于物品的协同过滤算法

用户对物品的预测评分可以由该用户对与该物品相似度最高的K个邻居物品的评分通过加权平均计算得到，所以计算物品之间的相似度来计算得到推荐的物品。

基于Spark ALS的协同过滤算法

Spark的MLlib中已经包含了ALS算法包，开发者可以直接调用它，设置相关技术参数来建模

基于用户的协同过滤算法

进一步数据处理

通过设置***minItemsRatedPerUser=2***过滤掉评价数目小于2的用户，提高运行稳定性和推荐速度。同时进一步计算每一个**用户**的平均评分，用于后续的相似度计算。

计算用户的相似度

处理完成之后，将形式***(user,(item,userItemRating,userMeanRating))***，的数据转化为***((userA, userB), (ratingA, meanRatingA, ratingB, meanRatingB))*** 同时过滤掉自身匹配和重复的情况，随后通过**Jaccard公式**计算用户之间的相似度即 ***val similarity = min(ratingA, ratingB) / (meanRatingA + meanRatingB)*** 计算用户之间的相似度，并将格式转化为 ***((userA, userB), similarity)*** 形成用户的相似度矩阵。

计算所有用户的推荐物品并保存

利用上述的用户相似度获得最后的推荐结果并按照相似度排序，最后形式为 ***(user, itemSimList)*** 同时最后取前 ***recommendItemNum*** 保存到指定位置。

基于物品的协同过滤算法

进一步数据处理

通过设置***minItemsRatedPerUser=2***过滤掉评价数目小于2的用户，提高运行稳定性和推荐速度。同时进一步计算每一个**物品**的平均评分，用于后续的相似度计算。

计算物品的相似度

处理完成之后，将与基于用户的协同过滤算法类似，随后通过 **Jaccard公式**计算物品之间的相似度，并将格式转化 ***((item1, item2), similarity)*** 形成物品的相似度矩阵。

计算所有用户的推荐物品并保存

利用上述的物品相似度获得生成推荐模型 (item,List(item))，随后将该模型与训练数据相结合，按相似度排序，生成推荐结果集(user,List(item))，并限制推荐数目为 ***recommendItemNum***

基于Spark ALS算法建模

基于验证集寻找最佳的模型参数

首先基于验证集以RMSE为评估标准寻找到最优的参数，主要是ALS的rank,lambda,iteration参数，通过网格遍历的方法寻找到最优参数。
通过训练，最优参数为rank=10,iteration=10,lambda=0.4

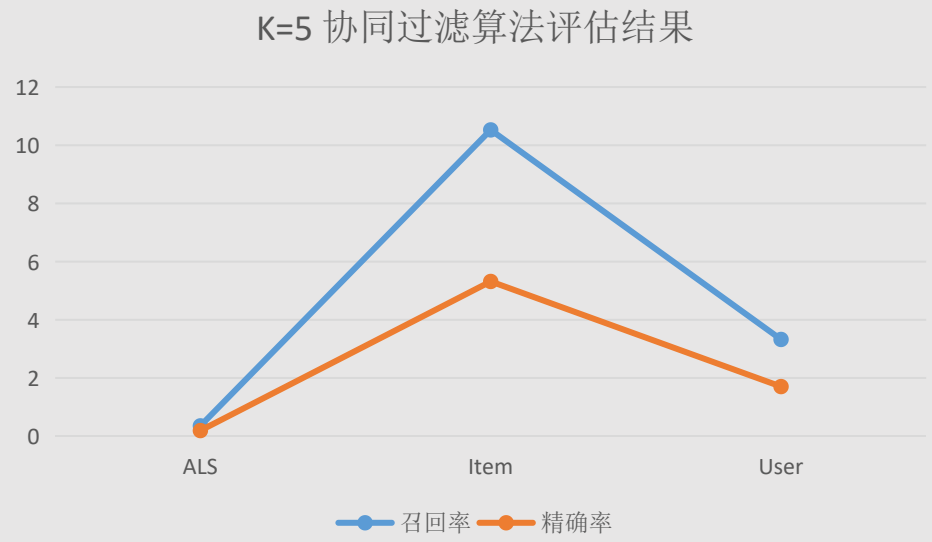
训练模型

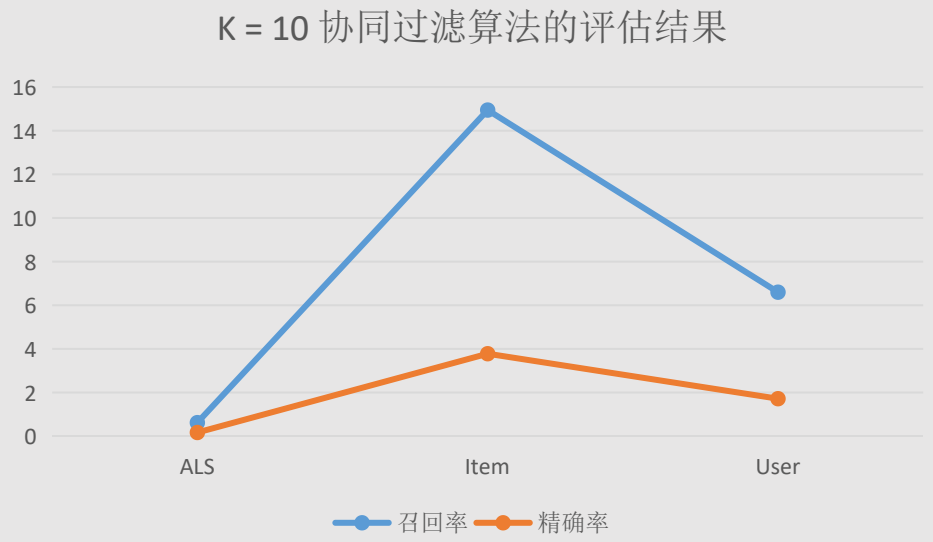
寻找到最优参数之后，基于最优参数，调用ALS的train方法进行训练的得到最优的模型，同时保存到特定目录下。

协同过滤算法的评估

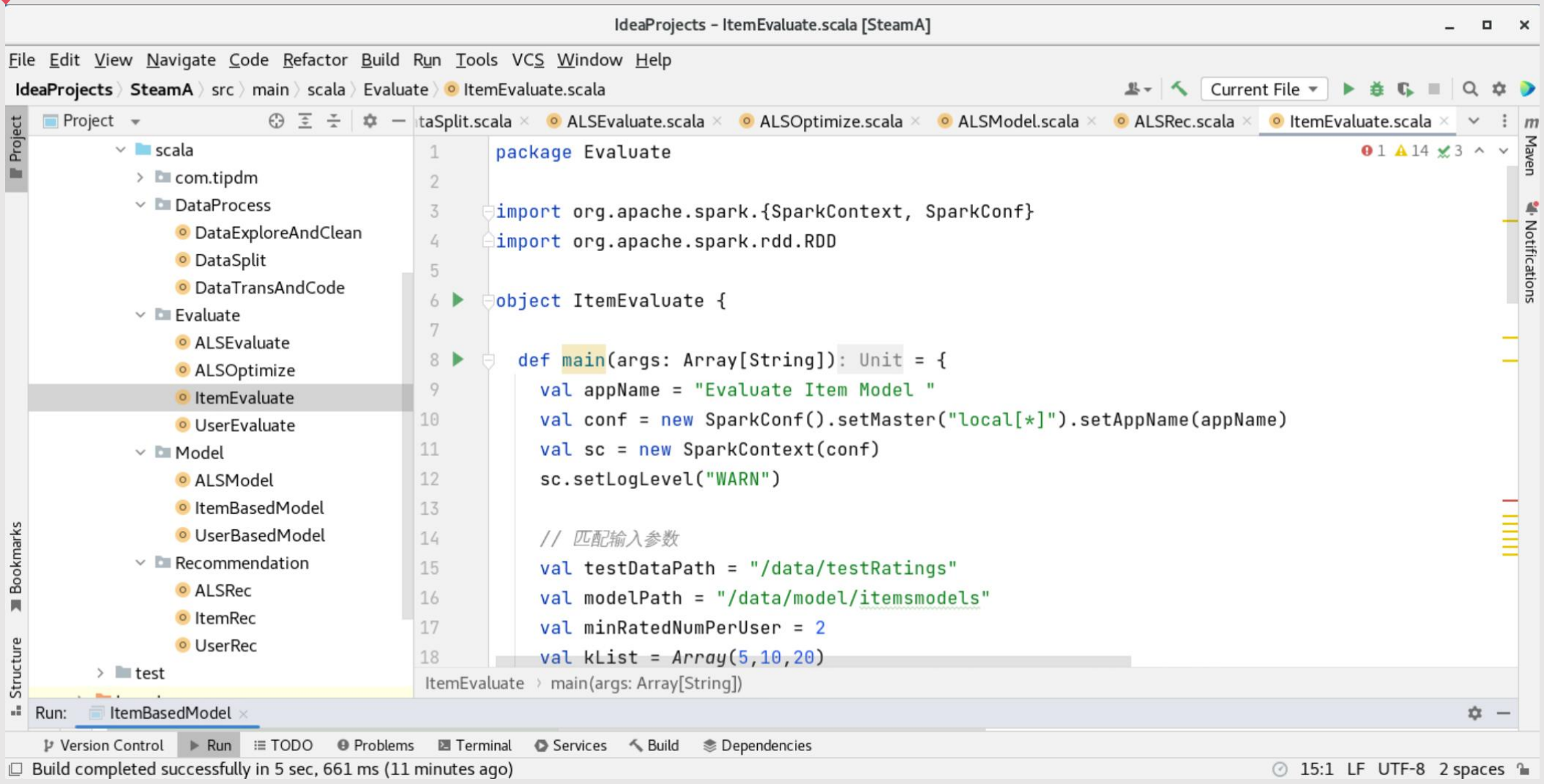
整体流程

导入模型之后，先在测试集内寻找与训练集有相同UserID的数据，并于前K个推荐物品进行连接形成 ***(user, (test_items, recommended_items))*** 随后基于这一数据进行召回率以及精确率的计算，最后返回K次的平均值作为模型的评估结果。

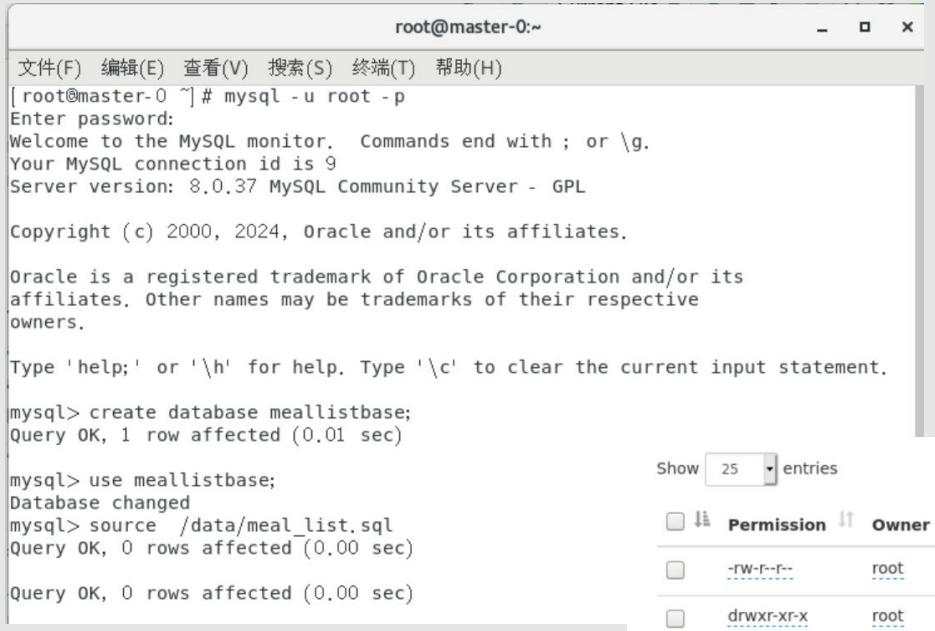




运行过程中的部分截图


IdeaProjects – ItemEvaluate.scala [SteamA]
File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
IdeaProjects 〉 SteamA 〉 src 〉 main 〉 scala 〉 Evaluate 〉 ItemEvaluate.scala
Current File
Project
scala
com.tipdm
DataProcess
DataExploreAndClean
DataSplit
DataTransAndCode
Evaluate
ALSEvaluate
ALSOptimize
ItemEvaluate
UserEvaluate
Model
ALSModel
ItemBasedModel
UserBasedModel
Recommendation
ALSRec
ItemRec
UserRec
test
taSplit.scala
ALSEvaluate.scala
ALSOptimize.scala
ALSModel.scala
ALSRec.scala
ItemEvaluate.scala
1 14 3
1 package Evaluate
2
3 import org.apache.spark.{SparkContext, SparkConf}
4 import org.apache.spark.rdd.RDD
5
6 object ItemEvaluate {
7
8 def main(args: Array[String]): Unit = {
9 val appName = "Evaluate Item Model "
10 val conf = new SparkConf().setMaster("local[*]").setAppName(appName)
11 val sc = new SparkContext(conf)
12 sc.setLogLevel("WARN")
13
14 // 匹配输入参数
15 val testDataPath = "/data/testRatings"
16 val modelPath = "/data/model/itemsmodels"
17 val minRatedNumPerUser = 2
18 val kList = Array(5,10,20)
ItemEvaluate 〉 main(args: Array[String])
Run: ItemBasedModel
Version Control Run TODO Problems Terminal Services Build Dependencies
Build completed successfully in 5 sec, 661 ms (11 minutes ago)
15:1 LF UTF-8 2 spaces
Maven
Notifications
Bookmarks
Structure

运行过程中的部分截图



UserID	MealID	Rating	ReviewTime
A1029QPGAKN80T	B00DQISQX6	5.0	1497465600
A1029QPGAKN80T	B00DQISQX6	3.0	1496961600
A1041053SID37WN8GTT8	B004AUGJS8	4.0	1493664000
A1041053SID37WN8GTT8	B004AUGJS8	5.0	1495752000
A108QN0VQPX1W2	B003NRWVMC	4.0	1494009600
A108QN0VQPX1W2	B003NRWVMC	5.0	1498689600
[illegible]6	[illegible]	3.0	1497120000
[illegible]6	[illegible]	4.0	1496616000
[illegible]8	[illegible]	3.0	1495822400
[illegible]8	[illegible]	5.0	1496326400

Show 25 entries Search:

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
-rw-r--r--	root	supergroup	4.42 MB	Jul 14 15:00	2	128 MB	MealRatings_201705_201706.json
drwxr-xr-x	root	supergroup	0 B	Jul 14 15:08	0	0 B	RatingCodeList
drwxr-xr-x	root	supergroup	0 B	Jul 14 15:06	0	0 B	cleanedMealRatings
drwxr-xr-x	root	supergroup	0 B	Jul 14 16:31	0	0 B	evaluation_results
drwxr-xr-x	root	supergroup	0 B	Jul 14 15:08	0	0 B	mealZipCode
drwxr-xr-x	root	supergroup	0 B	Jul 14 16:39	0	0 B	model
drwxr-xr-x	root	supergroup	0 B	Jul 14 16:21	0	0 B	results
drwxr-xr-x	root	supergroup	0 B	Jul 14 15:10	0	0 B	testRatings
drwxr-xr-x	root	supergroup	0 B	Jul 14 15:10	0	0 B	trainRatings
drwxr-xr-x	root	supergroup	0 B	Jul 14 15:08	0	0 B	userZipCode
drwxr-xr-x	root	supergroup	0 B	Jul 14 15:10	0	0 B	validateRatings

THREE

项目二：航天公司价值客户数据分析

项目背景

国内某航空公司面临着常旅客流失，竞争力下降和航空资源未充分利用等经营危机。通过建立合理的客户价值评估模型，对客户进行分群，分析比较不同客户群的客户价值，制定相应营销策略，对不同的客户群提供个性化的客户服务是必须且有效的。

数据探索

通过对数据的观察，可发现原始数据中存在票价为空值以及票价为0，折扣率不为0，但总飞行公里数大于0的情况。

- 票价为空值——可能是数据录入异常造成的
- 票价为0，折扣率为不0，总飞行公里数大于0 ——消费记录异常。

Spark01_explore探索出异常数据的个数如下

属性名称	SUM_YR_（票价收入	SEG_KM_SUM（观测窗口的总飞行公里数）	AVG_DISCOUNT（平均折扣率）
空值记录数	591	0	0
异常值	799		

数据处理

Spark02_etl和**Spark03_etl**丢弃票价为空的记录和丢弃票价为0、平均折扣率不为0、总飞行公里数大于0的记录。

```
//获取数据表
val air_data = spark.read.table("air.air_data_base")
val sum_yr_data = air_data.filter("sum_yr_1 is not null")
sum_yr_data.write.mode("overwrite").saveAsTable("air.air_data_base_temp")
spark.stop()
```

```
//获取数据表
val air_data = spark.read.table("air.air_data_base_temp")
val sum_yr_data = air_data.filter("!(sum_yr_1=0 and avg_discount<>0.0 and seg_km_sum <> 0
sum_yr_data.show(10,false)
sum_yr_data.write.mode("overwrite").saveAsTable("air.air_data_base_avg")
spark.stop()
```

特征选择

本项目选择客户在一定时间内累积的**飞行里程M**和客户在一定时间内乘坐舱位所对应的**折扣系数的平均值C**两个特征代替消费金额。

此外，航空公司**会员入会时间**的长短在一定程度上能够影响客户价值，所以在模型中增加客户关系长度L，作为区分客户的另一特征。

将客户关系长度L，消费时间间隔R，消费频率F，飞行里程M和折扣系数的平均值C作为航空公司识别客户价值的关键特征，记为**LRFMC模型**。

Spark04_etl筛选出相关数据，并将数据存入hive的表air.lrfmc中。

模型	L	R	F	M	C
航空公司LRFMC模型	会员入会时间距观测窗口结束的月数	客户最近一次乘坐公司飞机距观测窗口结束的月数	客户在观测窗口内乘坐公司飞机的次数	客户在观测窗口内累计的飞行里程	客户在观测窗口内乘坐舱位所对应的折扣系数的平均值

数据归一化

考虑到不同特征的值范围跨度较大，因此对每列特征值进行归一化。Normalizer()是对每行进行归一化，可以看到累计里程（m)明显大于其他数值，若按行归一化则m列的数值基本一样。

```
hive> select * from lrfmc limit 5;
OK
lrfmc.l lrfmc.r lrfmc.f lrfmc.m lrfmc.c
88.94   0.03    210     580717  0.96
85.39   0.23    140     293678  1.25
85.97   0.37    135     283712  1.25
67.29   3.23    23      281336  1.09
59.68   0.17    152     309928  0.97
```

Spark04_normal将特征进行归一化。使用的是**MinMaxScaler**，它可以针对列进行行的归一化，首先利用udf函数将列的每个的特征变成vector类型，然后针对每列进行MinMaxScaler操作，最后再利用VectorAssembler 将所有将分散的vector分散的特征向量聚合成一个vector，将其聚合为"**features**"

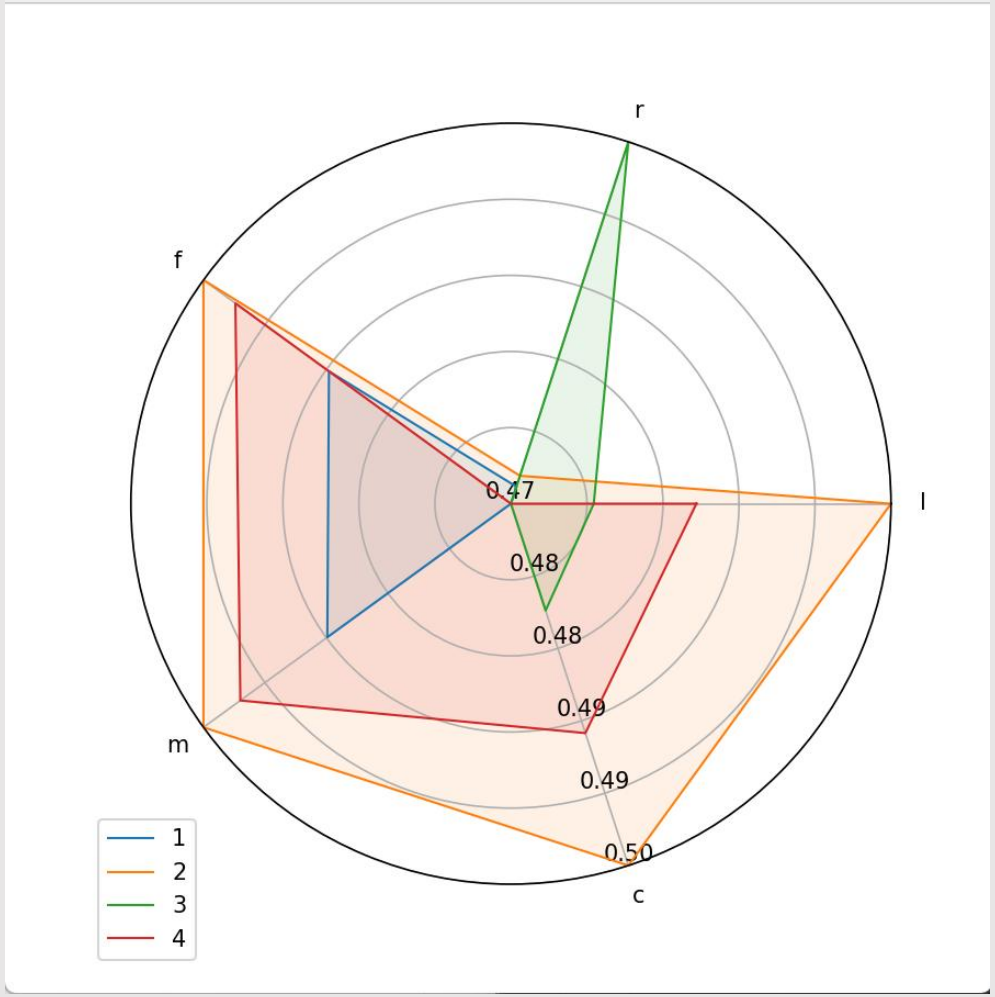
```
+--------------------+----------------------+--------------------+--------------------+--------------------
|l_scaled            |r_scaled              |f_scaled            |m_scaled            |c_scaled
+--------------------+----------------------+--------------------+--------------------+--------------------
|[0.76]              |[0.0]                 |[0.9857819905213271]|[1.0]               |[0.6399999999999999]
|[0.73]              |[0.008216926869350863]|[0.6540284360189574]|[0.5054027834975161]|[0.8333333333333333]
|[0.73]              |[0.013968775677896466]|[0.6303317535545024]|[0.4882303579397914]|[0.8333333333333333]
|[0.55]              |[0.1314708299096138]  |[0.0995260663507109]|[0.48413626972735374]|[0.7266666666666667]
```

算法选择

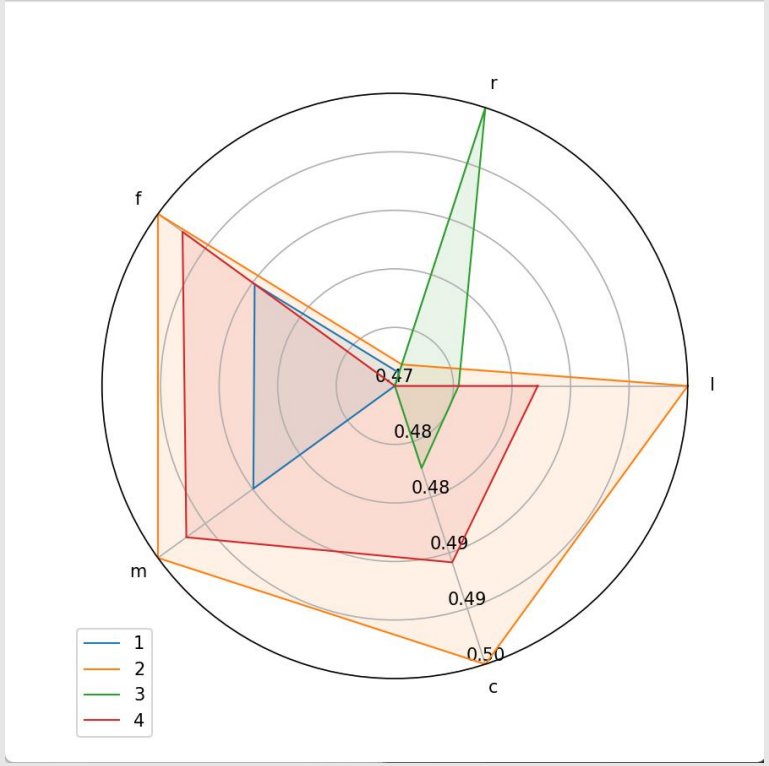
将客户分为几个类别，采用无监督学习的Kmeans算法

```
// 从 Hive 表读取归一化的数据
val dfNormalized = spark.sql( sqlText = "SELECT * FROM air.normal_lrfmc")
// 训练 KMeans 模型
val kmeans = new KMeans()
  .setK(4) // 假设我们希望分成4个簇
  .setSeed(1L)
  .setFeaturesCol("features")
  .setPredictionCol("prediction")
val model = kmeans.fit(dfNormalized)
// 显示簇中心
val centers = model.clusterCenters
println("Cluster Centers: ")
centers.foreach(println)
```

聚类中心可视化（python）



数据分析



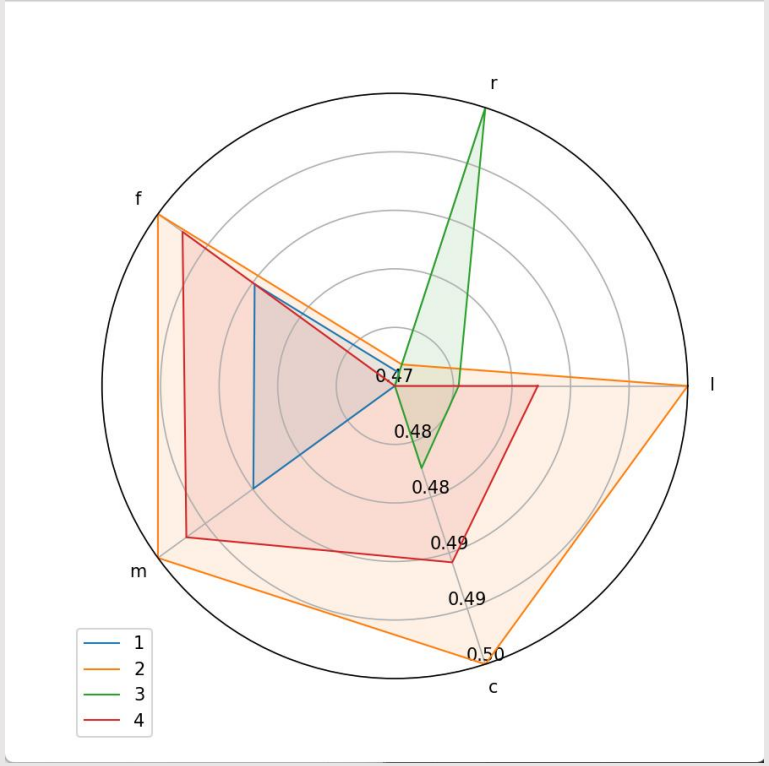
客户群2：重要保持客户，C 折扣率最高，表明仓位等级最高，其他指标较平均。

用户策略：这类客户是对航空公司最重要的VIP客户群，最理想的高价值客户类型，对公司贡献最大，所占比例最小航空公司应该将资源优先投放到它们身上，对它们进行差异化管理和一对一营销，设法提高客户的忠诚度和满意度，尽可能延长这类客户的高水平消费；

客户群4：重要发展客户，F消费频率和M飞行里程最高，C折扣系数/L会员时长，第二高.

用户策略：这类客户是公司潜在高价值客户，虽然这类客户当前的价值并不是最高，但却有很大发展潜力，公司要努力促使这类客户增加在本公司的乘机消费和合作伙伴处的消费（增加客户的钱包份额）。通过客户价值的提升，加强客户满意度，提高他们转向竞争对手的转移成本，使他们逐渐转为公司的忠诚客户；

数据分析



客户群1:重要挽留客户，有一定的消费频率和飞行里程，但会员时长较少，这类客户不确定性很高，所以掌握客户最新信息、维持与客户的互动尤为重要。

用户策略：航空公司应该根据这些客户最近的消费时间、消费次数变化情况，推测客户消费的异动状况，并列出客户名单，对其重点联系，采取一定的营销手段，延长客户的生命周期；

客户群3：低价值客户，各种指标全面低，注册时间短，里程少，消费频率低，打折比例低；唯独R高；用户策略：较长时间没做过本公司航班了，他们是航空公司低价值客户，对公司没有归属感，只有在公司机票打折促销时候才会乘坐本公司航班。应设法吸引他们提升消费等级；

FOUR

总结

Summary

项目一：某团大数据智能推荐系统

1.学会了利用spark做数据处理以及数据分析
2.初步掌握了三种协同过滤的方法，并做了初步的实践
3.项目实践中，**要避免重复造轮子**，充分利用已有的成果，大胆CV，提高工作效率

项目二：航天公司价值客户数据分析

1.掌握了分析数据的方法和选择数据的策略
2.学会了如何对每列的特征数据进行归一化
3.掌握了Kmeans的使用方法，如何设置模型的参数

谢谢观看