## 引言

本小组成员均未担任市场营销相关岗位，也未在消费品相关行业从业，因此在原始数据的获取上面临较大困难。因此，我们选择了Kaggle中的相关数据[[1]](#footnote-1)来进行分析。

数据中包含了对2240位客户进行观察的结果，有28个与营销数据相关的变量，这些变量提供了客户画像、购买的产品、活动成功与否、渠道表现等信息。数据中有AcceptedCmp 1-5五列数据，对于第i次活动的结果，可以使用第1至第i-1次的结果来进行预测，从而可以通过精准化提高营销效率。

我们将首先从老师上课讲授的RFM模型出发进行探索，探究RFM模型应用于预测该数据集第i次活动用户是否参与的效果。此后，将进一步探索通过使用新算法的优化和引入更多数据来改进模型效果[[2]](#footnote-2)。

## 一、模型

### 方法一：RFM-GPMEAN

RFM模型，即利用R（Recency：距离最近一次消费时间）、F（Frequency：消费频次）、M（Monetary：总消费金额）这三个变量来分析预测客户的反应。在这里，利用老师课堂上介绍的方法，将各变量排序后分成五组（数据总共只有两千条），因此所有变量就被分成125组。用每个分组成员之间是否接受的平均值，来预测下一次客户是否仍然会接受此类营销活动。代码如下：

def pred\_is\_accept(df, target):

    r = df[cst.H\_RECENCY].values

    m = cc.sum\_col(df, REG\_M)

    f = cc.sum\_col(df, REG\_F)

    gp = get\_gp\_id(r, m, f)

    gp\_df = pd.DataFrame(dict(gp=gp, target=target))

    # use groupby.transform is better

    gp\_dict = gp\_df['target'].groupby(gp\_df['gp']).\

        mean().to\_dict()

    pred = gp\_df['gp'].map(gp\_dict)

该方法简单、直接，但缺点也很显然，分组过少结果粗糙，过于局域性没有考虑整体上的联系。

### 方法二：RFM-LOGIT

还是基于RFM模型，输入：R、F、M三个变量值，利用statmodels包中的Logit回归。代码如下：

def pred\_is\_accept(df, target):

    r = df[cst.H\_RECENCY].values

    m = cc.sum\_col(df, REG\_M)

    f = cc.sum\_col(df, REG\_F)

    rfm\_df = pd.DataFrame(dict(r=r, m=m, f=f))

    x = sm.add\_constant(df\_zscore(rfm\_df))

    est = sm.Logit(target, x).fit()

    print(est.summary())

    pred = est.predict(x).values

    np.clip(pred, 0., 1., out=pred)

### 方法三：RFM-TREE

同样地，仍然输入R、F、M这三个变量值，用sklearn中的RandomForestClassifier来做预测。代码如下：

def pred\_is\_accept(df, target):

    r = df[cst.H\_RECENCY].values

    m = cc.sum\_col(df, REG\_M)

    f = cc.sum\_col(df, REG\_F)

    rfm\_df = pd.DataFrame(dict(r=r, m=m, f=f))

    x = rfm\_df.values

    tree = RandomForestClassifier(

        max\_samples=0.5, oob\_score=True, max\_depth=2,

        min\_samples\_split=10, min\_samples\_leaf=5,

        min\_impurity\_decrease=0.00001)

    est = tree.fit(x, target.astype(bool))

    pred = tree.predict\_proba(x)[:,1].copy()

### 方法四：F8-TREE

在方法三的基础上做更新，多增加了五个变量的输入：包括网站浏览频率，教育水平，家中子女数，还是用随机森林的方法。代码如下，注意大部分超参数的设置更为保守：

def pred\_is\_accept(df, target):

    data = dict(r=df[cst.H\_RECENCY].values)

    data['m'] = cc.sum\_col(df, REG\_M)

    data['f'] = cc.sum\_col(df, REG\_F)

    data['inc'] = df[cst.H\_INCOME].values

    data['yr'] = df['Year\_Birth']

    data['edu'] = df[cst.H\_EDUCATION].values

    data['vf'] = df['NumWebVisitsMonth'].values

    data['child'] = df['Kidhome'].values+df['Teenhome'].values

    rfm\_df = pd.DataFrame(data)

    x = rfm\_df.values

    tree = RandomForestClassifier(

        max\_samples=0.8, oob\_score=True, max\_depth=2,

        min\_samples\_split=15, min\_samples\_leaf=10,

        min\_impurity\_decrease=0.0001)

    est = tree.fit(x, target.astype(bool))

    pred = tree.predict\_proba(x)[:,1].copy()

## 三、实验结果

我们利用上节的四种模型，分别检验了从第二轮活动开始，每轮活动用户是否接受的预测效果。第二轮和第五轮的结果将被重点展示，因为第二轮的结果代表了在少量数据情况下的实验结果，而第五轮时数据更为丰富，此时可能更复杂的模型表现效果会更加理想。检验的统计指标，我们将采用老师课堂上讲述的以下两个指标：Top-decile Lift和Gini Coefficient。

第二轮活动的预测（附录表1），从Gini Coefficient的角度，和从5%分组Lift来看，rfm-tree模型的表现效果最好。笔者认为，此处有三点原因：一是，按我们此处选择的随机森林的设置，优化目标为Gini Coefficient，所以并不意外在此项统计指标上比简单的分组平均和优化均方误差的逻辑回归效果要好；二是，由于决策树类模型的数学性质，决定了其在掐尖方面的特有优势；三是，相比于f8-tree，该模型仅有三个输入变量，比六因子的模型鲁棒性更高，即使六因子模型在超参数的设置上已经比三因子的模型更为保守。

第五轮活动的预测（附录表2），从Gini Coefficient的角度，和从5%,、10%、20%、30%分组Lift来看，变量最多、模型最复杂的f8-tree模型的表现效果最好。八因子新加入的收入和家中孩子数量变量也具有显著的解释作用，而简单地RFM模型中并未包含该变量。不论各个模型，影响最大的属性均为m，即过去两年间的购买总额；r，上次购买时间的贡献均不显著，可能和商品种类多，消费频率不固定，r无比较意义有关。

## 四、营销建议

经过数据分析我们发现有较高收入且有小孩是比较重要的影响变量，而上次购买时间、年龄等变量影响不大。针对上述人群，我们有如下营销建议：

1. 瞄准目标客户：充分利用各种媒体广告，营造轻奢的购物环境，如财经类作家公众号、地铁广告、中高档住宅电梯广告。目标客户为更看重商品质量、安全和非高度价格敏感的中高端收入群体消费者，这要求商家1）注重产品质量，2）通过大容量包装降低成本，吸引需要家庭装的用户，3）用有特色、差异化的食品等吸引客户。

2. VIP客户维护：给会员进行分级，不同级别的会员缴纳的会费不一样因此也会享受到不同的服务。VIP会员在享受普通会员服务的基础上增添特殊服务，如：积分返券、高端齿科服务、生活服务、网购免邮。

3. 渠道开阔：和质量有保障的供货商建立长期合作关系，并给供应商提供原材料采购融资服务。

4. 精准营销：在购物环境中充分考虑幼童游玩区域，将高毛利的玩具零食等放置在商场显眼区域，广告投放时着重推送母婴、孩童用品。

## 附录：统计数据

表 1 第二轮活动预测的表现

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **rfm-gpmean** | | **rfm-logit** | | **rfm-tree** | | **f8-tree** | |
| **Gini** | 53.26% |  | 59.09% |  | **59.59%** |  | 54.09% |  |
| **pct(%)** | **pi\_p** | **lift** | **pi\_p** | **lift** | **pi\_p** | **lift** | **pi\_p** | **lift** |
| **5** | 5.36% | 400.00% | 5.36% | 400.00% | 6.25% | **466.67%** | 6.25% | 466.67% |
| **10** | 5.36% | **400.00%** | 5.36% | **400.00%** | 4.91% | 366.67% | 4.91% | 366.67% |
| **20** | 4.24% | 316.67% | 4.46% | **333.33%** | 4.46% | 333.33% | 4.02% | 300.00% |
| **30** | 3.42% | 255.56% | 3.57% | **266.67%** | 3.42% | 255.56% | 3.42% | 255.56% |
| **50** | 2.32% | 173.33% | 2.50% | 186.67% | 2.50% | 186.67% | 2.41% | 180.00% |

表 2 第五轮活动预测的表现

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **rfm-gpmean** | | **rfm-logit** | | **rfm-tree** | | **f8-tree** | |
| **Gini** | 70.39% |  | 77.35% |  | 76.04% |  | **82.29%** |  |
| **pct(%)** | **pi\_p** | **lift** | **pi\_p** | **lift** | **pi\_p** | **lift** | **pi\_p** | **lift** |
| **5** | 33.93% | 466.26% | 51.79% | 711.66% | 50.89% | 699.39% | 63.39% | **871.17%** |
| **10** | 33.48% | 460.12% | 42.86% | 588.96% | 44.20% | 607.36% | 47.77% | **656.44%** |
| **20** | 29.69% | 407.98% | 30.80% | 423.31% | 31.25% | 429.45% | 34.38% | **472.39%** |
| **30** | 21.88% | 300.61% | 22.62% | 310.84% | 22.02% | 302.66% | 23.96% | **329.24%** |
| **50** | 14.02% | 192.64% | 14.55% | 200.00% | 14.38% | 197.55% | 14.55% | **200.00%** |

1. https://www.kaggle.com/jackdaoud/marketing-data [↑](#footnote-ref-1)
2. 代码及详细结果见https://github.com/victor-zou/hw-bdm [↑](#footnote-ref-2)