会员数据化运营

1 会员数据化运营概述

- 会员数据化运营几乎是所有企业的必备运营工作，企业要生存必须有会员（客户，用户）
- 会员数据化运营辅助于客户关系管理（CRM），可以用来解决以下几方面问题：
    - 会员的生命周期状态是什么?
    - 会员的核心诉求是什么?
    - 会员的转化习惯和路径是什么?
    - 会员的价值如何?
    - 如何扩大市场覆盖、获得更多的新会员?
    - 如何更好地维系老会员?
    - 应该在什么时间、采取何种措施、针对哪些会员做哪些运营活动?

2 会员数据化运营关键指标

会员数据化运营的关键指标包括会员整体指标、营销指标、活跃度指标、价值度指标、终生价值指标和异动指标。

2.1会员整体指标

1.注册会员数

2.激活会员数 3.购买会员数

- 购买会员数可以延伸出相对转化率指标
    - 注册-购买转化率：从注册到购买的会员转化比例。
    - 激活-购买转化率：从激活到购买的会员转化比例。

2.2会员营销指标

- 可营销会员数
    - 指整体会员中可通过一定方式进行会员营销以满足企业特定需求的会员数量
    - 会员可营销的方式包括：手机号、邮箱、QQ号、微信等具有可识别并可接触的信息点
    - 具备这些信息中的任何一种便形成可营销会员。
- 营销费用
    - 一般包括营销媒介费用、优惠券费用和积分兑换费用3种。
    - 营销媒介费用：特定营销媒介而产生的费用，例如短信费用、会员渠道推广费用、电子邮件费用等。
    - 优惠券费用：根据不同的使用条件和金额可以划分成多种，如30元红券、50元店铺券等。企业促销时申请的优惠券费用是会员营销费用的重要组成部分。
    - 积分兑换费用：大部分网站都有会员积分系统，会员积分通常可以兑换成金额使用
        - 如网站的积分兑换比例为20：1，即每20个积分可以兑换1元钱
        - 在开展促销活动时，除了前期投入的广告费用、促销优惠券费用外，还会包含两种情况的积分费用：
            - 一部分是积分可以直接兑换成人民币来支付订单

        - 另一部分是订单生成后会赠送一定数量的积分又形成可供兑换的金额（对企业来说是费用）
        - 这两种情况的积分兑换都构成会员营销费用。·
- 营销收入 通过会员营销渠道和会员相关运营活动产生的收入，包括电子邮件、短信、会员通知、线下二维码、特定会员优惠码等。
- 用券会员/金额/订单比例
    - 会员营销时大多数情况下都会使用优惠券，这不仅是促销销售的一种方式，也是识别不同会员订单来源的重要途径。用券类指标包括以下几种。
        - 用券会员比例：使用优惠券下单的会员占总下单会员的比例
        - 用券金额比例：使用优惠券下单的订单金额占总下单金额的比例。
        - 用券订单比例：使用优惠券下单量占总下单量的比例。 除此以外，还包括基于用券数据产生的用券用户平均订单金额、用券用户复购率等相关指标。
- 营销费率：会员营销费用占营销收入的比例。营销费率分析的目的是监督营销费用的支出情况，确保其不超出计划指标
- 每注册/订单/会员收入：
    - 每注册收入：每个注册用户带来多少收入
    - 每订单收入：每个订单带来多少收入
    - 每会员收入：每个会员带来多少收入
- 每注册/订单/会员成本 单位成本的考量是精细化业务动作的关键指标之一，包括以下几种
    - 每注册成本：每获得一个注册用户需要多少成本
    - 每订单成本：每获得一个订单需要多少成本
    - 每会员成本：每获得一个会员需要多少成本
- 除了上述单位成本指标外，还可能包括其他类型的成本，例如每挽回一个流失客户成本、每完成一个特定目标成本（例如下载企业白皮书）、每单位线索成本（例如获得一个联系方式）等。

2.3会员价值度指标

- 会员价值分群 会员价值分群是以用户价值为出发点，通过特定模型或方法将会员分为几个群体或层级。常见的分群结果如：高、中、低，钻石、黄金、白银、青铜等。 会员价值分群并不是一个真正的指标，而是给用户打标签，该标签用来显示用户的状态、层次和价值区分等。
- 复购率 复购率是一定周期内购买2次或2次以上的会员比例。不同公司对复购率的定义有所差异，基本定义逻辑分为3种，现以1个月为周期说明复购的定义。 第1种：1个月内购买2次或2次以上的会员 第2种：1个月内购买2次或2次以上，以及1个月之前有购买行为，在1个月之内又产生购买行为（可能是1次）的会员 第3种：1个月之前有购买行为，1个月之内又有购买行为的会员 以上3种定义可根据自身情况调整，同时1个月的时间周期也可根据商品或服务销售频次进行重新定义。
- 消费频次 消费频次跟复购相关，二者都是重复消费指标。消费频次是将用户的消费频率，按照次数做统计，统计结果是在一定周期内消费了不同次数，例如2次、3～5次、6～10次、11次以上。该指标可以有效分析用户对于企业的消费黏性。
- 最近一次购买时间 最近一次购买时间的含义就是字面意义，该指标也可以作为会员消费价值黏性的评估因素。如果会员距离上次的购买或消费时间过长，那么意味着用户可能处于沉默或将要流失甚至已经流失的阶段，此时应该采取措施挽回用户。
- 最近一次购买金额 最近一次购买金额和最近一次购买时间类似，该指标衡量的是用户最近一次购买或消费时的订单，该金额越大说明用户最近一次的消费能力越高。根据二八法则，20%的老会员会贡献80%的消费金额。

2.4会员终生价值指标

- 会员生命周期价值/订单量/平均订单价值 会员生命周期指标是从用户成为企业会员开始到现在的总数据统计值，该指标与任何时间周期无关，衡量的是用户完整生命周期内的价值。包括以下几种：
    - 会员生命周期价值（Lifetime Value，LTV）：用户整个生命周期内下单金额总和（利润）
    - 会员生命周期订单量：用户整个生命周期内下单量总和
    - 会员生命周期平均订单价值：用户整个生命周期内下单金额/下单量
    - 会员生命周期相关指标由于突破了时间的限制，能从整体上获得会员的宏观状态，因此是重要的宏观价值衡量指标
- LTV价值：检测运营活动的投入产出比，测试各个渠道拉新质量，以及用来监测业务发展的健康度
    - *CAC*：Customer Acquisition Cost 用户获取成本 通常和LTV一起用来判断渠道质量
    - 获取用户投入总花费/投入拉来的有效用户数
    - LTV持续增长：业务在一个良性轨道上运转，要么用户的贡献增大，要么用户流失降低
    - LTV未持续增长：业务大概率存在问题，需要考虑提升营收降低成本，或者降低用户的流失率
    - LTV计算相关指标
        - 毛利率 成本控制
        - 留存率 服务能力，产品设计，运营效率
        - 客单价 运营效率 产品能力
    - CAC：市场效率 运营效率
- 使用会员生命周期价值进行运营效率管理需要注意的问题
    - 适合使用会员生命周期价值进行运营效率管理的行业：游戏，母婴
        - 周期表现强且周期比较稳定
        - 客群生命周期的状态变化有较为明显的指针
        - 行业变化不剧烈
    - 不太适合采用会员生命周期价值进行运营效率管理的行业：社交，工具
        - 用户需求没有明确的周期化体现
        - 对用户贡献的量化能力不够强
        - 行业竞争激烈，需要不断采用新方法
    - 用户直接献价值的产品和行业：电商，周期订阅消费产品，游戏 LTV适用
    - 非用户直接贡献价值的产品或行业：社交，资讯，专注于用户的活跃度，停留时长，PV 不适用LTV
- 会员生命周期转化率
    - 在完整生命周期内完成的订单和到达网站、企业、门店的次数比例
    - 该指标衡量了用户是否具有较高的转化率
        - 用户一共到达网站100次，但是只有1次消费，那么会员生命周期转化率为1%。
- 会员生命周期剩余价值
    - 是一类预测性的指标，用来预测用户在其生命周期内还能产生多少价值
    - 该指标可以细分出很多相关指标，例如
        - 预期未来30天的会员转化率
        - 预期生命周期剩余订单价值
        - 预期7天内下单数量
        - 预计下1个订单的订单金额
        - 下一次购买的商品名称
    - 这种预测性的指标通常会基于特定的算法和模型做训练，然后预测未来的数据，其中回归和分类是主要预测性应用方法，在某些情况下也可以使用关联算法

2.5会员异动指标

- 会员流失率
  - 会员流失指会员不再购买业务、商品和服务，会员流失率= 流失的会员数量/全部会员数量的比例
  - 会员流失是一个正常现象，没有任何一个企业能做到不让一个会员流失。但是会员流失意味着企业会失去相应的利润来源，因此需要从两个方面重点关注该指标：
    - 会员流失率的数值。正常情况下会员流失率应该是一个比较小的比例，不同行业有不同的基准，各企业要制定符合行业特定的基准作为参考。
    - 会员流失率的走向。尽管会员流失不可避免，但我们仍然希望流失用户的比例越小越好，因此需要关注流失率的走向。比较好的状态是流失率处于平稳或下降状态，如果出现流失率上升的情况则需要引起警惕。
- 会员异动比: 新增会员与流失会员之间的比例关系，计算公式为：会员异动比=新增购买会员/流失会员
  - 会员异动比等于1，说明企业在一定周期内新增会员与流失会员数相等
  - 如果大于1，说明新增会员多于流失会员，这是良好的发展状态
  - 如果小于1，说明会员增长不如流失快，企业面临会员枯竭的危机

3 会员数据化运营应用场景

3.1 会员营销

数据化运营应用于会员营销主要体现在以下几个方面：

- 以信息化的方式建立基于会员的客户关系管理系统，促进所有会员数据的信息化。
- 基于用户历史消费记录，挖掘出用户潜在消费需求及消费热点。
- 基于历史数据，为会员营销活动提供策略指导和建议，促进精准营销活动的开展。
- 从会员营销结果中寻找异常订单或转化，作为识别黄牛或VIP客户的参考。
- 挖掘会员传播关系，找到口碑传播效应的关键节点。

3.2 会员关怀

数据化运营应用于会员关怀主要体现在以下几个方面：

- 分析会员行为，为会员提供个性化、精准化和差异化服务。
- 通过会员喜好分析，提高客户忠诚度、活跃度和黏性。
- 通过会员分析，预防会员流失，并找到挽回已经流失会员的方法。
- 基于会员群体行为，更好地划分会员群体属性并挖掘群体性特征。
- 基于群体用户和内容相似度，发现有价值的会员互动方式。
- 基于会员生命周期的关怀管理，促进用户终生价值最大化。

4 会员数据分析模型

4.1会员细分模型

- 将整体会员划分为不同的细分群体或类别，然后基于细分群体做管理、营销和关怀。
- 会员细分模型常用于整体会员的宏观性分析以及探索性分析，通过细分建立初步认知，为下一步的分析和应用提供基本认知。会员细分也是做精准营销的基本前提。
- 常用的细分模型包括：基于属性的方法、ABC分类法、聚类法等。

- 基于属性的方法,常用的细分属性包括:
    - 会员地域（例如北京、上海、武汉等）
    - 产品类别（例如大家电、3C数码、图书等）
    - 会员类别（例如大客户、普通客户、VIP客户等）
    - 会员性别（例如男、女、未知）
    - 会员消费等级（例如高价值会员、中价值会员、低价值会员）
    - 会员等级（例如钻石、黄金、白银）等
    - 这种细分方法可以直接利用现有会员数据库数据，无须做二次开发和计算
- ABC分类法（Activity Based Classification）
    - 根据事物的主要特征做分类排列，从而实现区别对待、区别管理的一种方法
    - ABC法则是由帕累托二八法则衍生出来的一种法则
    - 二八法则强调的是抓住关键，ABC法则强调的是分清主次，并将管理对象划分为A、B、C三类。
    - 在ABC分析法中，先将目标数据列倒序排序，然后做累积百分比统计，最后将得到的累积百分比按照下面的比例值划分为A、B、C三类。
        - A类因素：发生累计频率为0%～80%，是主要影响因素。
        - B类因素：发生累计频率为80%～90%，是次要影响因素。
        - C类因素：发生累计频率为90%～100%，是一般影响因素。
    - 下面以示例数据说明如何使用ABC分类法对会员做细分。 1）建立一个二维表格数据，数据中包括会员ID和订单金额（或其他关键指标）两列。 2）二维表格数据按照订单金额做倒序排序。 3）对订单金额列做累积百分比统计。 4）按照A、B、C划分标准将会员划分为不同的分类
- 聚类法 使用聚类法做会员细分是常用的非监督式方法，该方法无须任何先验经验，只需要指定要划分的群体数量即可。

4.2会员活跃度模型

- RFE模型基于用户的普通行为（非转化或交易行为）产生，思路与RFM类似
    - 最近一次访问时间R（Recency）会员最近一次访问或到达网站的时间
    - 访问频率F（Frequency） 用户在特定时间周期内访问或到达的频率
    - 页面互动度E（Engagements）互动度的定义可以根据不同企业的交互情况而定，例如可以定义为页面浏览量、下载量、视频播放次数等
- 在RFE模型中，由于不要求用户发生交易，因此可以做未发生登录、注册等匿名用户的行为价值分析，也可以做实名用户分析。该模型常用来做用户活跃分群或价值区分，可用于内容型（例如论坛、新闻、资讯等）企业的会员分析。
- RFM和RFE模型的实现思路相同，仅仅是计算指标发生变化。而RFE的数据来源，既可以从企业自己监控的用户行为日志获取，也可以从第三方网站分析工具获取。
- 在得到用户的RFE得分之后，有两种应用思路
    - 基于3个维度值做用户群体划分和解读，对用户的活跃度做分析
        - RFE得分为313的会员说明其访问频率低，但是每次访问时的交互都非常不错
        - 重点要做用户回访频率的提升，如通过活动邀请、精准广告投放、会员活动推荐等
    - 基于RFE的汇总得分评估所有会员的活跃度价值，并可以做活跃度排名，该得分还可以作为输入维度与其他维度一起作为其他数据分析和挖掘模型的输入变量，为分析建模提供基础

- 无论是RFM还是RFE，都不要忽略不同的消费频率、品类和周期对于结果的影响。例如，大家电的更换周期可能是2年、手机的更换频率是1年、日用消费品的消费周期却是7天。由于不同品类的商品差异性很大，最终得到的得分结果没有必然的可比性。例如偏向于购买大家电品类的RFM得分为113属于“正常现象”，因为大家电的购买属性决定了这就是一个长周期、低频、大金额的行为。

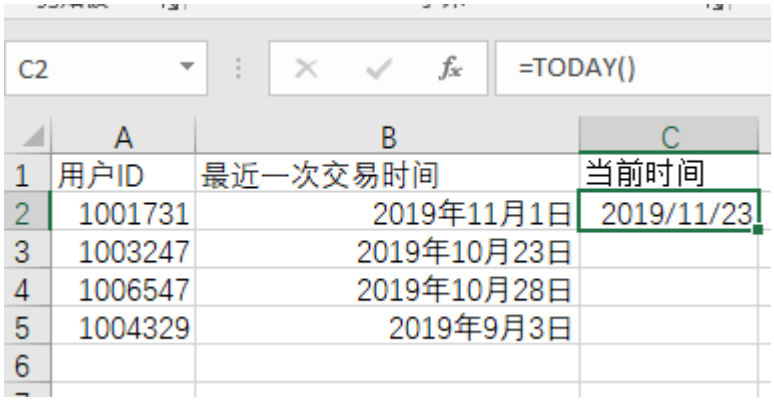
4.3 会员价值度模型

- 会员价值度用来评估用户的价值情况，是区分会员价值的重要模型和参考依据，也是衡量不同营销效果的关键指标之一。
- 价值度模型一般基于交易行为产生，衡量的是有实体转化价值的行为。常用的价值度模型是RFM
- RFM模型是根据会员
  - 最近一次购买时间R（Recency）
  - 购买频率F（Frequency）
  - 购买金额M（Monetary）计算得出RFM得分
  - 通过这3个维度来评估客户的订单活跃价值，常用来做客户分群或价值区分
  - RFM模型基于一个固定时间点来做模型分析，不同时间计算的的RFM结果可能不一样

R	F	M	用户类别
高	高	高	重要价值用户
高	低	高	重要发展用户
低	高	高	重要保持用户
低	低	高	重要挽留用户
高	高	低	一般价值用户
高	低	低	一般发展用户
低	高	低	一般保持用户
低	低	低	一般挽留用户

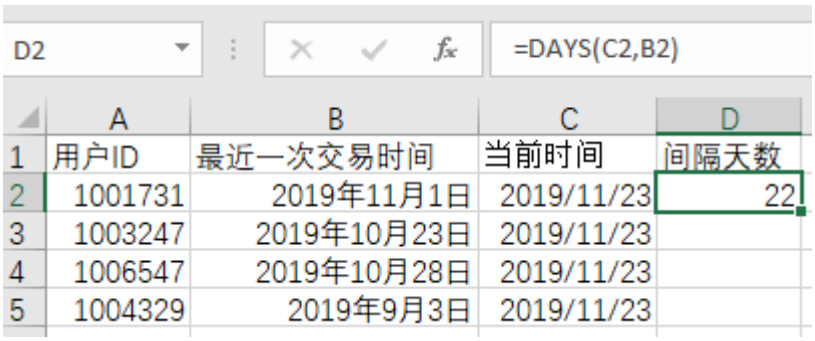
- RFM模型的基本实现过程：
  - ①设置要做计算时的截止时间节点（例如2017-5-30），用来做基于该时间的数据选取和计算。
  - ②在会员数据库中，以今天为时间界限向前推固定周期（例如1年），得到包含每个会员的会员ID、订单时间、订单金额的原始数据集。一个会员可能会产生多条订单记录。
  - ③ 数据预计算。从订单时间中找到各个会员距离截止时间节点最近的订单时间作为最近购买时间；以会员ID为维度统计每个用户的订单数量作为购买频率；将用户多个订单的订单金额求和得到总订单金额。由此得到R、F、M三个原始数据量。
  - ④ R、F、M分区。对于F和M变量来讲，值越大代表购买频率越高、订单金额越高；但对R来讲，值越小代表离截止时间节点越近，因此值越好。对R、F、M分别使用五分位（三分位也可以，分位数越多划分得越详细）法做数据分区。需要注意的是，对于R来讲需要倒过来划分，离截止时间越近的值划分越大。这样就得到每个用户的R、F、M三个变量的分位数值。
  - ⑤ 将3个值组合或相加得到总的RFM得分。对于RFM总得分的计算有两种方式，一种是直接将3个值拼接到一起，例如RFM得分为312、333、132；另一种是直接将3个值相加求得一个新的汇总值，例如RFM得分为6、9、6。
- Excel实现RFM划分案例

- 以某电商公司为例
  - R：例如：正常新用户注册1周内交易，7天是重要的值，日用品采购周期是1个月，30天是重要的值
  - F：例如：1次购买，2次购买，3次购买，4~10次，10次以上
  - M：例如：客单价300，热销单品价格240 等
- 常见的确定RFM划分区间的套路
  - 业务实际判断
  - 平均值或中位数
  - 二八法则
- 提取用户最近一次的交易时间，算出距离计算时间的差值
  - 获取当前时间=TODAY()



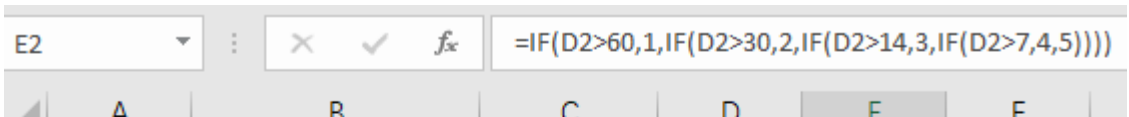
	A	B	C
1	用户ID	最近一次交易时间	当前时间
2	1001731	2019年11月1日	2019/11/23
3	1003247	2019年10月23日	
4	1006547	2019年10月28日	
5	1004329	2019年9月3日	
6			

  - 计算时间间隔



	A	B	C	D
1	用户ID	最近一次交易时间	当前时间	间隔天数
2	1001731	2019年11月1日	2019/11/23	22
3	1003247	2019年10月23日	2019/11/23	
4	1006547	2019年10月28日	2019/11/23	
5	1004329	2019年9月3日	2019/11/23	

- 根据天数长短赋予对应的R值
  - =IF(D2>60,1,IF(D2>30,2,IF(D2>14,3,IF(D2>7,4,5))))



	A	B	C	D	E	F
1	用户ID	最近一次交易时间	当前时间	间隔天数	R	
2	1001731	2019年11月1日	2019/11/23	22	3	
3	1003247	2019年10月23日	2019/11/23	31	2	
4	1006547	2019年10月28日	2019/11/23	26	3	
5	1004329	2019年9月3日	2019/11/23	81	1	

- 从历史数据中取出所有用户的购买次数，根据次数多少赋予对应的F分值
  - =IF(E2>10,5,IF(E2>3,4,IF(E2>2,3,IF(E2>1,2,1))))

=IF(E2>10,5,IF(E2>3,4,IF(E2>2,3,IF(E2>1,2,1))))

C	D	E	F	G
前时间	间隔天数	购买次数	R	F
019/11/23	22	3	3	3
019/11/23	31	5	2	4
019/11/23	26	1	3	1
019/11/23	81	12	1	5

- 从历史数据中汇总，求得该用户的交易总额，根据金额大小赋予对应的M值
  - =IF(F2>1000,5,IF(F2>500,4,IF(F2>300,3,IF(F2>230,2,1))))

=IF(F2>1000,5,IF(F2>500,4,IF(F2>300,3,IF(F2>230,2,1))))

编辑栏

C	D	E	F	G	H	I
当前时间	间隔天数	购买次数	消费金额	R	F	M
2019/11/23	22	3	230	3	3	1
2019/11/23	31	5	310	2	4	3
2019/11/23	26	1	500	3	1	3
2019/11/23	81	12	100	1	5	1

- 求出RFM的中值，例如中位数，用中值和用户的实际三值进行比较，高于中值的为高，否则为低

=IF(G2>2,"高","低")

C	D	E	F	G	H	I	J	K	L
当前时间	间隔天数	购买次数	消费金额	R	F	M	R值判断	F值判断	M值判断
2019/11/23	22	3	230	3	3	1	高	低	低
2019/11/23	31	5	310	2	4	3	低	高	高
2019/11/23	26	1	500	3	1	3	高	低	高
2019/11/23	81	12	100	1	5	1	低	高	低

- 在得到不同会员的RFM之后，根据步骤⑤产生的两种结果有两种应用思路
  - 思路1：基于3个维度值做用户群体划分和解读，对用户的价值度做分析
    - 得分为212的会员往往购买频率较低，针对购买频率低的客户应定期发送促销活动邮件
    - 得分为321的会员虽然购买频率高但是订单金额低等，这些客户往往具有较高的购买黏性，可以考虑通过关联或搭配销售的方式提升订单金额。
  - 思路2：基于RFM的汇总得分评估所有会员的价值度价值，并可以做价值度排名。同时，该得分还可以作为输入维度与其他维度一起作为其他数据分析和挖掘模型的输入变量，为分析建模提供基础。

4.4 会员流失预测模型

- 会员流失预测模型用来预测会员是否流失，是做会员生命周期管理的重要预防性应用
- 做会员流失模型的关键因素之一是要定义好“流失”，即处于何种状态、具备哪些特征的会员属于流失会员。另外，流失也要区分为永久性流失和临时性流失。常见的属于流失的状态定义示例如下：
  - 会员已经退订公司的促销活动
  - 会员打电话要求将自己的信息加入通知黑名单
  - 会员已经连续6个月没有登录过网站
  - 针对会员发送的关怀激励活动后没有得到任何有效的反馈和互动。
  - 会员最近1年内没有任何订单
  - 上述流失状态可以归为两类：

        - 一类是会员有明确的表达，不再希望接收到公司的相关信息
        - 另一类是会员没有明确的表示，但是在业务关注的主要领域内没有得到有效反馈
    - 会员流失预测模型的实现方法属于分类算法，常用算法包括逻辑回归、随机森林等

- 在做会员流失预警模型时，需要注意以下几个问题：
    - 流失会员的样本分类一定是少数类，需要注意处理样本不均衡问题。
    - 对于流失会员的预测结果，得到概率性的输出可以结合流失预测标签一起应用，因为业务方可以基于概率再结合业务经验做判断。
    - 对于参与训练模型的维度变量的选择，一定要结合业务经验，因为业务方对于特定场景的判断是影响训练模型和应用结果的关键因素之一。
    - 输入的维度变量中一定要包含发生转化前的行为数据，假如业务定义为最近6个月没有订单的客户为流失客户，那么在做预测模型时需要将用户的匿名访问、登录、页面浏览、搜索、活动咨询等转化前的数据考虑在内，而不能只考虑订单转化本身。
    - 会员流失预警模型不是一次性的，而是周期性监视和运行的，例如每天、每周或至少是每月。 通过会员流失模型得到每个会员是否属于流失标签后，可以将该结果给到会员运营人员，运营人员一般会根据业务经验做二次审查和确认，然后再通过会员挽回、激励等机制提升会员的忠诚度，延缓或防止会员流失。而关于如何挽回以及激励的问题，通常需要数据参与来帮助运营人员制定相应的策略，例如在合适的时间、以恰当的方式提供个性化的内容给特定会员，这些都需要数据的支持。

4.5会员特征分析模型

- 会员特征分析模型提供的结果可能是模糊的，也可能是明确的
    - 明确的特征，它提供了业务所要行动的细节要素，是一种具有极高落地价值的数据分析工作。
    - 模糊的特征，它指数据分析结果未提供详细的动作因素，仅指明了下一步行动方向或目标。
- 会员特征分析主要应用于以下两种业务场景
    - ①在没有任何前期经验或特定目标，希望通过整体特征分析了解会员全貌。可以通过一定方法先将用户划分为几个类别，然后再做基于类别的特征分析
        - 聚类：通过聚类将用户划分为几个群组，然后再分析不同群组的典型特征和群组间的差异性。
            - 举例：通过聚类方法将会员划分为3类，然后每个类别都有各自显著性特征，会员部门可根据不同类别做特定分析并指定群体性策略
        - 统计分析：先对整体用户做统计分析，包括描述性统计、频数分布等，了解整体数据概括。
    - ② 有明确的业务方向，希望找到能达到事件目标的会员特征，用于做进一步的会员运营。对于这类分析模型，常用的实现方法和应用包括
        - 分类：利用分类规则，例如决策树找到符合目标的关键变量及对应的变量值，进而确定会员特征。例如：收入>5400元，最近购买时间是5个月之前，总订单金额在4300元以下的会员最可能购买商品
        - 关联：使用关联规则找到不同属性、项目间的关联发生或序列发生关系，然后将会员的属性特征（频繁项集）提供给运营人员。例如：购买X商品的客户一般来自上海、购物频率为1周3次、客单价为100元以下。
        - 异常检测：使用非监督式的异常检测方法，从一堆数据中找到异常数据样本，然后将这些数据样本特征提供给运营人员做进一步确认和审查。例如：异常客户的特征往往是每次订单的商品数量超过4件、地域集中在江苏和浙江、一般拥有超过3个以上的子账户。
    - 上述两类结果，第1类结果往往作为辅助性、启发性和提示性结果，用于为运营提供进一步业务动作的思考，这种一般开始于数据工作项目的开始或业务方对数据主题的先验经验不足的情况下；第2类结果则可以作为运营下一步动作的直接“触点”。

4.6 营销响应预测模型

- 营销响应预测模型是针对营销活动展开的，通常在做会员营销活动之前，通过营销响应预测模型分析，找到可能响应活动的会员特征及整体响应的用户比例、数量和可能带来的销售额。这对在会员营销之前的有关策略制定的辅助价值非常明显。
- 营销响应预测模型的实施一般采用分类算法，常用算法包括逻辑回归、随机森林等。
- 在做营销响应模型之前，需要先收集训练所需的数据集。
    - 从所有会员中随机选择一定量的会员样本，具体数量要根据企业实际情况而定，一般情况下，至少要有1000条数据以上（同时要兼顾总体会员数量）才能满足模型训练的需要。
    - 针对选择的会员样本通过一定媒介和渠道发送营销活动信息，例如手机短信、电子邮件等。需要注意的是，一定要记录好营销活动发送的时间、频率、信息等关键运营要素，这些需要与后期的实施保持一致。
    - 收集营销活动数据。
    - 经过上述步骤收集到分类所需的样本集之后，接着就需要通过分类模型做营销响应预测，这是典型的二分类问题。
    - 通过营销响应预测模型得到的结果一般包括以下两个方向：
        - 基于模型找到最可能产生购买转化行为的会员规则特征
            - 如最近一次购买时间在3个月以内、会员等级为三级以上、总订单金额大于3000、订单量大于10的客户
            - 通过这些条件直接从数据库中筛选对应的会员列表，并可以对该列表中的会员发送营销活动。
        - 基于模型预测可能产生的订单转化数量、转化率
            - 例如选择10000个客户，会有4000个客户产生转化
            - 有转化客户的客单价（通过训练样本集选择有转化客户，然后用订单金额/会员量计算得到）大体计算出此次发送能得到的营销收入
            - 这些信息可以作为此次营销活动计划提报的数据量化指标和资源申请的数据支持

5 RFM计算案例

5.1 案例背景

- 用户价值细分是了解用户价值度的重要途径，针对交易数据分析的常用模型是RFM模型
- 业务对RFM的结果要求
    - 对用户做分组
    - 将每个组的用户特征概括和总结出来，便于后续精细化运营不同的客户群体，且根据不同群体做定制化或差异性的营销和关怀
- 规划目标将RFM的3个维度分别做3个区间的离散化
    - 用户群体最大有3×3×3=27个
    - 划分区间过多则不利于用户群体的拆分
    - 区间过少则可能导致每个特征上的用户区分不显著
- 交付结果
    - 给业务部门做运营的分析结果要导出为Excel文件，用于做后续分析和二次加工使用
    - RFM的结果还会供其他模型的建模使用，RFM本身的结果可以作为新的局部性特征，因此数据的输出需要有本地文件和写数据库两种方式
- 数据说明
    - 选择近4年订单数据，从不同的年份对比不同时间下各个分组的绝对值变化情况，方便了解会员的波动
    - 案例的输入源数据sales.xlsx

- 程序输出RFM得分数据写入本地文件sales_rfm_score.xlsx和MySQL数据库sales_rfm_score表中

5.2 用到的技术点

- 通过Python代码手动实现RFM模型，主要用到的库包括
  - time、numpy和pandas
  - 在实现RFM组合时，我们使用了sklearn的随机森林库来计算R、F、M的权重
  - 在结果展示时使用了pyecharts的3D柱形图

5.3 案例数据

- 案例数据是某企业从2015年到2018年共4年的用户订单抽样数据，数据来源于销售系统
- 数据在Excel中包含5个sheet，前4个sheet以年份为单位存储为单个sheet中，最后一张会员等级表为用户的等级表
- 前4张表的数据概要如下。
  - 特征变量数：4
  - 数据记录数：30774/41278/50839/81349
  - 是否有NA值：有
  - 是否有异常值：有

    ·会员ID：每个会员的ID唯一，由纯数字组成。 ·提交日期：订单日提交日期。 ·订单号：订单ID，每个订单的ID唯一，由纯数字组成。 ·订单金额：订单金额，浮点型数据。
- 具体数据特征如下：
  - 会员ID：每个会员的ID唯一，整型
  - 提交日期：订单日提交日期
  - 订单号：订单ID，每个订单的ID唯一，整型
  - 订单金额：订单金额，浮点型数据
- 会员登记表中是所有会员的会员ID对应会员等级的情况，包括以下两个字段
  - 会员ID：该ID可与前面的订单表中的会员ID关联
  - 会员等级：会员等级以数字区分，数字越大，级别越高

5.4 代码

- **导入模块**

```
import time  # 时间库

import numpy as np  # numpy库
import pandas as pd  # pandas库
import pymysql  # mysql连接库
from sklearn.ensemble import RandomForestClassifier # RF库

from pyecharts import Bar3D # 3D柱形图
```

- 用到了6个库：time、numpy、pandas、pymysql、sklearn和pyecharts。
  - time：用来记录插入数据库时的当前日期
  - numpy：用来做基本数据处理等

- pandas：有关日期转换、数据格式化处理、主要RFM计算过程等
- pymysql：数据库连接工具，读写MySQL数据库。
- sklearn：使用其中的随机森林库
- pyecharts：展示3D柱形图

- **读取数据**

```
sheet_names = ['2015','2016','2017','2018','会员等级']
sheet_datas = [pd.read_excel('sales.xlsx',sheet_name=i) for i in sheet_names]
```

- **数据审查**

```
for each_name,each_data in zip(sheet_names,sheet_datas):
    print('[data summary for ==============={}===============]'.format(each_name))
    print('Overview:','\n',each_data.head(4))# 展示数据前4条
    print('DESC:','\n',each_data.describe())# 数据描述性信息
    print('NA records',each_data.isnull().any(axis=1).sum()) # 缺失值记录数
    print('Dtypes',each_data.dtypes) # 数据类型
```

输出结果

```
[data summary for ==============2015===============]
Overview:
           会员ID          订单号        提交日期     订单金额
0  15278002468  3000304681 2015-01-01   499.0
1  39236378972  3000305791 2015-01-01  2588.0
2  38722039578  3000641787 2015-01-01   498.0
3  11049640063  3000798913 2015-01-01  1572.0
DESC:
                会员ID            订单号            订单金额
count  3.077400e+04  3.077400e+04   30774.000000
mean   2.918779e+10  4.020414e+09     960.991161
std    1.385333e+10  2.630510e+08    2068.107231
min    2.670000e+02  3.000305e+09       0.500000
25%    1.944122e+10  3.885510e+09      59.000000
50%    3.746545e+10  4.117491e+09     139.000000
75%    3.923593e+10  4.234882e+09     899.000000
max    3.954613e+10  4.282025e+09  111750.000000
NA records 0
Dtypes 会员ID              int64
订单号               int64
提交日期    datetime64[ns]
订单金额           float64
dtype: object
[data summary for ==============2016===============]
Overview:
           会员ID          订单号        提交日期     订单金额
0  39288120141  4282025766 2016-01-01    76.0
1  39293812118  4282037929 2016-01-01  7599.0
2  27596340905  4282038740 2016-01-01   802.0
3  15111475509  4282043819 2016-01-01    65.0
DESC:
```

```
                会员ID           订单号           订单金额
count  4.127800e+04  4.127800e+04   41277.000000
mean   2.908415e+10  4.313583e+09     957.106694
std    1.389468e+10  1.094572e+07    2478.560036
min    8.100000e+01  4.282026e+09       0.100000
25%    1.934990e+10  4.309457e+09      59.000000
50%    3.730339e+10  4.317545e+09     147.000000
75%    3.923182e+10  4.321132e+09     888.000000
max    3.954554e+10  4.324911e+09  174900.000000
NA records 1
Dtypes 会员ID             int64
订单号              int64
提交日期    datetime64[ns]
订单金额           float64
dtype: object
[data summary for ==============2017===============]
Overview:
           会员ID         订单号        提交日期    订单金额
0  38765290840  4324911135 2017-01-01  1799.0
1  39305832102  4324911213 2017-01-01   369.0
2  34190994969  4324911251 2017-01-01   189.0
3  38986333210  4324911283 2017-01-01   169.0
DESC:
                会员ID           订单号           订单金额
count  5.083900e+04  5.083900e+04   50839.000000
mean   2.882368e+10  4.332466e+09     963.587872
std    1.409416e+10  4.404350e+06    2178.727261
min    2.780000e+02  4.324911e+09       0.300000
25%    1.869274e+10  4.328415e+09      59.000000
50%    3.688044e+10  4.331989e+09     149.000000
75%    3.923020e+10  4.337515e+09     898.000000
max    3.954554e+10  4.338764e+09  123609.000000
NA records 0
Dtypes 会员ID             int64
订单号              int64
提交日期    datetime64[ns]
订单金额           float64
dtype: object
[data summary for ==============2018===============]
Overview:
           会员ID         订单号        提交日期    订单金额
0  39229691808  4338764262 2018-01-01  3646.0
1  39293668916  4338764363 2018-01-01  3999.0
2  35059646224  4338764376 2018-01-01    10.1
3      1084397  4338770013 2018-01-01   828.0
DESC:
                会员ID           订单号           订单金额
count  8.134900e+04  8.134900e+04   81348.000000
mean   2.902317e+10  4.348372e+09     966.582792
std    1.404116e+10  4.183774e+06    2204.969534
min    2.780000e+02  4.338764e+09       0.000000
25%    1.902755e+10  4.345654e+09      60.000000
50%    3.740121e+10  4.349448e+09     149.000000
```

```
75%    3.923380e+10  4.351639e+09     899.000000
max    3.954614e+10  4.354235e+09  174900.000000
NA records 1
Dtypes 会员ID              int64
订单号               int64
提交日期    datetime64[ns]
订单金额           float64
dtype: object
[data summary for ==============会员等级===============]
Overview:
           会员ID  会员等级
0       100090     3
1  10012905801     1
2  10012935109     1
3  10013498043     1
DESC:
                会员ID           会员等级
count  1.543850e+05  154385.000000
mean   2.980055e+10       2.259701
std    1.365654e+10       1.346408
min    8.100000e+01       1.000000
25%    2.213894e+10       1.000000
50%    3.833022e+10       2.000000
75%    3.927932e+10       3.000000
max    3.954614e+10       5.000000
NA records 0
Dtypes 会员ID    int64
会员等级    int64
dtype: object
```

- 结果说明
  - 每个sheet中的数据都能正常读取，无任何错误
  - 日期列（提交日期）已经被自动识别为日期格式，后期不必转换
  - 订单金额的分布是不均匀的，里面有明显的极大值
    - 例如2016年的数据中，最大值为174900，最小值仅为0.1
    - 极大极小值相差过大，数据会受极值影响
  - 订单金额中的最小值包括0、0.1这样的金额，可能为非正常订单，与业务方沟通后确认
    - 最大值的订单金额有效，通常是客户一次性购买多个大家电商品
    - 而订单金额为0.1元这类使用优惠券支付的订单，没有实际意义
    - 除此之外，所有低于1元的订单均有这个问题，因此需要在后续处理中去掉
- 有的表中存在缺失值记录，但数量不多，选择丢弃或填充均可

- **数据预处理**

```
# 去除缺失值和异常值
for ind,each_data in enumerate(sheet_datas[:-1]):
    sheet_datas[ind] = each_data.dropna()# 丢弃缺失值记录
    sheet_datas[ind] = each_data[each_data['订单金额'] > 1]# 丢弃订单金额<=1的记录
    sheet_datas[ind]['max_year_date'] = each_data['提交日期'].max() # 增加一列最大日期
值
```

- 通过for循环配合enumerate方法，获得每个可迭代元素的索引和具体值
- 处理缺失值和异常值只针对订单数据，因此sheet_datas通过索引实现不包含最后一个对象（即会员等级表）
- 直接将each_data使用dropna丢弃缺失值后的dataframe代原来sheet_datas中的dataframe
- 使用each_data[each_data['订单金额']>1]来过滤出包含订单金额>1的记录数，然后替换原来sheet_datas中的dataframe
- 最后一行代码的目的是在每个年份的数据中新增一列max_year_date，通过each_data['提交日期'].max()获取一年中日期的最大值，这样方便后续针对每年的数据分别做RFM计算，而不是针对4年的数据统一做RFM计算。

```
# 汇总所有数据
data_merge = pd.concat(sheet_datas[:-1],axis=0)
# 获取各自年份数据
data_merge['date_interval'] = data_merge['max_year_date']-data_merge['提交日期']
data_merge['year'] = data_merge['提交日期'].dt.year
# 转换日期间隔为数字
data_merge['date_interval'] = data_merge['date_interval'].apply(lambda x: x.days) #
转换日期间隔为数字
#data_merge.head()
```

- 汇总所有数据: 将4年的数据使用pd.concat方法合并为一个完整的dataframe data_merge，后续的所有计算都能基于同一个dataframe进行，而不用写循环代码段对每个年份的数据单独计算
- 获取各自年份数据:
  - 先计算各自年份的最大日期与每个行的日期的差，得到日期间隔
  - 再增加一列新的字段，为每个记录行发生的年份，使用data_merge['提交日期'].dt.year实现
- 关于pandas的 datetime类型
  - dt是pandas中Series时间序列datetime类属性的访问对象
  - 除了代码中用到的year外，还包括：date、dayofweek、dayofyear、days_in_month、freq、days、hour、microsecond、minute、month、quarter、second、time、week、weekday、weekday_name、weekofyear等
- 转换日期间隔为数字：data_merge['date_interval'].apply(lambda x: x.days) 是将data_merge['date_interval']的时间间隔转换为数值型计算对象，这里使用了apply方法。Apply方法是对某个Pandas对象（dataframe或Series）使用自定义函数

```
# 按会员ID做汇总
rfm_gb = data_merge.groupby(['year','会员ID'],as_index=False).agg(
    {'date_interval': 'min',  # 计算最近一次订单时间
     '提交日期': 'count', # 计算订单频率
     '订单金额': 'sum'})  # 计算订单总金额
# 重命名列名
rfm_gb.columns =  ['year','会员ID','r','f','m']
rfm_gb.head()
'''
   year  会员ID    r  f    m
0  2015   267  197  2   105.0
1  2015   282  251  1    29.7
2  2015   283  340  1  5398.0
3  2015   343  300  1   118.0
4  2015   525   37  3   213.0
```

```
'''
```

- 上面代码框中的第一行代码，是基于年份和会员ID，分别做RFM原始值的聚合计算
- 这里使用groupby分组，以year和会员ID为联合主键，设置as_index=False意味着year和会员ID不作为index列，而是普通的数据框结果列。后面的agg方法实际上是一个“批量”聚合功能的函数，它实现了对date_interval、提交日期、订单金额三列分别以min、count、sum做聚合计算的功能。否则，我们需要分别写3条goupby来实现3个聚合计算

- **确定RFM划分区间** 在做RFM划分时，基本逻辑是分别对R、F、M做离散化操作，然后再计算RFM。而离散化本身有多种方法可选，由于我们要对数据做RFM离散化，因此需要先看下数据的基本分布状态

```
# 查看数据分布
desc_pd = rfm_gb.iloc[:,2:].describe().T
print(desc_pd)
# 定义区间边界
r_bins = [-1,79,255,365] # 注意起始边界小于最小值
f_bins = [0,2,5,130]
m_bins = [0,69,1199,206252]

'''
      count         mean          std  min   25%    50%     75%       max
r  148591.0   165.524043   101.988472  0.0  79.0  156.0   255.0     365.0
f  148591.0     1.365002     2.626953  1.0   1.0    1.0     1.0     130.0
m  148591.0  1323.741329  3753.906883  1.5  69.0  189.0  1199.0  206251.8
'''
```

- 从基本概要看出
  - 汇总后的数据总共有14万条
  - r和m的数据分布相对较为离散，表现在min、25%、50%、75%和max的数据没有特别集中
  - 而从f（购买频率）则可以看出，大部分用户的分布都趋近于1，表现是从min到75%的分段值都是1且mean（均值）才为1.365
  - 计划选择25%和75%作为区间划分的2个边界值
- f的分布情况说明
  - r和m本身能较好地区分用户特征，而f则无法区分（大量的用户只有1个订单）
  - 行业属性（大家电）原因，1年购买1次比较普遍（其中包含新客户以及老客户在当年的第1次购买）
  - 与业务部门沟通，划分时可以使用2和5来作为边界
    - 业务部门认为当年购买>=2次可被定义为复购用户（而非累计订单的数量计算复购用户）
    - 业务部门认为普通用户购买5次已经是非常高的次数，超过该次数就属于非常高价值用户群体
    - 该值是基于业务经验和日常数据报表获得的
- 区间边界的基本原则如下
  - 中间2个边界值：r和m是分别通过25%和75%的值获取的，f是业务与数据部门定义的。
  - 最小值边界：比各个维度的最小值小即可。
  - 最大值边界：大于等于各个维度的最大值即可
  - 最小值边界为什么要小于各个维度的最小值:
    - 这是由于在边界上的数据归属有一个基本准则，要么属于区间左侧，要么属于区间右侧。如，f_bins中的2处于边界上，要么属于左侧区间，要么属于右侧区间

        - 在后续使用pd.cut方法中，对于自定义边界实行的是左开右闭的原则，即数据属于右侧区间，f_bins中的2就属于右侧区间。最左侧的值是无法划分为任何区间的，因此，在定义最小值时，一定要将最小值的边界值
        - 举例：[1，2，3，4，5]，假如数据划分的区间边界是[1，3，5]，即划分为2份
            - 其中的2/3被划分到(1，3]区间中
            - 3/4/5被划分到(3，5]区间中
            - 1无法划分到任何一个正常区间内
- **计算RFM因子权重**
    - 在计算RFM组合得分时，我们可以直接将结果组合为一个新的分组，例如322 、132
    - 加权求和得到一个新的rfm得分指标时，则必须要确定一个权重值
    - 通过会员等级来要确定rfm三个维度的权重
    - 在指定会员等级时，各个公司都已经综合考虑到了多种与公司整体利益相关的因素，设置各种会员权益都与会员等级有关
        - 免运费门槛、优惠券使用、特殊商品优惠价格、会员活动和营销等
        - 基本思路是，建立一个rfm三个维度与会员等级的分类模型，然后通过模型输出维度的权重。

```
# 匹配会员等级和rfm得分
rfm_merge = pd.merge(rfm_gb,sheet_datas[-1],on='会员ID',how='inner')
# rf获得rfm因子得分
clf = RandomForestClassifier()
clf = clf.fit(rfm_merge[['r','f','m']],rfm_merge['会员等级'])
weights = clf.feature_importances_
print('feature importance:',weights)
#feature importance: [0.40596048 0.00533821 0.58870131]
```

    - 上述过程非常简单，先建立rf模型对象，然后将rfm三列作为特征，将会员等级作为目标输入模型中做训练，最后通过模型的feature_importances_获得权重信息
    - 结果分析
        - 用户的等级首先侧重于会员的价值贡献度（实际订单的贡献）
        - 其次是新近程度，最后是频次。
        - 大多数公司的整体会员等级主要侧重于购物金额（京东，淘宝）
- **RFM计算过程**

```
# RFM分箱得分
rfm_gb['r_score'] = pd.cut(rfm_gb['r'], r_bins, labels=[i for i in
range(len(r_bins)-1,0,-1)])  # 计算R得分
rfm_gb['f_score'] = pd.cut(rfm_gb['f'], f_bins, labels=[i+1 for i in
range(len(f_bins)-1)])  # 计算F得分
rfm_gb['m_score'] = pd.cut(rfm_gb['m'], m_bins, labels=[i+1 for i in
range(len(m_bins)-1)])  # 计算M得分
```

    - 每个rfm的过程使用了pd.cut方法，基于自定义的边界区间做划分
    - labels用来显示每个离散化后的具体值。F和M的规则是值越大，等级越高
    - 而R的规则是值越小，等级越高，因此labels的规则与F和M相反
    - 在labels指定时需要注意，4个区间的结果是划分为3份

```
# 计算RFM总得分
# 方法一：加权得分
rfm_gb = rfm_gb.apply(np.int32) # cate转数值
rfm_gb['rfm_score'] = rfm_gb['r_score'] * weights[0] + rfm_gb['f_score'] *
weights[1] + rfm_gb['m_score'] * weights[2]
```

- 加权得分：可以对所有会员从一个维度做衡量，尤其用于价值度排序
- 将每个得分值转换为np.int32类型，否则上面pd.cut的实现结果是类别型结果
- 将rfm三列分别乘以权重，得到新的rfm加权得分

```
# 方法二：RFM组合
rfm_gb['r_score'] = rfm_gb['r_score'].astype(np.str)
rfm_gb['f_score'] = rfm_gb['f_score'].astype(np.str)
rfm_gb['m_score'] = rfm_gb['m_score'].astype(np.str)
rfm_gb['rfm_group'] = rfm_gb['r_score'].str.cat(rfm_gb['f_score']).str.cat(
rfm_gb['m_score'])
```

- 将3列作为字符串组合为新的分组
  - 代码中，先针对3列使用astype方法将数值型转换为字符串型
  - 然后使用pandas的字符串处理库str中的cat方法做字符串合并，该方法可以将右侧的数据合并到左侧
  - 再连续使用两个str.cat方法得到总的R、F、M字符串组合
- Series.str.cat(others=None, sep=None, na_rep=None) 参数: others : 列表或复合列表,默认为None,如果为None则连接本身的元素 sep : 字符串 或者None,默认为None na_rep : 字符串或者 None, 默认None。如果为None缺失值将被忽略。 返回值: concat : 序列(Series)/索引(Index)/字符串(str)

```
#如果连接的是两个序列，则会对应
>>> pd.Series(['a', 'b', 'c']).str.cat(['A', 'B', 'C'], sep=',')
0    a,A
1    b,B
2    c,C
dtype: object
#否则则会连接自身序列里的值
>>> pd.Series(['a', 'b', 'c']).str.cat(sep=',')
'a,b,c'
#也可以同时连接复合列表
>>> pd.Series(['a', 'b']).str.cat([['x', 'y'], ['1', '2']], sep=',')
0    a,x,1
1    b,y,2
dtype: object
```

- **保存RFM结果到Excel**

```
rfm_gb.to_excel('sales_rfm_score1.xlsx')  # 保存数据为Excel
```

- **写数据到数据库**

```
# 数据库信息
config = {'host': '127.0.0.1',  # 默认127.0.0.1
          'user': 'root',  # 用户名
          'password': 'MyNewPass',  # 密码
          'port': 3306,  # 端口，默认为3306
          'database': 'test',  # 数据库名称
          'charset': 'utf8'  # 字符编码
          }
```

```
# 建表操作
con = pymysql.connect(**config)  # 建立mysql连接
cursor = con.cursor()  # 获得游标
cursor.execute("show tables")  # 查询表
table_list = [t[0] for t in cursor.fetchall()]  # 读出所有库
# 查找数据库是否存在目标表，如果没有则新建
table_name = 'sales_rfm_score'  # 要写库的表名
if not table_name in table_list:  # 如果目标表没有创建
    cursor.execute('''
    CREATE TABLE %s (
    userid                VARCHAR(20),
    r_score               int(2),
    f_score               int(2),
    m_score               int(2),
    rfm_score               DECIMAL(10,2),
    rfm_group               VARCHAR(10),
    insert_date               VARCHAR(20)
    )ENGINE=InnoDB DEFAULT CHARSET=utf8
    ''' % table_name)  # 创建新表
```

```
# 梳理数据
write_db_data = rfm_gb[['会员
ID','r_score','f_score','m_score','rfm_score','rfm_group']] # 主要数据
timestamp = time.strftime('%Y-%m-%d', time.localtime(time.time()))  # 日期
```

```
# 写库
for each_value in write_db_data.values:
    insert_sql = "INSERT INTO `%s` VALUES ('%s',%s,%s,%s,%s,'%s','%s')" % \
                 (table_name, each_value[0], each_value[1], each_value[2], \
                  each_value[3],each_value[4],each_value[5],
                  timestamp)  # 写库SQL依据
    cursor.execute(insert_sql)  # 执行SQL语句，execute函数里面要用双引号
    con.commit()  # 提交命令
cursor.close()  # 关闭游标
con.close()  # 关闭数据库连接
```

- **RFM图形展示**
  - 为了更好地了解不同周期下RFM分组人数的变化，通过3D柱形图展示结果
  - 展示结果时只有3个维度，分别是年份、rfm分组和用户数量。

```
# 图形数据汇总
display_data = rfm_gb.groupby(['rfm_group','year'],as_index=False)['会员
ID'].count()
display_data.columns = ['rfm_group','year','number']
display_data['rfm_group'] = display_data['rfm_group'].astype(np.int32)
display_data.head()
```

- 第1行代码使用数据框的groupby以rfm_group和year为联合对象，以会员ID会为计算维度做计数，得到每个RFM分组、年份下的会员数量
- 第2行代码对结果列重命名
- 第3行代码将rfm分组列转换为int32形式 低版本写法：

```
# 显示图形
from pyecharts import Bar3D  # pyecharts==0.1.9.4
bar3d = Bar3D("", width=900, height=600)
range_color = ['#313695', '#4575b4', '#74add1', '#abd9e9', '#e0f3f8', '#ffffbf',
               '#fee090', '#fdae61', '#f46d43', '#d73027', '#a50026']
bar3d.add(
    "rfm分组结果",
    "",
    "",
    [d.tolist() for d in display_data.values],
    is_visualmap=True,
    visual_range=[0, display_data['number'].max()],
    visual_range_color=range_color,
    grid3d_width=200,
    grid3d_height=80,
    grid3d_depth=80
)
bar3d
```

高版本写法：

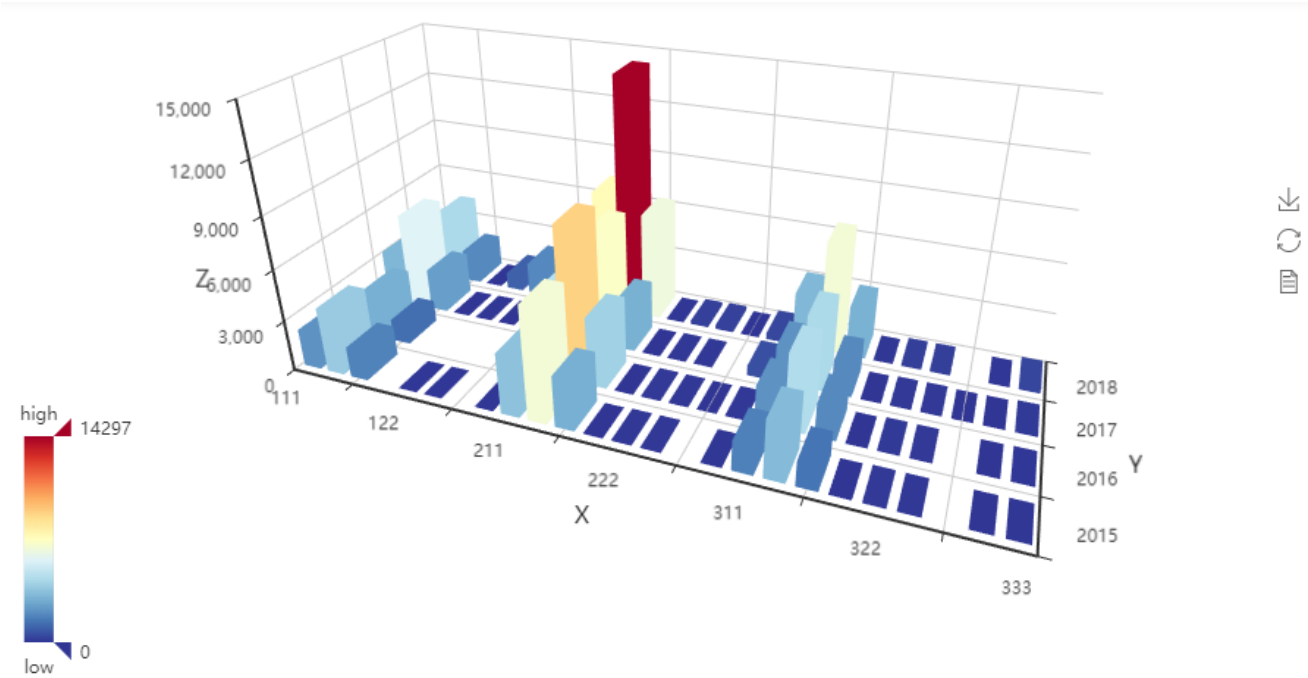
```
import pyecharts.options as opts
from pyecharts.charts import Bar3D  # 高版本 pyecharts==1.9.1 -i
x_data = display_data.rfm_group.values.tolist()
y_data = display_data.year.values.tolist()
data = [d.tolist() for d in display_data.values]

res = (
    Bar3D(init_opts=opts.InitOpts(width="900px", height="600px")).add(
        series_name="",
        data=data,

    ).set_global_opts(
        title_opts=opts.TitleOpts("标准3D柱状图"),
        visualmap_opts=opts.VisualMapOpts(
            max_=14297,
            range_color=[
                '#313695', '#4575b4', '#74add1', '#abd9e9', '#e0f3f8', '#ffffbf',
                '#fee090', '#fdae61', '#f46d43', '#d73027', '#a50026',
```

```
            ],
        )
    )
)
# res.render_notebook()
res.render(path='RFM_pyecharts.html')
```

- 第1行设置了一个3D柱形图对象，标题为空，宽和高分别是900像素和600像素
- 第2行和第3行的range_color设置了一个颜色列表，该列表用来展示不同数据下的变化范围。一般情况下，该类颜色列表都是有规律地变化颜色，例如从红到蓝的渐变色，可以表示热度或数量从多到少的变化。
- bar3d.add方法插入3D图形所需的数据并具体设置展示信息
    - 第1个参数设置了图例名称
    - 第2和第3个参数设置x轴和y轴的标签数据，这里留空
    - 第4个参数是3D柱形图的数据，该数据是一个由列表组成的列表，即列表嵌套列表，内部的每个列表是一条包含X、Y、Z轴3个维度的数据，这里通过列表推导式将display_data的每行数据读出，并使用tolist方法转换为列表
    - 第5个参数is_visualmap为True，表示设置显示可调节视图功能；
    - 第6个参数visual_range设置了可调节视图功能的最大值和最小值区间，这里的最大值取的是各个分组数量的最大值；
    - 第7个参数visual_range_color设置可调节的颜色列表区间；
    - 最后3个参数grid3d_width、grid3d_height和grid3d_depth设置3D图X、Y和Z轴各自的长度
- bar3d 绘制3D图像



- 输出3D图像中
    - X轴为RFM分组、Y轴为年份、Z轴为用户数量
    - 该3D图可旋转、缩放，以便查看不同细节
    - 调节左侧的滑块条，用来显示或不显示特定数量的分组结果

5.5 案例结论

- **基于图形的交互式分析**

    - 重点人群分布：
        - 在整个分组中，212群体的用户是相对集中且变化最大的
        - 从2016年到2017年用户群体数量变化不大，但到2018年增长了近一倍
        - 这部分人群将作为重点分析人群
    - 重点分组分布：
        - 除了212人群外，312、213、211及112人群都在各个年份占据很大数量
        - 虽然各自规模不大，但组合起来的总量超过212本身，也要重点做分析。
        - 如果拖动左侧的滑块，仅过滤出用户数量在4085以内的分组结果。观察图形发现，很多分组的人群非常少，甚至没有人
- **基于RFM分组结果的分析** 通过RFM分组的Excel结果数据，更进一步确定要分析的主要目标群体。
    - 打开导出的sales_rfm_score.xlsx，然后建立数据透视表
        - ①将rfm_group拖入“行”区块
        - ②将会员ID拖入“值”区块
        - ③单击会员ID项
        - ④在弹出的窗口中选择“值字段设置”
        - ⑤在弹窗中选择“计算类型”为“计数”
        - ⑥单击“确定”按钮。到此数据就以RFM分组为主体，以用户数量作为分类汇总了
    - 鼠标左键单击“计数项：会员ID”中的任意数值，单击顶部的排序功能，按从大到小排序
    - 排序后发现，会员在所有分组的分布不均匀（与图形结果一致），需要按百分点汇总，获取每组占比
        - 在透视表的“计数项：会员ID”列单击任意数值
        - 然后单击右键，在弹出的菜单中选择“值显示方式”
        - 从右侧菜单中选择“父行汇总的百分比”

行标签	计数项:会员ID
212	24.79%
211	12.80%
312	12.55%
112	11.34%
213	11.02%
311	6.24%
111	6.14%
313	5.61%
113	5.07%
123	1.30%
233	0.70%
122	0.68%
333	0.33%
133	0.32%
322	0.28%

- **RFM用户特征分析**
    - 经过上面的分析，得到了要分析的重点客户群体。可根据用户的量级分为两类
        - 第1类是用户群体占比超过10%的群体
        - 第2类是占比在个位数的群体。这两类人由于量级不同，因此需要分别有针对性的策略场景。
        - 除此以外，我们还会增加第3类人群，虽然从用户量级上小，但是单个人的价值度非常高。
    - 第1类人群：占比超过10%的群体。由于这类人群基数大，必须采取批量操作和运营的方式落地运营策略，一般需要通过系统或产品实现，而不能主要依赖于人工
        - 212：可发展的一般性群体

        - 购买新近度和订单金额一般，且购买频率低
        - 数量大，以及在新近度和订单金额上都可以，可采取常规性的礼品兑换和赠送、购物社区活动、签到、免运费等手段维持并提升其消费状态。
    - 211：可发展的低价值群体
        - 相对于212群体在订单金额上表现略差
        - 在211群体策略的基础上，可以增加与订单相关的刺激措施，例如组合商品优惠券发送、积分购买商品等
    - 312：有潜力的一般性群体。
        - 购买新近度高，说明最近一次购买时间间隔短，对于公司尚有比较熟悉的接触渠道和认知状态
        - 购物频率低，说明对网站的忠诚度一般
        - 订单金额处于中等层级，说明其还具有可提升的空间。
        - 可借助其最近购买的商品，定制一些与上次购买相关的商品，通过向上销售等策略提升购买频次和订单金额
    - 112：可挽回的一般性群体
        - 购买新近度较低，说明距离上次购买时间较长，很可能用户已经处于沉默或预流失、流失阶段
        - 购物频率低，说明对网站的忠诚度一般
        - 订单金额处于中等层级，说明其还可能具有可提升的空间
        - 对这部分群体的策略：
            - 首先是通过多种方式（例如邮件、短信等）触达客户并挽回，
            - 然后通过针对流失客户的专享优惠（例如流失用户专享优惠券）措施促进其消费
            - 通过增加接触频次和刺激力度的方式，增加用户的回访、复购以及订单价值回报
    - 213：可发展的高价值群体
        - 重点是提升购物频率
        - 可指定不同的活动或事件来触达用户，促进其回访和购买
        - 例如不同的节日活动、每周新品推送、高价值客户专享商品等。
- 第2类人群：占比为1%～10%的群体。这部分人群数量适中，在落地时无论是产品还是人工都可接入
    - 311：有潜力的低价值群体
        - 与211群体类似，但在购物新近度上更好，因此对其可采取相同的策略
        - 在这类群体的最近接触渠道上可以增加营销或广告资源投入，通过这些渠道再次将客户引入网站完成消费。
    - 111：这是一类在各个维度上都比较差的客户群体
        - 会在其他各个群体策略和管理都落地后才考虑他们
        - 主要策略
            - 先通过多种策略挽回客户
            - 然后为客户推送与其类似的其他群体，或者当前热销的商品或折扣非常大的商品
            - 在刺激消费时，可根据其消费水平、品类等情况，有针对性地设置商品暴露条件
            - 先在优惠券及优惠商品的综合刺激下使其实现消费，再考虑消费频率以及订单金额的提升。
    - 313：有潜力的高价值群体
        - 消费新近度高且订单金额高，但购买频率低，只要提升购买频次，用户群体贡献价值就会倍增
        - 提升购买频率上，除了在其最近一次的接触渠道上增加曝光外，与最近一次渠道相关的其他关联访问渠道也要考虑增加营销资源
        - 213中的策略也要组合应用其中
    - 113：可挽回的高价值群体

        - 112群体类似，但订单金额贡献更高，因此除了应用112中的策略外，可增加部分人工的参与来挽回这些高价值客户，例如线下访谈、客户电话沟通等
    - 第3类群体：占比非常少，但却是非常重要的群体
        - 333：绝对忠诚的高价值群体
            - 用户绝对数量只有355，但由于其各方面表现非常突出，因此可以倾斜更多的资源
                - 设计VIP服务、专享服务、绿色通道等
                - 针对这部分人群的高价值附加服务的推荐也是提升其价值的重点策略
        - 233、223和133：一般性的高价值群体
            - 主要着手点是提升新近购买度，即促进其实现最近一次的购买
            - 可通过电话、客户拜访、线下访谈、微信、电子邮件等方式直接建立用户挽回通道，以挽回这部分高价值用户
        - 322、323和332：有潜力的普通群体
            - 最近刚完成购买，需要提升的是购买频次及购买金额
            - 可通过交叉销售、个性化推荐、向上销售、组合优惠券、打包商品销售等策略，提升其单次购买的订单金额及促进其重复购买

5.6 案例应用

- 针对上述得到的分析结论，会员部门采取了以下措施
    - 分别针3类群体，按照公司实际运营需求和当前目标，制定了不同的群体落地的排期
    - 录入数据库的RFM得分数据已经应用到其他数据模型中，成为建模输入的关键维度特征之一

5.7 案例注意点

- 不同品类、行业对于RFM的依赖度是有差异的，即使是一个公司在不同的发展阶段和周期下，3个维度的优先级上也会有调整
    - 大家电等消费周期较长的行业，R和M会更重要一些
    - 快消等消费周期短且快的行业，更看重R和F
    - 具体要根据当前运营需求与业务部门沟通
- 对R、F、M区间的划分是一个离散化的过程，具体需要划分为几个区间需要与业务方确认
    - 本案例划分为3个区间，结果对于业务分析而言有些多，意味着业务方需要制定十几套甚至更多的策略
    - 如果业务方要求简化，也可以划分为2个区间，这样出来的分组数最多有8组，策略制定更加简单
    - 具体是划分为2个还是3个，取决于当前业务方有多少资源可以投入到这个事情中来。
- R、F、M的权重打分
    - 除了案例中提到的建模方式外，结合业务经验的专家打分法也是常用的思路，这时推荐结合AHP层次分析法打分，这样出来的权重结果更加科学、严谨。
    - 虽然订单数据库中的数据质量相对较高，但可能由于数据采集、数据库同步、ETL、查询、误操作等问题，还是会导致NA值的出现，而NA值的处理非常重要。
    - R、F、M三个维度的处理（包括计算、离散化、组合、转换）之前都需要注意其数据类型和格式，尤其是有关时间项的转换操作应提前完成