队伍简介

队伍名“诗人都藏在水底”，三位队员分别是来自北大的黄伟鹏（wepon）和辛超（charles），来自中科大的陈靖（云泛天音）。

赛题回顾

本赛题提供用户在2016年1月1日至2016年6月30日之间真实线上线下消费行为，预测用户在2016年7月领取优惠券后15天以内是否核销。评测指标采用AUC，先对每个优惠券单独计算核销预测的AUC值，再对所有优惠券的AUC值求平均作为最终的评价标准。

解决方案概述

本赛题提供了用户线下消费和优惠券领取核销行为的纪录表，用户线上点击/消费和优惠券领取核销行为的纪录表，记录的时间区间是2016.01.01至2016.06.30,需要预测的是2016年7月份用户领取优惠券后是否核销。根据这两份数据表，我们首先对数据集进行划分，然后提取了用户相关的特征、商家相关的特征，优惠券相关的特征，用户与商家之间的交互特征，以及利用本赛题的leakage得到的其它特征（这部分特征在实际业务中是不可能获取到的）。最后训练了XGBoost，GBDT，RandomForest进行模型融合。

数据集划分

可以采用滑窗的方法得到多份训练数据集，特征区间越小，得到的训练数据集越多。以下是一种划分方式：

	预测区间（提取label）	特征区间（提取feature）
测试集	20160701~20160731	20160315~20160630
训练集1	20160515~20160615	20160201~20160514
训练集2	20160414~20160514	20160101~20160413

划取多份训练集，一方面可以增加训练样本，另一方面可以做交叉验证实验，方便调参。

特征工程

赛题提供了online和offline两份数据集，online数据集可以提取到与用户相关的特征，offline数据集可以提取到更加丰富的特征：用户相关的特征，商家相关的特征，优惠券相关的特征，用户-商家交互特征。

另外需要指出的是，赛题提供的预测集中，包含了同一个用户在整个7月份里的优惠券领取情况，这实际上是一种leakage，比如存在这种情况：某一个用户在7月10日领取了某优惠券，然后在7月12日和7月15日又领取了相同的优惠券，那么7月10日领取的优惠券被核销的可能性就很大了。我们在做特征工程时也注意到了这一点，提取了一些相关的特征。加入这部分特征后，AUC提升了10个百分点，相信大

多数队伍都利用了这一leakage，但这些特征在实际业务中是无法获取到的。

以下简要地说明各部分特征：

- 用户线下相关的特征
    - 用户领取优惠券次数
    - 用户获得优惠券但没有消费的次数
    - 用户获得优惠券并核销次数
    - 用户领取优惠券后进行核销率
    - 用户满0~50/50~200/200~500 减的优惠券核销率
    - 用户核销满0~50/50~200/200~500减的优惠券占所有核销优惠券的比重
    - 用户核销优惠券的平均/最低/最高消费折率
    - 用户核销过优惠券的不同商家数量，及其占所有不同商家的比重
    - 用户核销过的不同优惠券数量，及其占所有不同优惠券的比重
    - 用户平均核销每个商家多少张优惠券
    - 用户核销优惠券中的平均/最大/最小用户-商家距离
- 用户线上相关的特征
    - 用户线上操作次数
    - 用户线上点击率
    - 用户线上购买率
    - 用户线上领取率
    - 用户线上不消费次数
    - 用户线上优惠券核销次数
    - 用户线上优惠券核销率
    - 用户线下不消费次数占线上线下总的不消费次数的比重
    - 用户线下的优惠券核销次数占线上线下总的优惠券核销次数的比重
    - 用户线下领取的记录数量占总的记录数量的比重
- 商家相关的特征
    - 商家优惠券被领取次数
    - 商家优惠券被领取后不核销次数
    - 商家优惠券被领取后核销次数
    - 商家优惠券被领取后核销率
    - 商家优惠券核销的平均/最小/最大消费折率
    - 核销商家优惠券的不同用户数量，及其占领取不同的用户比重
    - 商家优惠券平均每个用户核销多少张
    - 商家被核销过的不同优惠券数量
    - 商家被核销过的不同优惠券数量占所有领取过的不同优惠券数量的比重
    - 商家平均每种优惠券核销多少张
    - 商家被核销优惠券的平均时间率
    - 商家被核销优惠券中的平均/最小/最大用户-商家距离
- 用户-商家交互特征
    - 用户领取商家的优惠券次数

    - 用户领取商家的优惠券后不核销次数
    - 用户领取商家的优惠券后核销次数
    - 用户领取商家的优惠券后核销率
    - 用户对每个商家的不核销次数占用户总的不核销次数的比重
    - 用户对每个商家的优惠券核销次数占用户总的核销次数的比重
    - 用户对每个商家的不核销次数占商家总的不核销次数的比重
    - 用户对每个商家的优惠券核销次数占商家总的核销次数的比重
- 优惠券相关的特征

    - 优惠券类型(直接优惠为0, 满减为1)
    - 优惠券折率
    - 满减优惠券的最低消费
    - 历史出现次数
    - 历史核销次数
    - 历史核销率
    - 历史核销时间率
    - 领取优惠券是一周的第几天
    - 领取优惠券是一月的第几天
    - 历史上用户领取该优惠券次数
    - 历史上用户消费该优惠券次数
    - 历史上用户对该优惠券的核销率
- 其它特征

    这部分特征利用了赛题leakage，都是在预测区间提取的。

    - 用户领取的所有优惠券数目
    - 用户领取的特定优惠券数目
    - 用户此次之后/前领取的所有优惠券数目
    - 用户此次之后/前领取的特定优惠券数目
    - 用户上/下一次领取的时间间隔
    - 用户领取特定商家的优惠券数目
    - 用户领取的不同商家数目
    - 用户当天领取的优惠券数目
    - 用户当天领取的特定优惠券数目
    - 用户领取的所有优惠券种类数目
    - 商家被领取的优惠券数目
    - 商家被领取的特定优惠券数目
    - 商家被多少不同用户领取的数目
    - 商家发行的所有优惠券种类数目

模型设计与模型融合

基于以上提取到的特征，进行模型设计与融合。

- 单模型

第一赛季只训练了XGBoost单模型提交，连续几周位居排行榜第一位。

第二赛季训练了XGBoost，GBDT，RandomForest三种单模型，其中GBDT表现最好，XGBoost次之，RandomForest相比之下最差。GBDT和XGBoost单模型在第二赛季仍然名列Top3,融合后效果更佳，尝试了以下两种方法：

- 加权融合

  得到了单模型的预测结果后，直接将概率预测值进行加权融合，我们简单地用`0.65 * GBDT + 0.35 * XGBoost`就得到了第一的成绩。

- Blending模型

  我们尝试了两层的blending模型，首先将训练集分为两部分（D1和D2），一部分用于第一层（level 1）的训练，另一部分用于第二层（level 2）的训练。level1 在D1上训练了4个XGBoost，4个GBDT，4个RandomForest，将这些模型的预测结果作为level2的feature，在D2上训练第二层模型。Blending模型的结果相比单模型有细微的提升，但这点提升相对于模型复杂度带来的计算代价显得微不足道。