模型与营销方案

团队名：赛队 8d8fd

前 16 强获优胜奖及奖金

目录：

1.初赛模型与方案介绍

1.1.赛题理解

本次大赛旨在针对客户购买各类理财产品存单概率进行预测，属于一个 user-prod-label 的类似于 ctr 预测的问题，只不过不是预测点击率 ctr 而是预测购买概率。赛事提供了 user 端及 prod 端各自的信息，以及 user-prod 的 APP 点击信息和交易流水表，除此之外还有 user-user 的借贷交易流水表，希望选手通过充分挖掘各个表里的信息更好的完成赛事任务。

我们团队通过初赛的数据分析与模型训练发现，该 ctr 预估问题的冷启动比例在 user 和 prod 两方面呈现极端现象。user 端的冷启动比例仅有 2.8%左右，但是 prod 端的冷启动比例则达到了 60%左右，甚至如果只看 prod 的种类冷启动的话，线上测试集里 70%+的 prod 都没有在训练集里出现过。所以如何缓解 prod 端冷启动问题，则是该问题提分的关键。

基于此，我们团队最终确定的冷启动解决方案是：①prod 端的特征尽可能只留下强特，以 user 端的特征构造为主，且尽量避免通过训练集来构造 prod 特征而采用其它表；②尽可能构造 user-prod 交互的特征，以在 prod 特征中引入 user 信息带来的偏差缓解 prod 冷启动问题。

这样一种方案也很符合该业务的直观理解，即生活中买不买理财产品主要还是 user 方面的信息影响更大一些，产品与产品尤其是同类产品之间的差异很小。

1.2.数据初步探索

拿到赛题数据后先对各表情况进行了一个初步的情况分析，为后续特征工程指引方向。以下则是我们团队分别针对 A、B 榜做过的数据探索，大部分在 A 榜里探索出来的东西在 B 榜里也基本一致，所以 A 榜的探索更为繁琐。

1.2.1. A 榜数据：

赛事提供的数据 a、b、c 三个表包含训练集 train 和线上测试集 test，d、e、f 是 user 端单独的表信息，g、h、i、j、k、l、m 是 prod 端单独的表信息，n、o、p、q、r 是 user-prod 的交互表信息，s 表是 user-user 的借贷交易表信息。

由于需要预测的 prod 包含 A、B、C、D 四个大类，所以所给的 ghijklm 七个表是不同的类分开储存的，这样如果直接 merge 虽然可以凸显不同大类之间的差异但特征本身的含义很不统一可能会影响后续模型的学习程度。所以我们团队将 ghijklm 七个表按列特征进行了统一，ghij 四个表合并为 prod1_df 的一个表，而 klm 合并为 prod2_df 的一个表。最终得到的统一列特征包含：

prod1_df （0 计价类型、1 周期类型、2 产品模式、3 产品风险等级、4 是否滚存型产品、5 是否允许变更分红方式、6 份额冻结的比率、7 产品品种、8 产品模式 2、9 收益计息基数、10 预期最低收益率、11 预期最高收益率、12 产品面值、13 发行价格、14 收益率、15 持有天数、16 数据日期）

prod2_df （0 募集方式、1 管理方式、2 业务模式、3 收益特点、4 产品期限、5 产品投资模式、6 预期收益率、7 最高收益率、8 最低收益率、9 产品展示等级、10 日期）

除此之外，为了区别每个 prod 属于 ABCD 哪个大类，引入了与’a2’特征含义相同的’class’列。

同理于 n、o、p、q 四个表，我们也把它们合并为一个表，并引入一个’class’列：

nopq_df （0 交易流水号、1 业务代码、2 渠道标识、3 净值、4 申请金额、5 折扣率、6 交易状态、7 资金状态、8 总金额、9 超额管理费金额、10 费率、11 交易日期）

完成数据的初步统一与合并后，进行了一些信息的统计：train 里 2386133 条用于训练（260814 个客户 id，90 个产品 id）正样本只有 25434 条，test 里 567362 条需要预测（112645 个客户 id，51 个产品 id）。各表数据条数：d 表(264055 条)、e 表(316116 条)、f 表(1038596 条)、prod1 表（9613 条）、prod2 表（26005 条）、nopq 表（2052078 条）、r 表（777075 条）、s 表（6507904 条）

与冷启动相关的信息统计：

test 里有 311774 条数据、39 个 prod 不在 train 表；test 里有 16319 条数据、3241 个 user 不在 train 表。冷启动条数比例占比（user=2.88%，prod=54.95%），冷启动个数比例占比（user=2.88%，prod=76.47%）。

train 里有 1216337 条数据、135015 个 user 不在 e 表；test 里有 271346 条数据、54136 个 user 不在 e 表。在 train 和 test 里的条数和个数缺失程度均在 50%左右。

d、f 表里基本包含 train+test 里所有 user，但 f 表缺失很严重，仅有 41.2%左右有非 0 记录，一个月内动账的账户更少仅占 5.92%，说明很多账户资金不太活跃。

prod1 表里包含 train+test 所有 prod 信息，但 train 里有 487715 条数据、6 个 prod 不在 prod2 表；test 里有 114703 条数据、5 个 prod 不在 prod2 表。可见 prod2 表有一定程度缺失，但比例也只有 10-20%左右。

train 里有 1823784 条数据、202222 个 user 不在 nopq 表；test 里有 416087 条数据、82804 个 user 不在 nopq 表。train 里有 118 条数据、1 个 prod 不在 nopq 表；test 里有 60280 条数据、10 个 prod 不在 nopq 表。可见 train+test 里绝大部分 user 没有任何产品交易流水信息，因为绝大部分 label 都是 0 符合常理，而 prod 则几乎绝大部分都在 nopq 里，这种现象也在 r 表里有所反应，所以通过 nopq+r 表可以在一定程度上缓解 prod 端的冷启动问题。

train 里有 1101552 条数据 user 不在 s 表，test 里有 251987 条数据 user 不在 s 表。可见 s 表的情况与 e 表类似。

分类别统计 prod 冷启动信息：

A 类产品里在 train 里的 67 个（104w 个里 1w 个正样本）都不在 test（37 种，共 26w 条）里，B 类产品有 2 种在 train(34w 个里 7000 个正样本)和 test(5w)里一样【SSTJMZKF001，SSTJMZKF002】，C 类产品在 train(27w/52w 里 560 个正样本)里有 15 种和 test(13w)里 7 种交叉了 7 种【DECD21090102，DECD21090103，DECD21090104，DECD21090105，DECD21090106，DECD21090107，DECD21090108】，D 类产品在 train(27w/49w 个里 6800 个正样本)里有 6 种和 test(11w)里 5 种，交叉了 3 种【90318011，91318017，HD1D001】

BCD 共交叉了 train：34+27+27 万条，test:5+13+11 万条，可见 A 类产品不论是在 train 还是 test 里均占了一半左右，但很有意思的一个现象是 A 类产品在 train 里出现的 67 种居然都没在 test 里出现。同时，C 类产品购买率最低，达到千分之一左右。

1.2.2.最终数据情况：

B 榜同样将 ghij 统一为 prod1_df 表，klm 统一为 prod2_df 表，nopq 统一为 nopq_df 表，各表数据分布与 A 榜差别不大，不影响后续特征构造思路。

test 里有 353206 条数据、31 个 prod 不在 train 表（共 587805 条、43 种）；test 里有 39641 条数据、7864 个 user 不在 train 表（共 587805 条、114005 种）。冷启动条数比例占比（user=6.74%，prod=60.09%），冷启动个数比例占比（user=6.90%，prod=72.09%）。

A 榜 test 里有 ABCD 四类产品分别（条）: 267051 134147 51461 114703
A 榜 test 里有 ABCD 四类产品分别（个）: 37 7 2 5
B 榜 test 里有 ABCD 四类产品分别（条）: 232455 183505 46174 124951
B 榜 test 里有 ABCD 四类产品分别（个）: 23 10 2 8

为了方便后续 user2vec 或者用 label 统计热度信息，分日期统计了以下缺失情况：
20210801 以前 label=1 数据在 20210801 里：prod 缺失 22/26，user 缺失 229601/237493；
20210901 以前 label=1 数据在 20210901 里：prod 缺失 43/49，user 缺失 245344/258235；
20211001 以前 label=1 数据在 20211001 里：prod 缺失 42/51，user 缺失 103821/112645；
20211201 以前 label=1 数据在 20211201 里：prod 缺失 31/43，user 缺失 104639/114005；

1.3.特征工程

基于 1.2 节的数据分析情况，我们团队最终采用了以下特征构造方式，主要分为 user、prod 基础统计特征 + 热度统计 + user-prod 交互特征 + user2vec 特征。由于 200w+的数据里 user 种数有 20w+个，而 prod 种数仅有 100+，故 user 聚合出来的时间序列会比 prod 短很多，这也与实际业务吻合。实际业务场景下，用户购买理财产品的时间周期一般以年月来计，所以基本上除了某些特殊特征以外，对 user 均统计历史所有信息；而 prod 则由于理财产品本身随时间波动的情况，只统计最近一个月的情况。

值得注意的是，prod1_df、prod2_df 表经过多次实验发现直接 merge 上去的效果并不好 ，这是由于两个表里的收益率特征十分零散、并不统一导致的噪音。后面缺失值处理里，会详细介绍我们团队是如何处理这个问题的，只有这个问题处理了，后面的交互特征才能得到较好的效果，否则噪音的交互只会带来更多的噪音。

1.3.1.各表特征构造

具体来说，对各个表做了的特征处理如下（共 231 个特征）：

d 表 3 个：性别、等级、年龄。

f 表 13 个：账户创建日期，加 5 列时点余额及其 sum，还有前述 6 项近一个月动账。

s 表 8 个：借和贷金额分别对应的 mean、sum、count 及对应交易代码的 mean。

train 对 label 统计 6 个 ：user 和 prod 对应 label 的 mean、sum、count 分别对应购买率、购买数、浏览数（热度特征）。

e 表 1 个：user 风险等级近一个月的 mean。

prod1+prod2 表 2 个：仅留下持有天数、收益率两列强特。

r 表 2 个：user 历史点击过 prod 的种数和次数。

n、o、p、q 表 38 个：四个表分开 merge 凸显 user 对四种类别的交易流水情况，分别统计了 user 历史全部和近一个月交易过的 prod 的次数和种数及对应申请金额的 mean、sum。除此之外还将合并后的 nopq_df 表进行特征补充，统计 prod 近一个月交易对应的折扣率与净值作为其新的补充基础特征，然后统计 user 历史全部交易对应折扣率、净值的 min、max，判定需要预测的 prod 净值和折扣率是否在该[min ,max]区间内（交互特征）。

user2vec 特征 16 个：对 label=1 的 train 表进行 16 维度的 user2vec 得到的 embedding 特征。

train 表好评差评 user-prod 交互信息统计 49 个：除了 user 历史购买过的 prod 种数、prod 历史全部和近一个月被购买的 user 的种数 3 个基础特征外，其余 46 个全是与下面两个 nopq、

r 表类似的交互特征。

nopq 表 user-prod 交互信息统计 46 个。

r 表 user-prod 交互信息统计 46 个。

交互特征的交互思路主要分为两个：①统计 user 端的历史某一 prod 属性的 mean，然后与需要预测的 prod 的该属性的值做差的绝对值，这是 user 对 prod，反之 prod-user 也一样；②统计 user 端的历史某一 prod 属性的 min、max，然后判定需要预测的 prod 的该属性的值是否在该[min, max]区间内，这是 user 对 prod，反之 prod-user 也一样。

以此思路做了前述净值、折扣率的交互特征，以及后面 3 组分别对应历史购买过、交易过、点击过对应的各 46 个交互特征。包括对 prod 端的（持有天数、收益率、产品类型、热度特征）6 个基础特征，user 端的（性别、等级、年龄、风险等级、热度特征、账户总资金及动账、借和贷的交易代码的 mean、借和贷的交易金额的 mean、sum、count）17 个基础特征的交互特征构造。

1.3.2.缺失值处理

前面说了，交互特征的基础是基础特征的纯度比较高，不能包含太多的噪音。通过不断的实验，最终我们团队只留下了 prod1、prod2 两个表里的持有天数、收益率两个强特，以尽可能减少 prod 端的不重要信息干扰，本身 prod 端的冷启动比例就高，所以特征就得尽可能强相关一点。而对收益率特征的处理，由于两个表里存在（预期最低收益率、预期最高收益率、收益率、预期收益率、最高收益率、最低收益率）六列特征与其相关，且本身的收益率该列缺失度极其严重，所以需要做一个统一处理。

具体操作就是，通过观察表里内容发现带最低、最高字眼的特征列基本与原收益率列差一个 100 倍数值的关系，所以通过最低和最高收益率的 100 倍的 mean 来进行收益率列缺失值的填补。同时发现当最低、最高收益率为 0.0011 时，与原收益率不符 100 倍关系，故不能用 0.0011 对应行进行上述方式的填补。通过该方式对 prod1 表进行收益率列的填补后，发现 prod2 表里的预期收益率很完整不需要通过最低最高的 mean 来填补，且可以对 prod1 表里的收益率缺失值进行补充，故进一步将 prod2 表里的预期收益率列作为 prod1 收益率特征的补充规则。

至此，就只使用 prod1_df 表作为 prod 端单独信息的表就行了，且收益率特征十分完整，而且训练集和测试集里每个 prod 都在 prod1 表里，可以说在一定程度上解决了 prod 端的冷启动问题。

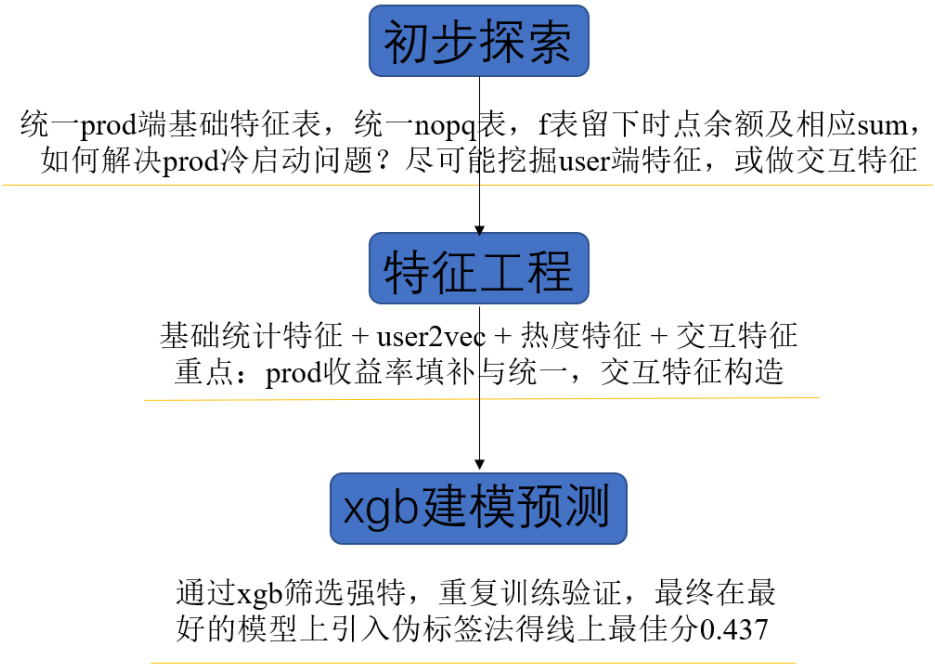
除了做了收益率特征的统一填补操作，针对金额和热度特征也需要做一个填充，因为后续做交互特征时需要使用这些特征的值做差或者比较，如果是 nan 则 bool 型均为 false、差值均为 nan 会引入噪音。

对该类特征的缺失值填充则比较简单，就是直接填 0 即可，因为当 user 的金额为 nan 时，基本上是在 f 或者 s 表里没有出现过的 user，这类 user 可以认为是资金不活跃的用户可以用 0 来代表。同理于 user 或者 prod 的热度特征，当热度为 nan 时，表示从来没有过购买或被购买记录，认为是冷启动的 user 或 prod 即可，热度为 0 故填 0。即 1.3.1 里 f、s、train 对 label 的统计得到的 13+8+6 个特征全部填 0 即可。

除上述特征以外，其余特征均不做 nan 特殊处理，放入树模型种自动处理缺失值即可。

1.4.建模流程与训练评估

整体建模流程如下图 1 所示：



图(1):整体建模流程图

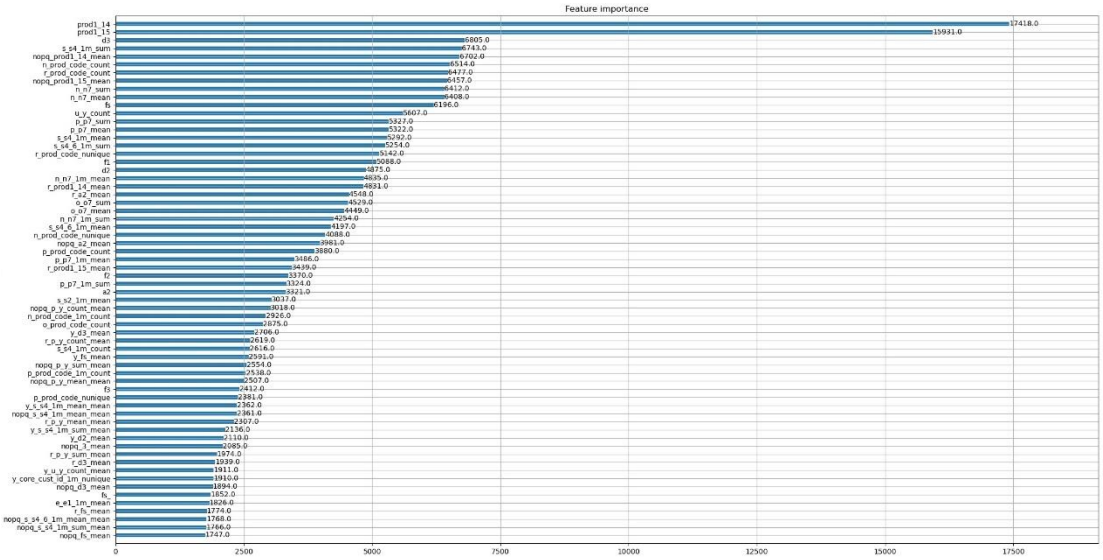
将 1.3 里构造出来的 231 个特征录入树模型进行 ctr 的预测，尝试了 lgb、ctb、xgb 三种树模型，以及 tabnet 新型神经网络模型，最终发现在我们的特征上 xgb 的表现最好也最为稳定，且模型融合基本没有提分，故最终选择用 xgb 单模进行训练与评估。

从 1.2.1 里分产品类别的 train+test 信息统计里可以发现，不同产品之间的差别非常大，但是同一类产品之间的差别却很小。因为在都是 A 类产品的数据里，虽然 train 里 67 个 prod 均未在 test 里出现属于冷启动产品，但是在我们团队将 ABCD 类产品数据分开来进行训练时发现 A 类产品的预测并不差，说明了同一类产品之间的差异是可以用 prod1 或 prod2 表里某一列强特来表征的。同时，ABCD 四类产品各自缺失的特征列差异极大，A 类产品（持有天数、收益率为主）、B 类（持有天数、收益率均缺失严重）、CD 类（持有天数一个全 0 一个全 1，收益率都很完整），且 C 类产品前面说了购买率很低仅有千分之一左右。所以通过统一的 prod1 表的特征列的值就可以很明显的区分开 ABCD 各类理财产品，也就没有必要将四类产品的数据分开来进行训练了。

前述交互特征最终是基于 prod 端的 6 个基础特征和 user 端的 17 个基础特征，但前期并没有 23 个这么多，之所以加到了这么多是因为后期实验发现该交互特征十分可靠，从最初的五六个不断加到 23 个，越加线上得分越高，说明了我们方案的稳定性与可靠性。且这些特征均有其业务上对应的可解释性与意义，表征了 user-prod 历史行为与当前需要预测的行为之间的偏差。

在该 231 个特征获得线上最高分私榜 0.433 的情况下，我们团队还引入了伪标签法进行负样本的数据扩增。具体来说就是通过该模型对 B 榜数据进行预测，将前 31w 个得分最低的样本全部作为负样本加入训练集重新进行一次训练，最终得到一个新的预测结果。而通过该方法，我们在线上也获得了 4 个 k 的提升，得到最终 0.437 的 B 榜成绩。

带伪标签法的 xgb 单模得到的最终 plot_importance 图如下图 2 所示，表征各特征重要程度。



图(2): xgb 对应最佳模型 plot_importance 图

前面的 prod1_14 则为前面处理得到的统一收益率强特，prod1_15 为持有天数强特，d3 为年龄，s_s4_1m_sum 为借方近一个月借出总金额，nopq_prod1_14_mean 为构造的关于收益率强特的交互特征是 user 历史的 mean 与原始收益率差的绝对值，n_prod_code_count 为 user 历史交易过 A 类产品的次数，r_prod_code_count 为 user 历史点击过的 prod 的次数，nopq_prod1_15_mean 为构造的关于持有天数强特的交互特征是 user 历史的 mean 与原始持有天数差的绝对值，同理于下面其它特征……

可见通过缺失值填补得到的收益率强特，以及基于较纯基础特征构造出来的交互特征对最终模型影响很大，起到了关键性的提分作用，且该交互特征越加得分越高，说明了模型的稳定性与合理性。

1.5.方案创新点

①通过将 ghij 统一为 prod1 表，klm 统一为 prod2 表，nopq 四个表统一为一个 nopq_df 表，实现了产品端强特的统一与凸显，减少不相关特征的干扰。

②在将①里得到 nopq 统一表里提取出 prod 随时间变化的折扣率、净值特征；通过 prod1 里最高最低收益率来填补收益率特征，并进一步将 prod2 的预期收益率对 prod1 里的收益率特征进行填补，构造了 prod 端的统一完整的强特“收益率”，减少了 prod 端基础特征的噪音，为后续特征交互提供坚实的基础。

③在基础特征纯度较高的情况下，进行各特征的交互，衍生了一系列 user-prod 交互特征，更好的凸显各特征对用户最终决策的影响。

④通过 user2vec 表征客户历史购买信息所包含的特征，将 embedding 录入树模型，可以在一定程度上体现 user 与 user 之间的差别，且不能用该 embedding 做进一步交互，仅用于一定程度的用户画像即可。

1.6.展望

①由于时间关系，仅尝试到了 23 个较纯基础特征构造出来的交互特征，后续可以考虑如何提高其它基础特征的纯度并用同样方式进行交互，如 n_n7_sum、n_n7_mean 用户对 A 类产品历史交易金额的 sum、mean 等特征。

②可以考虑验证以下后处理方式有没有用：对 test 里未在 nopq+r 表里出现过的 user 全置为 0，因为业务上讲没有任何交易流水和点击信息的用户大概率是对理财并不关心的 user。

③由于时间关系仅对 prod 端的收益率进行了深度信息挖掘，未对 user 端的账户资金情况进行业务联系的挖掘，所以如果能进一步在我们团队方案基础上加入对 user 端账户资金流水的业务交互或许能再获得一个较大的提升。例如，单独对 f 表看近一个月动账的 user 很少，但如果将 f 表本身的五项资金，与 s 表里的借贷业务金额或 nopq 表里的申请金额进行挂钩会不会使得动账信息更丰富？

2. 营销解决方案

通过前面的建模训练与评估，可以发现，对结果影响最大的就是理财产品 prod 对应的收益率和持有天数，还有客户 user 的年龄，其次就是其它的一些诸如 user 交易次数、点击次数、账户总资产余额等特征。而基于强特收益率、持有天数衍生出来的特征，也依然对模型的预测结果产生着巨大的影响，说明我们团队特征衍生方式的合理性。

基于该数据训练出来的模型可以计算得到对应客户可能购买对应理财产品的概率分数，分数越高则模型认为该 user 购买该 prod 的可能性越高。在面对某产品所需要营销的对象群体时，可以把这些 user 都进行前述建模预测得到各自对应响应分数，将该分数进行排序分组，为了尽可能节省营销成本且实现营销收益，使得 ROI>0，对不同分组内对应的 user 群体采用不同的营销方案。

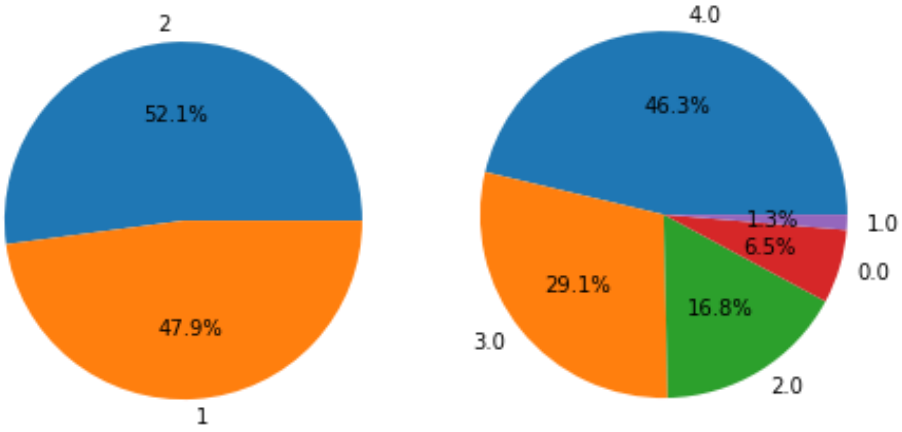
例如假设我们的营销方案有：1.电话营销，2. IVR 等智能语音手段营销，3.短信方式营销。则设定两个阈值 m、n，对分数在[0，n]内群体采用短信营销，[n，m]群体采用智能语音营销，[m，1]群体采用电话营销。

除了针对单个产品对应的这三种营销方案以外，还可以针对客户 user 端进行个性化的产品推荐，类似于推荐系统里的用户画像。通过用户历史交易、点击以及动账等信息对该特定用户进行现有每个理财产品的购买率预测，通过前面三种方式或者相应 APP，直接推荐得分最高的若干产品给到该用户。或者统计 user 历史交易、点击或购买过的产品类别，然后给推荐 user 经常浏览的那种类别的产品。

进一步，可以根据模型的可解释性进行探索，分析模型变量重要性。例如对重要变量进行分箱，计算客户对应该变量所处的箱，而得知该用户最喜欢什么样的产品，进而对客户进行相应类别产品的诱导营销。例如前面提到，产品收益率、持有天数两个特征是强特，则根据 user 历史交易、点击或者购买过的产品对应两个特征的数值在哪个区间内，然后给他推荐尽可能靠近区间中点的对应产品，其实这也就是前面交互特征衍生构造的思想。

例如，对经过 prepare_data.py 后用于训练的数据集进行 groupby，针对有过历史最多记录的 prod_code=91318017 产品，在 6079 次购买记录里：

性别和客户等级分布如图 3 所示：

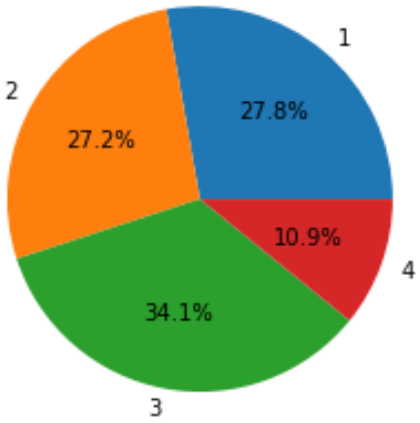


图(3): 购买过 91318017 产品的客户性别和客户等级分布图

发现性别对最终是否购买的影响并不大，但是客户等级有一定影响，等级越高越容易购买。

进而分析年龄的影响，对该产品购买过的用户年龄进行分箱，如图 4 所示：

（箱 1：年龄 18-35，箱 2：年龄 35-50，箱 3：年龄 50-70，箱 4：年龄 70 以上）



图(4): 购买过 91318017 产品的客户年龄分布图

可以发现 70 岁以上高龄人很少购买该产品，主要集中在有一定经济能力的 18-70 岁人群之间。

所以，根据上述分析和策略，类似的可以针对特定的产品、特定的用户进行具体的分析，以制定更为切实可靠的营销方案，达到更个性化更好的营销效果。