

第五章 会员数据化运营

- 1、会员数据化运营概述
 - 辅助与客户关系管理CRM
 - 解决问题
 - 会员生命周期状态
 - 会员核心诉求
 - 会员转化习惯和路径
 - 会员价值
 - 如何扩大市场覆盖
 - 如何维系老会员
 - 何时采取何种措施，针对哪部分会员做哪些运营活动
 - 如何制定会员管理策略
 - 行为管理
 - 体验管理
 - 增值服务
 - 信息管理
 - 营销管理
 - 客户关怀
- 2、会员数据化运营关键指标
 - 会员整体指标
 - 注册会员数
 - 激活会员数
 - 会员激活率
 - 注册会员中完成激活的比例
 - 购买会员数
 - 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
 - 会员营销指标
 - 可营销会员数
 - 营销费用
 - 营销媒介费用
 - 优惠券费用
 - 积分兑换费用
 - 营销收入
 - 用券会员/金额/订单比例
 - 用券下单的会员数/总下单会员数

- 用券下单的金额/总下单金额
 - 用券下单量/总下单量
- 营销费率
- 每注册/订单/会员收入
- 每注册/订单/会员成本
- 会员活跃度指标
 - 整体会员活跃度
 - 每日/每周/每月活跃用户量
- 会员价值度指标
 - 会员价值分群
 - 复购率
 - 消费频次
 - 最近一次购买时间
 - 最近一次购买金额
- 会员终生价值指标
 - 会员生命周期价值/订单量/平均订单量
 - 会员生命周期转化率
 - 会员生命周期剩余价值
- 会员异动指标
 - 会员流失率
 - 会员流失率的数值
 - 会员流失率的走向
 - 会员异动比
 - 新增会员/流失会员
- 3、会员数据化运营应用场景
 - 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；
 - ❑ 通过特定方法将普通用户拓展企业会员，并提高新会员留存率；
 - ❑ 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，作为黄牛或VIP客户识别的参考；
 - ❑ 挖掘会员传播关系，找到口碑传播效应的关键节点。
 - 会员关怀

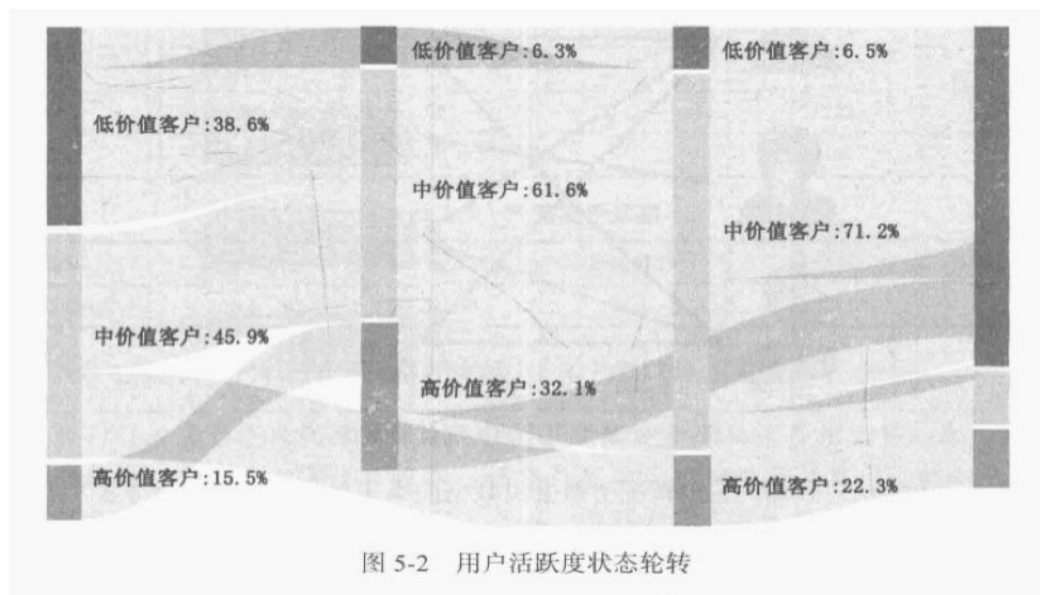
数据化运营应用于会员关怀主要体现在以下几个方面：

 - ❑ 为预警事件设置阈值，自动触发应急处理机制；
 - ❑ 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，更好的划分会员群体属性并挖掘群体性特征；
- 4、会员数据化运营分析模型

- 会员细分模型
 - 定义
 - 将整体会员划分为不同的细分群体或类别
 - 模型
 - 基于属性的方法
 - ABC分类法（主次因素分析法）
 - 根据事物的主要特征做分类排列
 - 聚类法
- 会员价值度模型
 - 定义
 - 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
 - 模型
 - RFM
 - 根据会员最近一次购买时间R，购买频率F和购买金额M三个维度评估
 - 基于一个固定时间点来分析
- 会员活跃度模型
 - 定义
 - 评估用户的活跃度情况，是会员状态的基本模型之一
 - 模型
 - RFE
 - 维度
 - 最近一次访问时间R
 - 访问频率F
 - 页面互动度E
- 会员流失预测模型
- 会员特征分析模型
- 营销响应预测模型
 - 定义
 - 找到可能响应活动的会员特征以及整体响应的用户比例，数量和可能带来的销售额
 - 算法
 - 逻辑回归
 - 支持向量机
 - 随机森林
- 5、会员数据化运营分析小技巧
 - 使用留存分析新用户质量
 - 留存用户
 - 新用户在经过一定时间后，还具有访问登录使用等行为
 - 指标
 - 日留存

- 每天相对于第一个天的新增用户中，仍有登录的用户数
 - 次日留存率
 - 第X日留存率
 - 周留存
 - 每周相对于第一个周的新增用户中，仍有登录的用户数
 - 月留存
 - 每月相对于第一个月的新增用户中，仍有登录的用户数
- 需要注意的问题
 - 区别应用不同留存周期
 - 日留存用来做短期结果、周留存用来看中期效果、月留存用来看长期效果
 - 注意观察和分析衰减率
- 使用AARRR做APP用户生命周期分析
 - 概述
 - AARRR是Acquisition、Activation、Retention、Revenue、Refer 五个单词的缩写，分别对应用户生命周期中的5个环节:获取用户、提高活跃度、提高留存率、获取收入、自传播。
 - 获取用户
 - 用户数量
 - 用户质量
 - 提高活跃度
 - 提高留存率
 - 获取收入
 - 付费用户数、比例
 - ARPU 平均用户收入
 - 新增付费用户
 - 付费转化周期
 - 重复消费比例
 - 消费金额在特定比较金额以上的客户
 - 消费分级
 - 自传播
 - 概述
 - 如果用户使用产品时体验良好，那么可能会产生传播效应，把产品推荐给其他朋友、亲戚等使用。
 - 分类
 - 产品引导传播
 - 用户主动传播
 - 标记方法
 - APP行为检测
 - APP交易检测
 - APP外部检测

- 借助动态数据流失关注会员状态的轮转



- 使用协同过滤算法为新会员推送个性信息
- 6、会员数据化运营分析的“大实话”
 - 虽然企业“不差钱”，但必要做会员精准营销
 - 用户满意度取决于期望和给以的匹配程度
 - 用户不购买不等于流失
 - 来自调研问卷的用户信息未必可信
 - 不要盲目相信二八法则
- 7、案例 基于RFM的用户价值度分析
 - 字符串合并
 - 使用 + 组合字符串
 - 使用%占位符做字符串
 - 使用.join方法将多个可抵达的对象合并
 - 使用.format做占位符将多个字符串合并
 - pandas自带字符串组合方法cat

小插曲 pandas.cat ¶

```
# cat(self, others=None, sep=None, na_rep=None)
'''
:param other: 要合并的另一个对象，如果为空则合并左侧的组合
:param sep: 合并的分隔符，默认为空
:param na_rep: 如果遇到NA缺失值时如何处理，默认忽略
'''
print '{: ^60}'.format('示例一')
print pd.Series(['a', 'b', 'c']).str.cat(sep='--')
print '{: ^60}'.format('示例二')
print pd.Series(['a', 'b', 'c']).str.cat(['A', 'B', 'C'], sep='!=')
# 更多信息可以help(pd.Series.cat)
```

```
-----示例一-----
a--b--c
-----示例二-----
0    a!=A
1    b!=B
2    c!=C
dtype: object
```

- 8、案例 基于AdaBoost的营销响应预测

- 背景

- 业务部门希望通过建好的模型周期性自动执行
 - 业务部门希望基于现有的辅助决策平台将会员数据筛选和查看功能跟该模型结合起来应用

- 引用库

- 相关知识点

- 使用Numpy的hstack和vstack做矩阵合并

- 小插曲 Numpy.hstack/vstack

```
# h(v)stack(para)
"""
:param para: 需要合并的两个矩阵，格式必须为元组tuple
:return: numpy.ndarray
"""
b = a = np.arange(6).reshape(2,3)
print 'a/b shape:', b.shape
print 'hstack:', np.hstack((a,b)).shape # 左右合并
print 'concatenate(..,axis=1):', np.concatenate((a,b), axis=1).shape # hstack等价于concatenate(..,axis=1)
print 'vstack:', np.vstack((a,b)).shape # 上下合并
print 'concatenate(..,axis=0):', np.concatenate((a,b), axis=0).shape # vstack等价于concatenate(..,axis=0)

a/b shape: (2, 3)
hstack: (2, 6)
concatenate(..,axis=1): (2, 6)
vstack: (4, 3)
concatenate(..,axis=0): (4, 3)
```

- 使用集成算法做模型训练

- adaboost 一种流行的集成算法，核心思想是针对同一个训练集训练不同的分类器，然后把把这些弱分类器集合起来，构成一个强分类器

- sklearn提供的集成算法

- 使用平均方法集成算法

- 基本原理：构建一些小的模型器然后基于每个模型器的结果做均值计算得到最终结果
 - 典型代表：Bagging、Random Forset等

- 使用提升方法集成算法

- 基本原理：每个模型器按顺序参与模型评估，并试图降低组合模型器的偏差
 - 典型代表：Adaboost、Gradient Tree Boosting等

- 使用Pipeline构建组合评估器

- 作用：一个复合评估器，用来将多个具有上下逻辑的过程连接起来形成一个复合对象

- 参数：step

- 由名称和模型对象组成的元组的列表
 - 不同元组之间有明确的先后关系，并且最后一个元组一定是一个评估算法

