# 生活大实惠 O2O 优惠券使用预测

# 算法说明文档


| | |
|---|---|
| 队　　名： | 杨树 yang1026 |
| 团队成员： | 杨 阳 |
| 学　　校： | 重庆邮电大学 |
| 完成日期： | 2016.12.21 |

# 目　录

# 一．解决方案概述

## 1.1 问题描述与数据概览

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 天以内的使用情况。使用优惠券核销预测的平均 AUC（ROC 曲线下面积）作为评价标准，即对每个优惠券 coupon_id 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。

原始数据包括两张表：真实线上、线下消费行为表

线下表中 3 种行为：领券、用券消费、纯消费

线上表中 3 种行为：点击、购买、领取优惠券

## 1.2 算法整体思路概述

本题可转化为传统的二分类问题，评价标准为 AUC，主要通过线下消费行为表中的 user_id, coupon_id, merchant_id, distance, discount_rate, date_receive 和 date_pay 这 7 个属性进行构建模型。由于评价标准为基于 coupon_id 的平均 AUC，其本质为排序优化问题，所以本队在模型融合阶段主要使用的策略为同样基于排序优化的 RANK_AVG 融合方法。

本文首先从数据预处理开始，介绍了对缺失值的填充、原始表中属性类型的转化以及折扣率属性的转换；接着介绍了训练集的打标原则，训练集和预测集的划分与构建；然后是特征工程的介绍，主要包括了属性类型的划分、相关属性离散化、三大特征群、具体子特征群以及排序子特征群等内容；随后介绍了特征选择和类别不平衡数据的处理；最后是模型设计与多模型融合的介绍，本文重点介绍了多模型的异构和 RANK_AVG 融合方法。

# 二．数据预处理

## 2.1 缺失值的填充

在原始数据表中，缺失值为字符串’null’，为了便于后续操作，统一转化为NULL 类型。

## 2.2 数据类型的转换

Distance 字段：

字符串->double；其中“null”转为 null；

日期转换，包括 date_received 字段和 date_pay 字段：

字符串->DateTime；其中“null”转为 null。

## 2.3 折扣率转换

将满减转为折扣的形式，并增加折扣率类型、满、减三列。

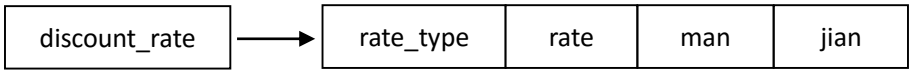


图 1 折扣率转换图

# 三．数据划分与打标

## 3.1 训练集打标原则

本文根据最后的评分原则，首先筛选出有领券日期的记录，然后将 15 天内消费的记录标 1，其余均标 0。具体打标原则如下图：

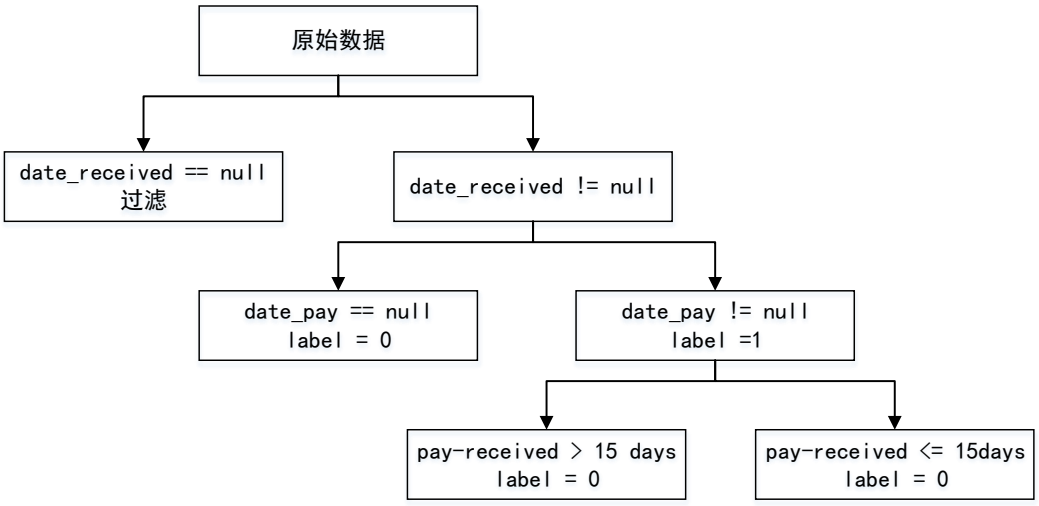


图 2 打标原则图

## 3.2 训练集与预测集构建

在进行特征提取之前，我们首先应该构建出自己的训练集合测试集。本文根据线上线下数据分布一致原则，分别测试了三种不同划分方案。最后选择了保留信息最完整同时也是效果最好的第三套方案。具体方案描述如下。

### 3.2.1 方案一

通过分析已有数据和我们要预测的数据，可以发现 date_received 相关记录时间段为 2016.01.01—2016.06.15，而任务要求预测数据中 date_received 时间段为 2016.07.01—2016.07.31。本着线上线下分布一致原则，本方案中训练集采用 2016.05.15—2016.06.15 时间段内数据（已过滤掉 date_received 为空数据）打标，2016.04.01—2015.05.01 时间段内数据提取特征，线上采用 2016.05.15—2016.06.15 时间段内数据提取特征。具体划分方案见下图：

| | 提取特征 | | 线下打标 |
|---|---|---|---|
| 线上待预测集合 | 5.15—6.15 | 6.15—6.30 | 7.1—7.31 |
| 线下训练集 1 | 4.1—5.1 | 5.1—5.15 | 5.15—6.15 |
| 线下训练集 2 | 3.1—4.1 | 4.1—4.15 | 4.15—5.15 |

图 3 数据集划分方案一

### 3.2.2 方案二

通过分析，我发现方案一种的划分方法会丢失 15 天的消费数据，所以在方案二中，我将特征提取和打标分别对待，即特征提取采用 date_pay 做划分依据，打标采用 date_received 做划分依据，这也是初赛时本队采用的划分方案，具体划分方案如下：

| | 提取特征<br>date_pay | 线下打标<br>date_received |
|---|---|---|
| 线上待预测集合 | 6.1—6.30 | 7.1—7.31 |
| 线下训练集 1 | 4.15—5.15 | 5.15—6.15 |
| 线下训练集 2 | 3.15—4.15 | 4.15—5.15 |
| 线下训练集 3 | 2.15—3.15 | 3.15—4.15 |

图 4 数据集划分方案二

### 3.2.3 方案三

进一步分析，可以看出方案二虽然保证了消费信息的完整性，但是却缺乏领取优惠券并消费相关信息。经过多次实验，本文最终采用了方案三的划分方案，即将原始表中的行为模式分为三类，其分别为打标当月纯领取优惠券行为、前一月消费行为以及前一个半月的领取优惠券并消费行为。具体划分方案如下：

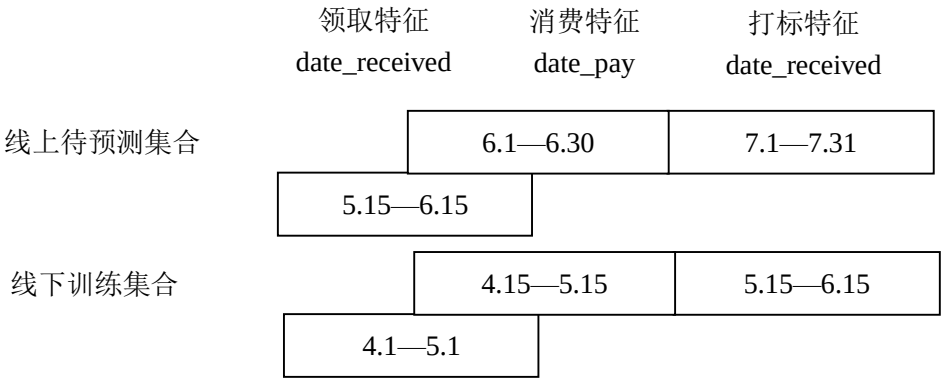


图 5 数据集划分方案三

# 四．特征工程

## 4.1 属性类型划分

在进行特征提取之前，本队首先做了属性类型划分工作。本文将原始 7 个属性分为了 key 类型和 value 类型。Key 类型属性主要包括 user_id, merchant_id 以及 coupon_id 三个属性，主要用于子特征群特征提取的 key 和多个子特征群合并时的 key, value 类型属性包括 distance, discount_rate, date_received 以及 date_pay 四个属性，主要用于提取相应特征。具体划分见下表：

| Key 类型属性 | Value 类型属性 |
|---|---|
| User_id<br>Coupon_id<br>Merchant_id | Distance<br>Discount_rate<br>Date_received<br>Date_pay |

表 1 特征划分表

## 4.2 特征群划分

本算法根据训练集的构建将特征群分为三大类别，分别为打标月特征群、消费月特征群、领券消费月特征群。三个特征群又根据每次做 key 的键不同分为 8 个子特征群。特征群的合并原则见 3.2.3。

## 4.3 相关特征离散化

### 4.3.1 distance 距离特征离散化

原始数据中对距离的表示存在 12 中表示(0-10，null)。在初赛中然后将距离等值离散到 3-5 个区间，线上 auc 有千分位提升。复赛中本队采用方案为，将距离特征作为数值型特征的同时也作为标称型特征，将其离散化为 12 个维度，除原数值相关统计外另增加其每个维度下的次数统计。在此以用户子特征群为例进行说明，统计该用户在每种距离下领取优惠券次数即可得到该用户的领取优惠券距离偏好。

### 4.3.2 时间信息处理

对时间信息处理最简单的方法是独热编码(one-hot encoding)，但这样得到的

特征过于稀疏，干扰模型的学习。我们在独热编码的基础上，先特征离散。下面介绍具体的方法。

赛题提供了用户在 2016.1.1 到 2016.6.31 的消费记录。时间特征有 date_pay、date_received 。我们分析了上中下旬的用户领券和消费的频率。

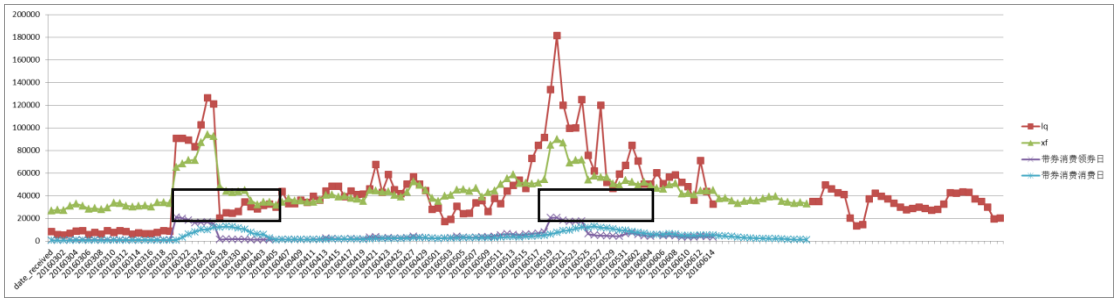

图 6 异常行为图

上图显示了每个月末的发券和消费量。因此我们构建一个特征；“是否上旬”、“是否中旬”、“是否下旬”，其取值为 0 或者 1。其实这就相当于对时间信息做了独热编码。同时我们将日期离散化为星期几、近 7 天、近 14 天和近 21 天 10 个维度，分别统计了每个子特征群中在这 10 个维度下的次数。在此以用户商家领取优惠券特征子群为例子进行业务说明。以 user_id 和 merchant_id 为 key，统计近 7 天的 date_received 数目其业务含义为用户最近是否领取了该商家的优惠券。

## 4.4 单个属性可提取特征

在进行特征提取前，本队先仔细分析了单个属性可得到的相关信息。通过此步骤为后续子特征群特征提取提供了完善的依据。单个属性可提取具体信息如下：

| 属性名 | 单个属性可提取特征 |
|---|---|
| user_id | 数目、众数、set 的数目 |
| merchant_id | 数目、众数、set 的数目 |
| coupon_id | 数目、众数、set 的数目 |
| discount_rate | 在每种折扣率下的数目统计 |
| discount | 数目、众数、set 的数目 |
| rate_type | 众数（用户最常领取的优惠券类型） |
| distance | 均值、方差、众数、每种 distance 下的数目 |
| date_received | 周一到周日每天的数目、近 7 天的数目、近 14 天的数目、近 21 天的数目<br>最近一次日期、最远一次日期 |
| date_pay | 周一到周日每天的数目、近 7 天的数目、近 14 天的数目、近 21 天的数目<br>最近一次日期、最远一次日期 |

表 2 单个属性可提取特征表

## 4.5 子特征群简介

根据每次的 key 值属性不同，将每个特征群进一步划分为 8 大子特征群。具体的特征如下：

1).user_id

merchant_id 数目、众数、set 的数目

coupon_id 数目、众数、set 的数目

discount 数目、众数、set 的数目

discount_rate 在每种折扣率下的数目统计

rate_type 的众数（用户最常领取的优惠券类型）

date_received 周一到周日每天的数目、近 7 天的数目、近 14 天的数目、近 21 天的数目

最近一次日期、最远一次日期

date_pay 周一到周日每天的数目、近 7 天的数目、近 14 天的数目、近 21 天的数目

最近一次日期、最远一次日期

distance 均值、方差、众数、每种 distance 下的数目

merchant_id

2).该商家发现数目

user_id 的众数

user_id 的 set

coupon_id 的 set

discount 的 set

doscount_rate 在每种折扣率数目下的统计

rate_type 的众数

date_received 周一到周五、最近、最远

distance 均值、方差、众数、每种 distance 下的数目

3).coupon_id

coupon_id 的数目

user_id 的众数、set

date_received 周一到周五、最近、最远

distance 均值、方差、众数、每种 distance 下的数目

4).discount_rate

数目

user_id 数目、众数

merchant_id 数目、众数

distance 均值、众数、方差、每种 distance 下的数目

5).user_id + merchant_id

数目

coupon_id 的 set

doscount_rate 每种 discount_rate 下的数目统计

distance 每种 ditance 下的数目统计、均值、方差、众数

date_received 每周天的数目统计、最近、最远、周末、平时、7 天、14

天、21 天

6).user_id + dicount_rate

数目

coupon_id 的 set

merchant_id 的数目、set、众数

distance 均值、方差、众数

date_received 每天的数目统计、最近、最远、周末、平时

7).user_id + ditance

数目

每种下的数目

merchant_id 的数目、set、众数

date_received 每天的数目统计、最近、最远、周末、平时

8).user_id + date_received

数目

每周天的数目、周末、平时数目

merchant_id 数目、set、众数

distance 均值、方差、众数、每种下的数目

## 4.6 消费日期行为特征

按日期统计训练集中每天的领券数和消费数，得到如下的曲线。横坐标是日期。纵坐标是每天的领券数或其他的数量。红色和绿色分为是每日发券数量和消费数量。

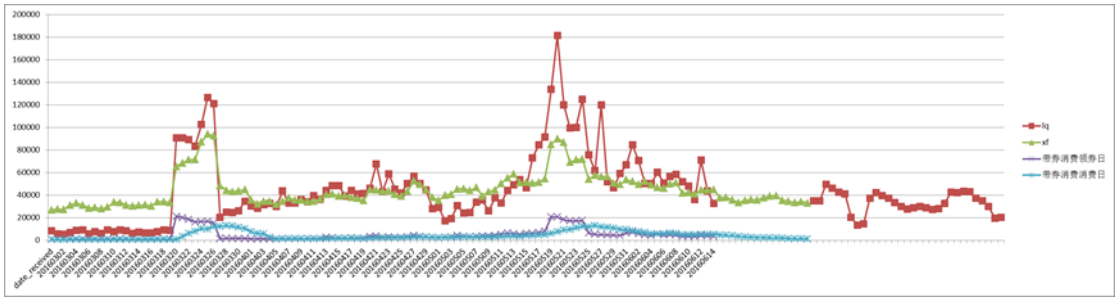

图 7 消费行为图

可以发现波峰主要集中在节假日的前面。在横坐标 5.1-5.7 对应的劳动节反而出现的发券低谷，这可能隐藏了某些信息，我们尚未挖掘出来。

每一次波峰后都会有一段时间的低谷。而后又缓慢上升。而消费波峰都会在发券波峰紧接着的后几天出现。因此我们我们可以将是否处于节假日前几天作为特征。

## 4.7 打标当月排序特征

通过分析本队发现一个用户当月可能会领取多张不同距离、不同折扣率以及不同日期的优惠券，基于这种事实，本队构建了 8 个基于打标月排序的排序特征，具体特征如下：

1.db_user_cid_date_received_rank：该用户当月领取同一张优惠券领取日期的排名

2.db_user_cid_oneday_cishu：该用户一天中同一张券的领取次数

3.db_user_everycid_rate：该用户在该商家领取优惠券次数比上该用户当月中的领取优惠券次数

4.db_user_rate_rank：转换过后的折扣率排名

5.db_user_distance_rank：距离排名（缺失值填 0）

6.db_user_date_received_rank：领取日期的排名

7.db_user_man_rank：满的排名

8.db_user_jian_rank：减的排名

## 4.8 行为比率特征

根据用户和商家的消费行为，本队统计了各个行为类别下的次数，并通过组合得到了行为比率特征群，通过添加该特征群，线上 AUC 由 0.770 提升到了 0.777（A 榜单）。部分特征如下：

1. 每个用户在每个商家的领券数占总领券数比率
2. 每个用户在每个商家的消费次数占总消费次数比率
3. 每个用户在每个商家的 15 天内用券数占 15 天内总消费次数比率
4. 每个用户用券消费次数占总消费次数比率
5. 每个用户不同距离下消费次数占比
6. 每个用户不同折扣率小消费占比
7. 节假日消费占比
8. 节假日领取优惠券占比
9. 每个商家发行优惠券数目占总优惠券数目比率
10. 每个商家每个优惠券发行数目占比

# 五．特征选择

## 5.1 传统的特征选择方案

在特征工程部分，本队构建了三个大特征群，每个特征群下又构建了10个子特征群，所有特征加起来共899维，这么多维特征一方面可能会导致维数灾难，另一方面很容易导致过拟合，需要做降维处理，降维方法常用的有如PCA，t-SNE等，这类方法的计算复杂度比较高。初赛阶段本队首先采用了采用的是基于模型的特征排序方法，其基于xgboost来做特征选择，xgboost模型训练完成后可以输出特征的重要性据此可以保留Top N个特征，从而达到特征选择的目的。采用此方案AUC提高了0.01。

## 5.2 基于视图的特征选择方案

在初赛阶段，本队最终采用了一种基于特征聚类的特征选择方案，该方案提升效果明显，使得AUC从0.72飙升到0.76。算法主要思路如下：初始化两个空集合，将已有数据集的所有属性放入其中集合A中，另一个集合B为空。从集合A随机选取一个子集放入集合B中。然后开始迭代：每轮迭代从集合A选择一个属性放入集合B，使得集合B属性的训练误差减小量与集合A属性的训练误差增加量的和sum最小。当B的训练误差与A的训练误差差值最小时，停止迭代。此时集合A，B就是分离的两个视图。最后在分离出的视图中利用基于xgb的特征选择各选择出TOPK个特征进行训练。在初赛阶段本队主要提取了消费特征群和打标特征群，故子分离了两个试图，一般视图个数同特征群个数，特此说明。算法流程图如下：

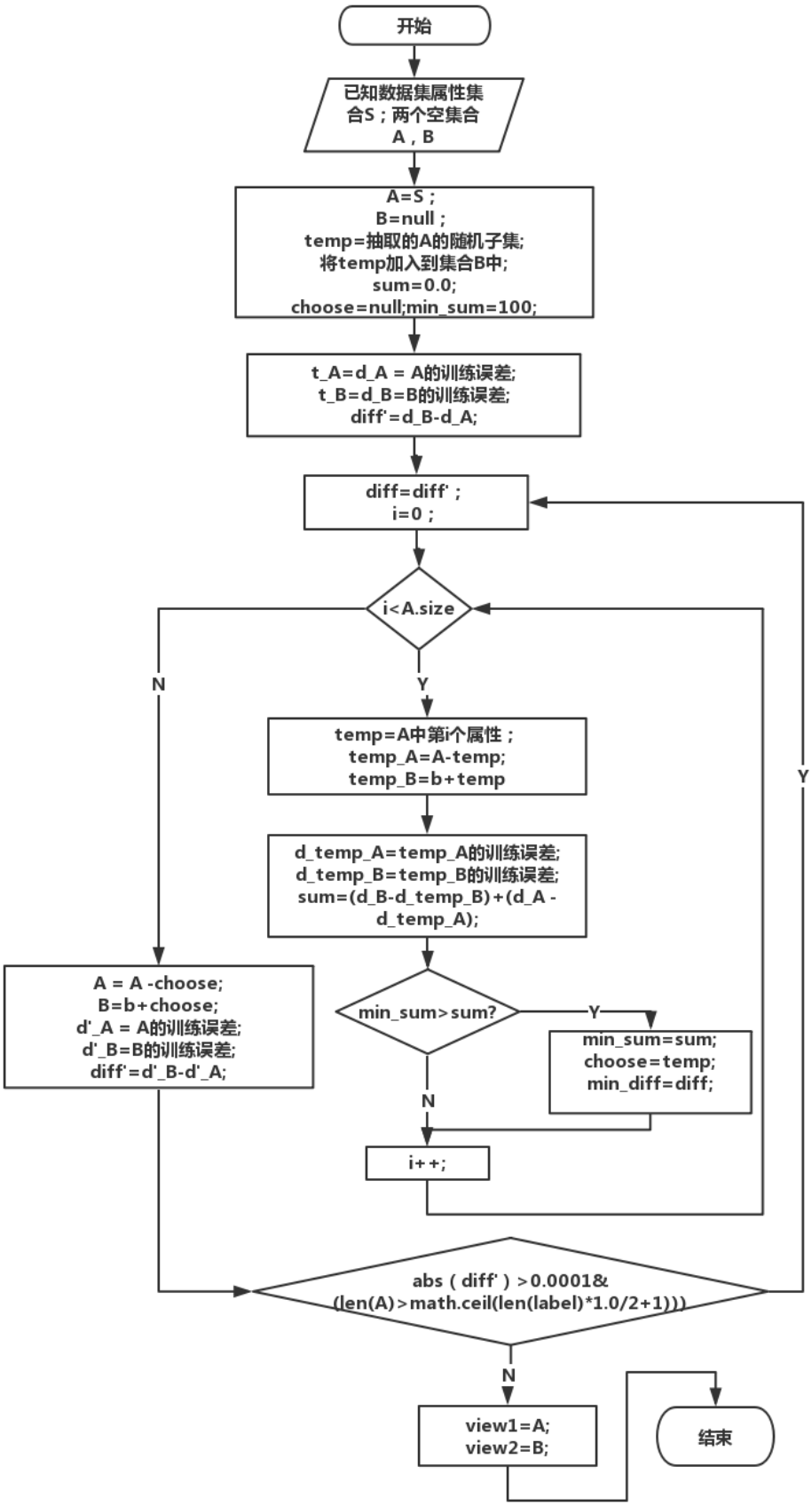


图8 特征视图分离流程图

# 六．类别不平衡处理

通过分析可发现该赛题的类别比例近14：1，为典型的类别不平衡分类问题。对于列别不平衡分类问题有两种常用的解决方案，一是在训练模型是设置类别权重，二是采用过采样方法。本队在初赛和复赛阶段均采用了过采样方案。

# 七．模型设计与融合

## 7.1 模型设计

本队在机器学习平台主要调用了三种算法，分别为逻辑回归，GBDT以及XGBBOOST。下面将对这三种模型进行简要介绍并对比其效果。

1. 逻辑回归（Logistic Regression）是机器学习中的一种分类模型，由于算法的简单和高效，在实际中应用非常广泛。其主要由两部分组成分别为：logistics方程以及规范化正则项。但是其在模型特征维度过大时，容易出现过拟合问题。本队初赛阶段尝试过该算法，效果并不好（0.67左右），因而复赛阶段并未采用此算法。
2. GBDT(Gradient Boosting Decision Tree) 又叫 MART(Multiple Additive Regression Tree)，是一种迭代的决策树算法，该算法由多棵决策树组成，所有树的结论累加起来做最终答案。它在被提出之初就和SVM一起被认为是泛化能力（generalization)较强的算法。近些年更因为被用于搜索排序的机器学习模型而引起大家关注。复赛阶段前期本队均采用的是GBDT，单模型AUC 0.770左右。
3. XGBOOST的全称是eXtreme Gradient Boosting。正如其名，它是Gradient Boosting Machine的一个C++实现，效率和精度都很高，在各类数据挖掘竞赛中被广泛使用。复赛阶段本队使用全部899个特征，单XGB模型AUC达到了0.786。

## 7.2 多模型融合

结合ensemble思想，我们可以知道单个模型的结果不够理想，多个异构模型的融合可以有效的提升算法精度。

### 7.2.1 异构模型的实现

在进行多模型融合之前，由Bagging思想可知，单个模型尽可能异构，结果的皮尔森系数在0.7—0.8之间融合效果最佳。常用的异构方案可从以下三个层面进行，分别为基于特征选择的异构、基于样本选择的异构以及基于不同分类算法的异构。本队经过尝试最终选取了三个AUC在0.787左右（分别为全部特征下的

XGB、全部特征下的GBDT以及700维度特征下的XGB）的模型进行融合，最终AUC提升到了0.75。

### 7.2.2 多模型融合方案简介

常用的多模型融合方案主要有三种，分别为：

1. 按照权重融合

结合真实数据走势，将多个结果按照权重进行融合，建模思路不同时效果提升明显；

2. 集成学习思路

训练两层模型，第一层的输出作为第二层的特征输入；

3. 基于排名的rank_avg

$$\Sigma weight_i / rank\_i$$

由于评价标准为基于coupon_id的平均AUC，其本质为。排序优化问题，所以我在模型融合阶段主要使用的策略为同样基于排序优化的RANK_AVG融合方法。具体融合方案如下：

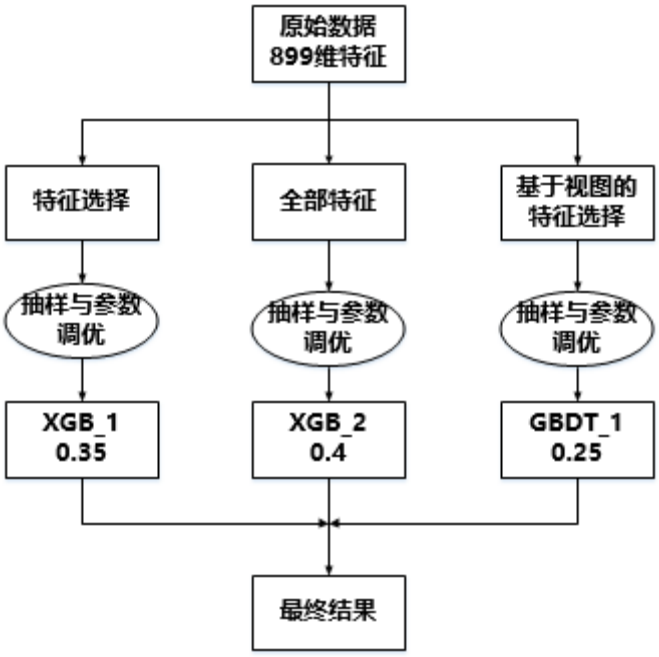


图 9 模型融合图