

同济大学
计算机科学与技术系

数据挖掘大作业实验报告
面向电信行业存量用户的智能套餐个性化
匹配模型



学 号 1754272

姓 名 王云杉

专 业 计算机科学与技术

目录

1.	任务简介.....	3
1.1.	挖掘目标.....	3
1.2.	数据描述.....	3
1.2.1.	训练集.....	3
1.2.2.	测试集.....	4
2.	模型.....	4
2.1.	数据预处理.....	5
2.2.	数据分析.....	5
2.3.	模型的选择.....	7
2.3.1.	GBDT 梯度提升决策树.....	7
2.3.2.	基于 Histogram 的决策树算法.....	8
2.3.3.	Leaf-wise 算法.....	10
2.4.	特征工程.....	10
2.5.	模型评估方法.....	11
2.5.1.	简单随机拆分.....	11
2.5.2.	K 折交叉验证.....	11
2.5.3.	简单随机拆分与 K 折交叉验证的区别.....	12
3.	实验.....	13
3.1.	参数说明.....	13
3.2.	评测指标.....	13
3.3.	实验结果.....	14
3.4.	结论.....	14

1. 任务简介

近年来，电信运营商推出大量的电信套餐以满足用户的差异化需求，面对种类繁多的套餐，如何选择最合适的一款对于运营商和用户来说都至关重要，尤其是在电信市场增速放缓，存量用户争夺愈发激烈的大背景下。针对电信套餐的个性化推荐问题，通过数据挖掘技术构建基于用户消费行为的电信套餐个性化推荐模型，根据用户业务行为画像结果，分析出用户消费习惯及偏好，匹配用户最合适的套餐，提升用户感知，带动用户需求，从而达到用户价值提升的目标。

1.1. 挖掘目标

根据所提供的用户结构化数据，如用户属性（个人基本信息、用户画像信息等）、业务属性、消费习惯及偏好等，基于本学期所学知识构建一个分类模型，为用户选择最适合的套餐。

1.2. 数据描述

数据包含训练集 train.csv，用来训练模型；测试集 test.csv，用来测试模型性能。

1.2.1. 训练集

训练集 train.csv 提供了用户相关的数据，包括数值型、二元型（binary）和类别型（categorical），各字段的含义如下：

字段	中文名	数据类型	说明
service_type	套餐类型	类别型	0:2\3G 融合 1:2I2C, 2:2G, 3:3G, 4:4G
is_mix_service	是否固移融合套餐	二元型	1. 是 0. 否
online_time	在网时长	数值型	单位：月
1_total_fee	当月总出账金额_月	数值型	单位：元
2_total_fee	当月前 1 月总出账金额_月	数值型	单位：元
3_total_fee	当月前 2 月总出账金额_月	数值型	单位：元
4_total_fee	当月前 3 月总出账金额_月	数值型	单位：元
month_traffic	当月累计-流量	数值型	单位：MB
many_over_bill	连续超套(连续 4 个月)	二元型	1-是，0-否
contract_type	合约类型	类别型	
contract_time	合约时长	数值型	
is_promise_low_consume	是否承诺低消用户	二元型	1. 是 0. 否

net_service	网络口径用户	类别型	
pay_times	交费次数	数值型	单位：次
pay_num	交费金额	数值型	单位：元
last_month_traffic	上月结转流量	数值型	单位：MB
local_traffic_month	月累计-本地数据流量	数值型	单位：MB
local_caller_time	本地语音主叫通话时长	数值型	单位：分钟
service1_caller_time	套外主叫通话时长	数值型	单位：分钟
service2_caller_time	上一月超套餐时长	数值型	单位：分钟
gender	性别	类别型	1：男 2：女
age	年龄	数值型	
complaint_level	投诉重要性	类别型	1：普通，2：重要，3：重大
former_complaint_num	交费金历史投诉总量	数值型	单位：次
former_complaint_fee	历史执行补费用交费金额	数值型	单位：分
current_service	向用户推荐的套餐		训练数据的标签
user_id	用户 ID		用户编码，标识用户的唯一字段

contract_type 合约类型编码 合约类型描述：

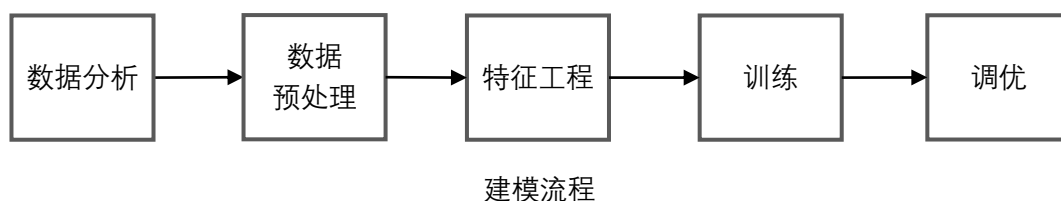
- 1 预付话费送手机
- 2 购手机送话费
- 3 存费送费
- 4 单卡
- 5 总部其他活动
- 6 存话费送业务（送语音）
- 7 存话费送业务（送流量）
- 8 存话费送业务（送短信）
- 9 存费送业务（不能区分赠送业务的类型）
- 10 存费送礼品
- 11 存费送积分
- 12 合约惠机终端
- 13 订业务送手机

1.2.2. 测试集

测试集 test.csv 的字段和训练集相同。

2. 模型

本次作业采用了基于 LightGBM 的 GBDT 模型，并使用了人工特征、K 折交叉验证算法建立智能套餐个性化匹配模型。建模流程如下



2.1. 数据预处理

通过对训练集和测试数据读取和分析可知，训练集数据大小：595192，训练集数据用户数：595192，没有重复数据，其中去掉用户列和标签列，原始特征为 25 项，通过数据描述和对数据的观察将其分为数字特征和分类特征，具体分类如下：

分类特征：['service_type', 'complaint_level', 'contract_type', 'is_mix_service', 'gender', 'is_promise_low_consume', 'many_over_bill', 'net_service']

数字特征：['1_total_fee', '2_total_fee', '3_total_fee', '4_total_fee', 'age', 'contract_time', 'former_complaint_fee', 'former_complaint_num',

分类标签为 current_service，共 11 类，{89950166, 89950167, 89950168, 90063345, 90109916, 90155946, 99999825, 99999826, 99999827, 99999828, 99999830}分别代表 11 种套餐类型。

测试集数据大小为 148798，测试集用户数也为 148798，无重复数据，而且测试集与训练集无相同 user_id 用户。

发现部分数存在异常类型，例如：1_total_fee、4_total_fee 和 2_total_fee、3_total_fee 应为相同的类型，需要转换为 float 类型，而 age、gender 需要转换为 int 类型，而 gender 为性别，按照数据描述 1: 男 2: 女，而数据中存在 0，判断应为性别未填写，统一替换为 np.nan，通过数据预处理，保证剔除脏数据，保证数据的一致性。

```
def deal(data):
    for i in origin_num_feature:
        data[i] = data[i].apply(lambda x: astype(x, float))
        data.loc[data[i] < 0, i] = np.nan
        data[i] = data[i].round(4)
    data['age'] = data['age'].apply(lambda x: astype(x, int))
    data.loc[data['age']==0, 'age'] = np.nan
    data['gender'] = data['gender'].apply(lambda x: astype(x, int))
    return data
train = deal(train)
test = deal(test)
```

2.2. 数据分析

经过数据预处理后，可以对数据进行初步的数据分析，通过 pandas 的 describe 函数对训练数据进行快速统计分析，了解数据的基本情况：

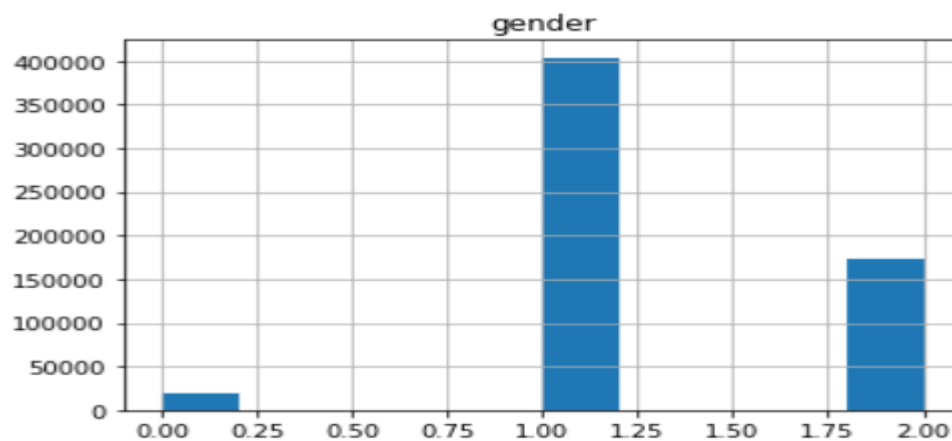
```
train[origin_num_feature].describe()
```

	1_total_fee	2_total_fee	3_total_fee	4_total_fee	age	contract_time
count	595192.000000	594365.000000	594270.000000	594532.000000	577610.000000	594038.000000
mean	105.349933	111.346573	103.006657	111.062438	32.367890	7.790852
std	90.910051	98.654249	94.909072	102.956568	10.673583	10.354471
min	0.000000	0.000000	0.000000	0.000000	12.000000	0.000000
25%	52.200000	54.600000	47.450000	51.700000	24.000000	0.000000
50%	76.000000	80.000000	76.000000	78.500000	30.000000	0.000000
75%	127.480000	136.000000	129.107500	137.050000	39.000000	12.000000
max	5940.830000	5825.570000	7242.010000	5141.270000	99.000000	52.000000

针对特征数据进行分析，例如性别的分布：

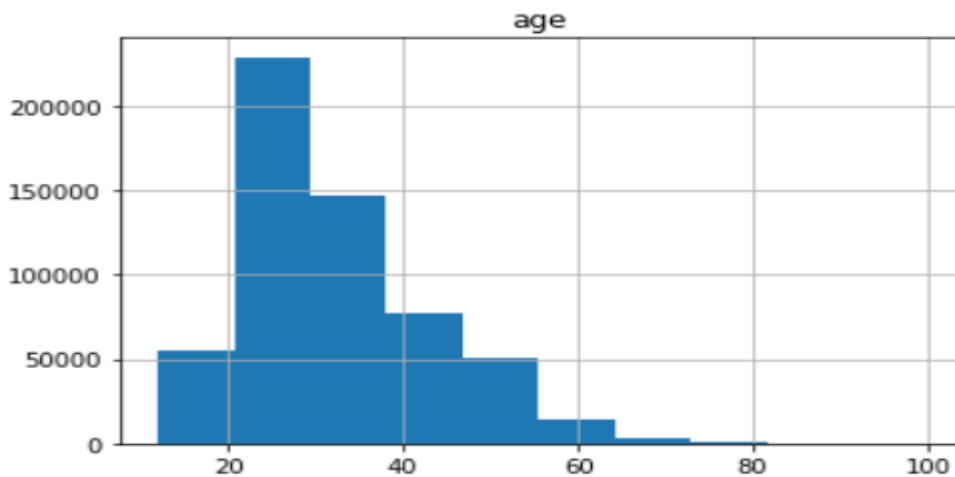
```
: train.hist('gender')
```

```
: array([[<matplotlib.axes._subplots.AxesSubplot object
dtype=object)])
```



可以观察到性别中有 0 的缺省值，对于这部分，在实验中使用了两种方法处理，一种是填充 service_type 对应字段的众数和原始值。最终选取了原始值，避免改变数据分布差异，因为默性别在不同套餐中的转换率呈现了分布差异。

年龄的分布：



可以看到对于年龄来讲，基本上符合电信用户群体的分布，但是有很多 0 岁的异常值，对于异常值，在数据预处理时选取了原始值，未进行修改，因为默认年龄在不同套餐中的转换率呈现了分布差异。

2.3. 模型的选择

本次作业在个人笔记本进行训练，根据实际情况采用了基于 LightGBM 的 GBDM 模型，其具有训练效率快、内存使用低、更好的准确率、可处理大规模数据的特点，具有较高的性价比。

LightGBM 是微软开发的一款快速、分布式、高性能的基于决策树的梯度 Boosting 模型，具有训练效率快、内存使用低、更好的准确率、支持并行学习、可处理大规模数据的特点。该模型优化了对类别特征的支持，可以直接输入类别特征，不需要额外的 0/1 展开，并在决策树算法上增加了类别特征的决策规则，因此适合与本次作业中电信个性化套餐匹配的分类要求。

2.3.1. GBDT 梯度提升决策树

GBDT 是梯度提升决策树（Gradient Boosting Decision Tree）的简称，梯度提升算法是一种通用的学习算法，其思想是通过调整模型，让损失函数的值不断减小，然后将各个模型加起来作为最终的预测模型。

GBDT 其主要思想是利用弱分类器（决策树）迭代训练以得到最优模型，通过多轮迭代，每轮迭代产生一个弱分类器，每个分类器在上一轮分类器的残差基础上进行训练。对弱分类器的要求一般是足够简单，并且是低方差和高偏差的。因为训练的过程是通过降低偏差来不断提高最终分类器的精度。

GBDT 基础模型使用 CART（也就是分类回归树）。由于上述高偏差和简单的要求，每个分类回归树的深度不会很深。最终的总分类器是将每轮训练得到的弱分类器加权求和得到的（也就是加法模型）。

GBDT 的核心是让损失函数沿着梯度方向的下降。利用损失函数的负梯度在当前模型的值作为回归问题提升树算法中的残差的近似值去拟合一个回归树。GBDT 每轮迭代的时候，都去拟合损失函数在当前模型下的负梯度。

在学习过程中，第 t 轮迭代的目标是找到一个 CART 回归树 $g_t(x)$ 让本轮的损失函数 $L(y, G_t(x)) = L(y, G_{t-1}(x) + g_t(x))$ 尽量小。

1. Initialize $f_0(x) = \arg \min_{\gamma} \sum_{i=1}^N L(y_i, \gamma)$.

2. For $m = 1$ to M :

(a) For $i = 1, 2, \dots, N$ compute

$$r_{im} = - \left[\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)} \right]_{f=f_{m-1}}.$$

(b) Fit a regression tree to the targets r_{im} giving terminal regions R_{jm} , $j = 1, 2, \dots, J_m$.

(c) For $j = 1, 2, \dots, J_m$ compute

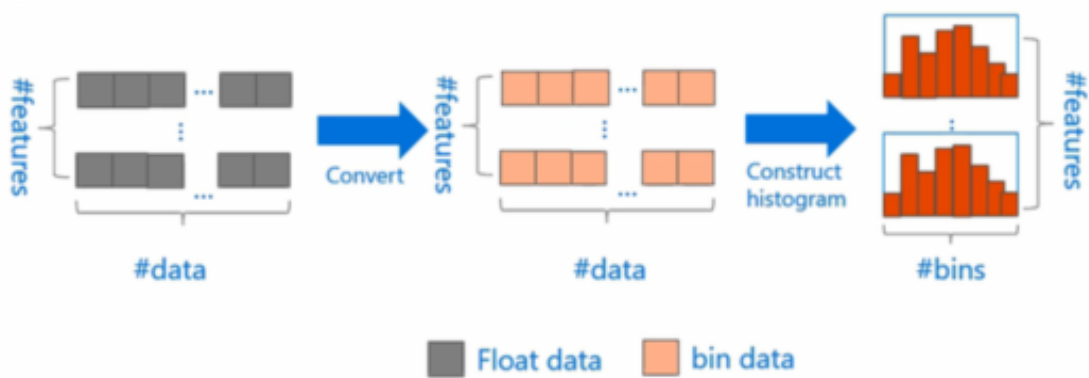
$$\gamma_{jm} = \arg \min_{\gamma} \sum_{x_i \in R_{jm}} L(y_i, f_{m-1}(x_i) + \gamma).$$

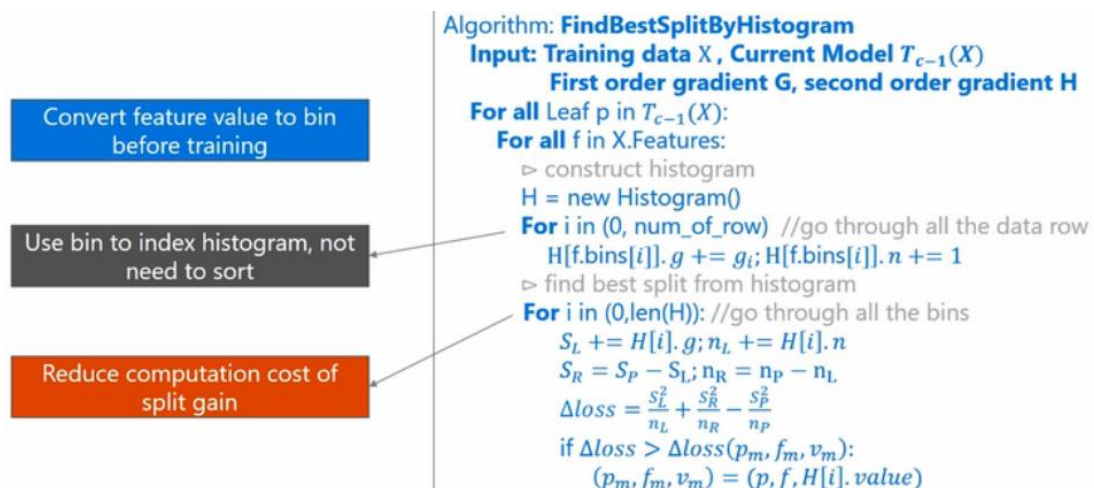
(d) Update $f_m(x) = f_{m-1}(x) + \sum_{j=1}^{J_m} \gamma_{jm} I(x \in R_{jm})$.

3. Output $\hat{f}(x) = f_M(x)$.

2.3.2. 基于 Histogram 的决策树算法

把连续的浮点特征值离散化成 k 个整数，同时构造一个宽度为 k 的直方图。在遍历数据的时候，根据离散化后的值作为索引在直方图中累积统计量，当遍历一次数据后，直方图累积了需要的统计量，然后根据直方图的离散值，遍历寻找最优的分割点。





直方图优化算法需要在训练前预先把特征值转化为 bin value，也就是对每个特征的取值做个分段函数，将所有样本在该特征上的取值划分到某一段（bin）中。最终把特征取值从连续值转化成了离散值。需要注意得是：feature value 对应的 bin value 在整个训练过程中是不会改变的。

最外面的 for 循环表示的意思是对当前模型下所有的叶子节点处理，需要遍历所有的特征，来找到增益最大的特征及其划分值，以此来分裂该叶子节点。

在某个叶子上，第二个 for 循环就开始遍历所有的特征了。对于每个特征，首先为其创建一个直方图。这个直方图存储了两类信息，分别是每个 bin 中样本的梯度之和（ $H[f.bins[i]].g$ ），还有就是每个 bin 中样本数量（ $H[f.bins[i]].n$ ）

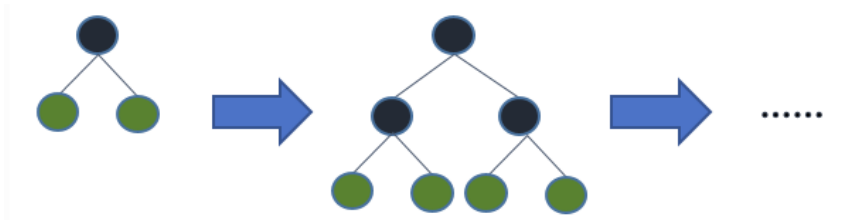
第三个 for 循环遍历所有样本，累积上述的两类统计值到样本所属的 bin 中。即直方图的每个 bin 中包含了一定的样本，在此计算每个 bin 中的样本的梯度之和并对 bin 中的样本记数。

最后一个 for 循环，遍历所有 bin，分别以当前 bin 作为分割点，累加其左边的 bin 至当前 bin 的梯度和（ S_L ）以及样本数量（ n_L ），并与父节点上的总梯度和（ S_P ）以及总样本数量（ n_P ）相减，得到右边所有 bin 的梯度和（ S_R ）以及样本数量（ n_R ），带入公式，计算出增益，在遍历过程中取最大的增益，以此时的特征和 bin 的特征值作为分裂节点的特征和分裂特征取值。

通过每个叶子的父亲节点的直方图与它兄弟节点的直方图做差获得该叶子的直方图，提升一倍速度。对于稀疏特征，只需要 $O(2 * \text{\#non_zero_data})$ 来构建直方图实现了稀疏特征优化直方图。histogram 算法的缺点是不能找到很精确的分割点，训练误差没有 pre-sorted 好。但从实验结果来看，histogram 算法在测试集的误差和 pre-sorted 算法差异并不是很大，甚至有时候效果更好。实际上可能决策树对于分割点的精确程度并不太敏感，而且较“粗”的分割点也自带正则化的效果。

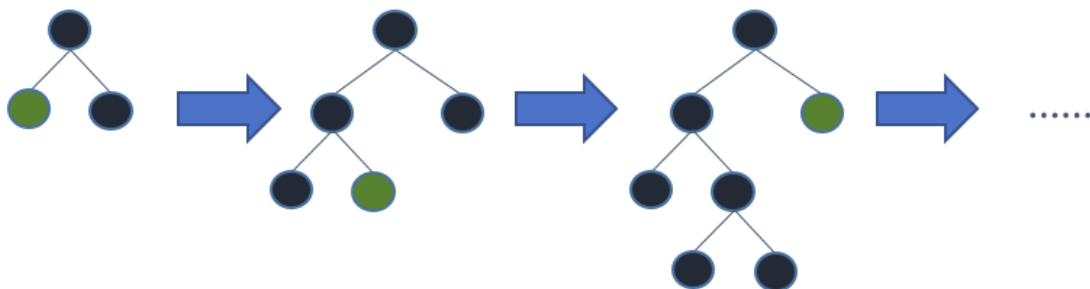
2.3.3. Leaf-wise 算法

Leaf-wise 摒弃了现在大部分 GBDT 使用的按层生长 (level-wise) 的决策树生长策略, 使用带有深度限制的按叶子生长 (leaf-wise) 的策略。level-wise 过一次数据可以同时分裂同一层的叶子, 容易进行多线程优化, 也好控制模型复杂度, 不容易过拟合。但实际上 level-wise 是一种低效的算法, 因为它不加区分的对待同一层的叶子, 带来了不少不必要的开销, 因为实际上很多叶子的分裂增益较低, 没必要进行搜索和分裂。



Leaf-wise tree growth 示意图

Leaf-wise 使用了一种更为高效的策略, 每次从当前所有叶子中, 找到分裂增益最大的一个叶子, 然后分裂, 如此循环。因此同 Level-wise 相比, 在分裂次数相同的情况下, Leaf-wise 可以降低更多的误差, 得到更好的精度。Leaf-wise 的缺点是可能会长出比较深的决策树, 产生过拟合。因此 LightGBM 在 Leaf-wise 之上增加了一个最大深度的限制, 在保证高效率的同时防止过拟合。



Leaf-wise tree growth 示意图

2.4. 特征工程

本次作业首先采用全部原始特征进行训练, 发现通过调参其 macro-f1 得分基本很难提高, 因此需要通过添加人工特征来提高准确率, 通过利用关联规则、业务特征对原始特征进行分析, 对用户业务行为画像。

1) 关联规则

关联规则是形如 $X \rightarrow Y$ 的蕴涵式, 其中, X 和 Y 分别称为关联规则的先导(antecedent 或 left - hand - side,)和后继(consequent 或 right - hand - side,RHS)。其中, 关联规则 XY , 存在支持度和信任度。例如: 观察发现连续至并不连续, 1、2、3、4 四个月的账单金额数据存在的一定的稳定性, 因为用户的行为习惯具有稳定性, 而且出现每个月缴费项目和套餐项目可能一致的情况, 与用户套餐存在着关联, 因此对四个月的账单进行统计计数和交叉组合, 对四个月的账单作 count、mean、max、min、std、gap, 可以构造出相应的特征; 再如: 流量与在网时间, 出现结转流量、合约时间, 与套餐更换相关; 缴费金额与使用项目相

关，可能随时需要缴费，这些都与套餐的选择有相关性。

2) 用户套餐使用特征

通过分析电信的套餐消费场景，了解各种套餐特点和用户在使用套餐的过程中的习惯趋向性，力求尽可能的描绘出一幅用户画像，例如：每个月的费用相同的用户与套餐选择和用户习惯存在着关系；超过套餐的通话时间与套餐选择的关系；两个月超套餐通话时间分别占比反应了历史通话和现在通话对套餐选择的影响；流量与在网时间，出现结转流量、合约时间，与用户套餐更换的习惯；缴费金额与用户使用项目的关系；上月结转和套餐内流量对套餐选择的影响等等。针对这些细节对用户的流量和通话计算出了一系列特征：比例、差值，求和等，来体现用户行为习惯的某一趋向性。。

通过利用关联规则、用户套餐使用特征分析方法，推导出用户消费习惯及偏好。共计算得到 74 个特征，将所有特征拼接，并通过归一化操作生成特征矩阵。

2.5. 模型评估及融合

在训练有监督的机器学习模型的时候，为了能够选出效果最好的，泛化能力最佳的模型，一般将数据划分为训练集、验证集和测试集。对原始数据进行三个数据集的划分，也是为了防止模型过拟合。当使用了所有的原始数据去训练模型，得到的结果很可能是该模型最大程度地拟合了原始数据，即该模型是为了拟合所有原始数据而存在。当新的样本出现，再使用该模型进行预测，效果可能还不如只使用一部分数据训练的模型。

训练集 (Training set)：

用来拟合模型，通过设置分类器的参数，训练分类模型。后续结合验证集作用时，会选出同一参数的不同取值，拟合出多个分类器。

验证集 (Cross Validation set)：

当通过训练集训练出多个模型后，为了能找出效果最佳的模型，使用各个模型对验证集数据进行检测，并记录模型准确率。选出效果最佳的模型所对应的参数，即用来调整模型参数。

测试集 (Test set)：

通过训练集和验证集得出最优模型后，使用测试集进行模型预测。用来衡量该最优模型的性能和分类能力。即可以把测试集当做从来不存在的数据集，当已经确定模型参数后，使用测试集进行模型性能评价。

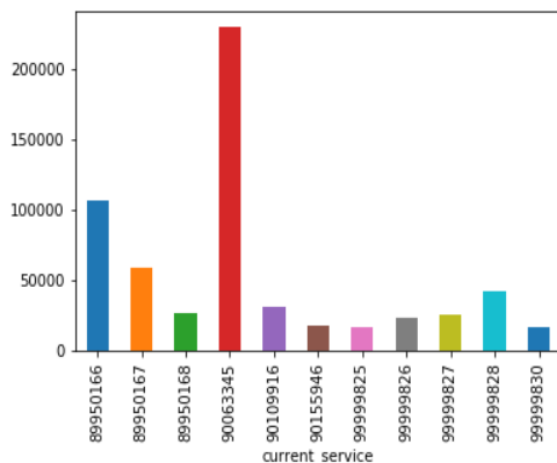
2.5.1. 简单随机拆分

通过将数据随机拆分为一定比例的训练集和验证集，一般拆为 80%训练集 20%验证集或 70%训练集 30%验证集。使用训练集训练，然后使用验证集测试模型效果。

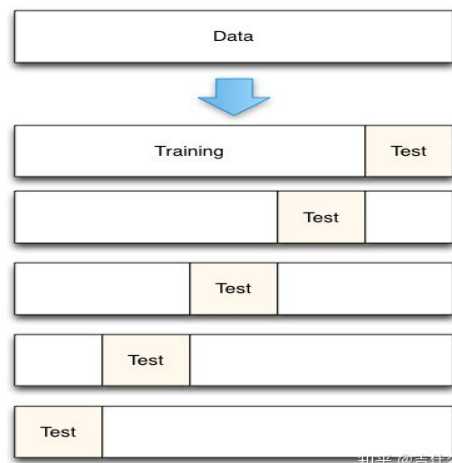
2.5.2. K 折交叉验证

K 折交叉验证是指将数据集划分为 k 份，依次选择其中一份作为验证集，其余 k-1 份作为训练集，可训练得到 k 个模型。通常 K 折交叉验证是通过 k 个模型的平均准确率来验证模型的泛化能力。

通过对训练集中的分类作频率分布分析，可以看出个套餐的分布并不均衡：



训练集中的套餐分类分布图

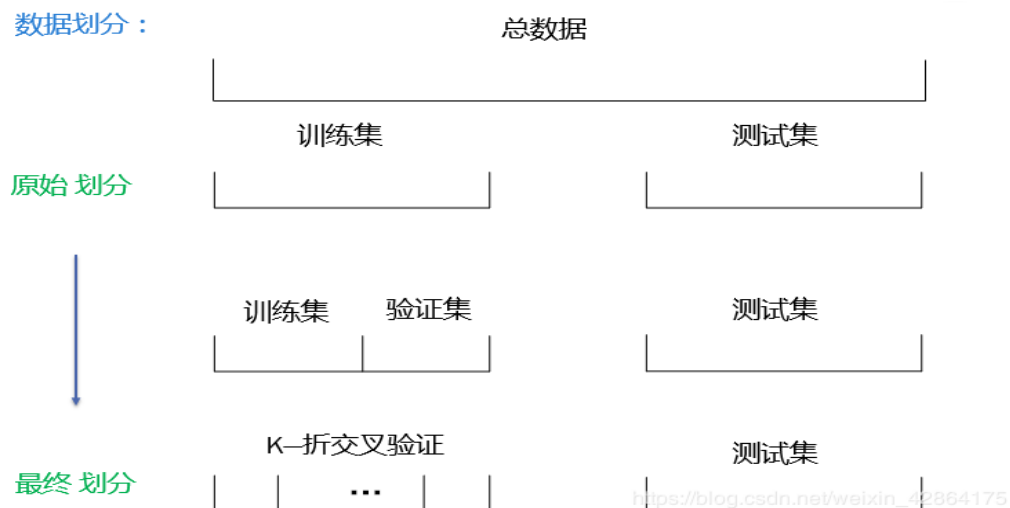


K 折交叉验证训练示意图

为了使 K 折交叉验证时，数据集切分的子训练集中各类样本比例与原始数据集中分布相同，因此本次作业采用了 StratifiedKFold 分层采样交叉切分，将 K 个模型的预测结果计算 **macro-f1** 取平均作为最终的结果。

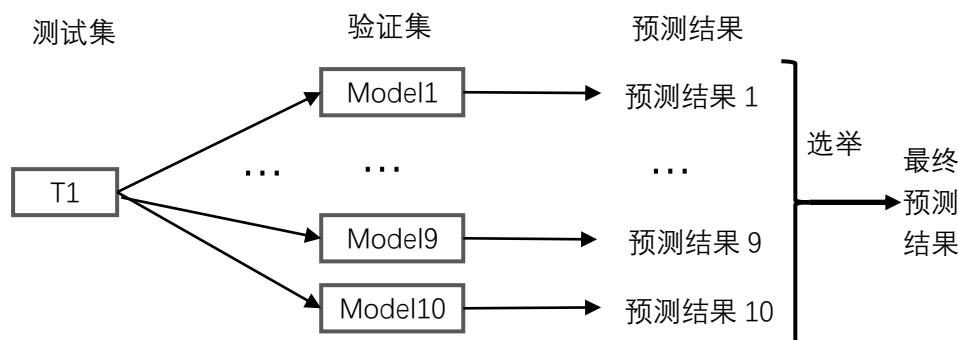
2.5.3. 简单随机拆分与 K 折交叉验证的区别

数据划分：



2.5.4. 多模型融合

在本次作业中，使用了最简单易实现的根据预测套餐类型的出现频率进行选举投票进行模型的融合，选择最大出现频率的套餐类型作为最终的结果，并分别实验了 5 模型融合和 10 模型融合，充分发挥多模型的优势。



K 折交叉验证预测示意图

3. 实验

电信套餐个性化推荐是利用已有的用户属性(如个人基本信息、用户画像信息等)、终端属性(如终端品牌等)、业务属性、消费习惯及偏好匹配用户最合适的套餐, 对用户进行推送, 完成后续个性化服务。本次作业中通过对原始特征、增加人工特征、单训练集验证、K 折交叉验证、参数调整等多种方式进行实验提高预测的准确率, 训练模型所使用的具体参数信息, 衡量模型性能的评测指标, 以及模型在训练和预测的实验结果。

3.1. 参数说明

电信套餐个性化推荐匹配模型为多分类问题, 因此将"objective"设置为'multiclass'; "num_class"为分类数量, 根据训练集中的套餐类型设置为 11; "learning_rate"为学习率, 设置为 0.035, 越大收敛速度越快, 减小可提高准确率; "boosting_type"提升算法类型为 gbdt, "num_leaves"为每个树上的叶子数, 设置为 128, 深度越高、准确率越高, 过大会导致过拟合; "lambda_l2"正则化系数设置为 0.25, "max_depth"为每棵树的最大深度或生长的层数上限, 设置为 -1 表示无限制; "subsample"为采样率, 设置为 0.9, 防止过拟合。

3.2. 评测指标

本次作业采用 **macro-f1** 作为衡量预测结果准确性的指标, 具体的计算方法如下:

- 1) 针对每个用户套餐类别, 分别统计 TP (预测答案正确), FP (错将其他类预测为本类), FN (本类标签预测为其他类标);
- 2) 通过第一步的统计值计算每个类别下的 precision 和 recall, 计算公式如下:

$$precision_k = \frac{TP}{TP + FP}$$

$$recall_k = \frac{TP}{TP + FN}$$

- 3) 通过第二步计算结果计算每个类别下的 F1-score, 计算方式如下:

$$f1_k = \frac{2 * precision_k * recall_k}{precision_k + recall_k}$$

4) 通过第三步求得的各个类别下的 F1-score 求均值, 得到最后的评测结果, 计算方式如下:

$$\text{score} = \frac{1}{n} \sum_{k=0}^n f1_k$$

3.3. 实验结果

本次作业分别采用了简单随机拆分 F1-score 验证和 K 折交叉验证两种方式进行了实验, 分别按照原始特征和原始+人工特征进行训练, 其中 K 折交叉验证的效果明显好于简单随机拆分, 增加了人工特征后取得了更好的效果, 利用训练所得的模型对测试集 test 进行预测, 通过老师提供的评测脚本 evaluate.py 计算的 F1-score 均高于训练的得分, 充分说明了模型的可用性。

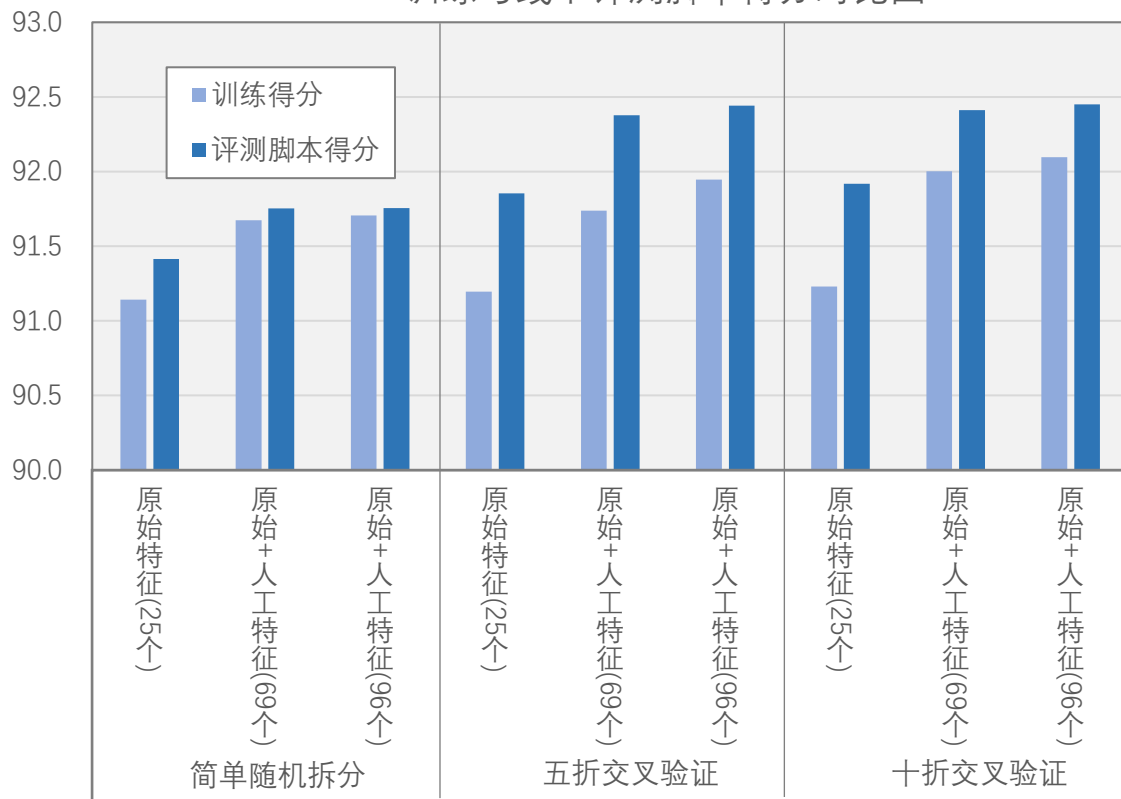
模型	训练得分	评测脚本得分
原始特征 (25 个) + 简单随机拆分	0.9114295095608923	0.9141496731371053
原始特征 (25 个) + 5 折交叉验证	0.9119553003815947	0.918539671402037
原始特征 (25 个) + 10 折交叉验证	0.9122940480327252	0.919187095689719
原始+人工特征(69 个)+简单随机拆分	0.9167484130696745	0.9175332734878495
原始+人工特征(69 个)+ 5 折交叉验证	0.9173900464685149	0.9237725218075343
原始+人工特征(69 个)+ 10 折交叉验证	0.9200269269136964	0.9241091209228712
原始+人工特征(96 个)+ 简单随机拆分	0.9170626387248074	0.9175577593915581
原始+人工特征(96 个)+ 5 折交叉验证	0.9194654365216082	0.9244148899073786
原始+人工特征(96 个)+ 10 折交叉验证	0.9209528054716338	0.9244997563303033

3.4. 结论

电信套餐个性化推荐是利用已有的用户属性(如个人基本信息、用户画像信息等)、终端属性(如终端品牌等)、业务属性、消费习惯及偏好进行预测, 为用户匹配最合适的套餐, 本次作业中通过对原始特征、增加人工特征、单训练集验证、K 折交叉验证、参数调整等多种方式进行实验提高预测的准确率, 实验采用了基于 LIGTHGBM 的 GBDT 模型, 通过实验结果可以得出以下结论:

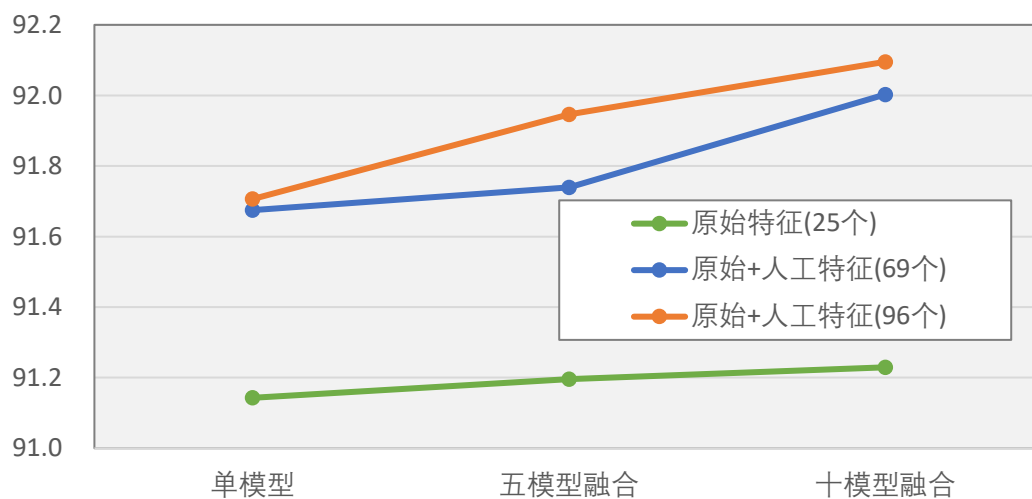
1、基于 LIGTHGBM 的 GBDT 模型在本次作业中取得了较好的预测效果, 线下评测脚本得分均高于训练得分, 具有较强的泛化能力。

训练与线下评测脚本得分对比图

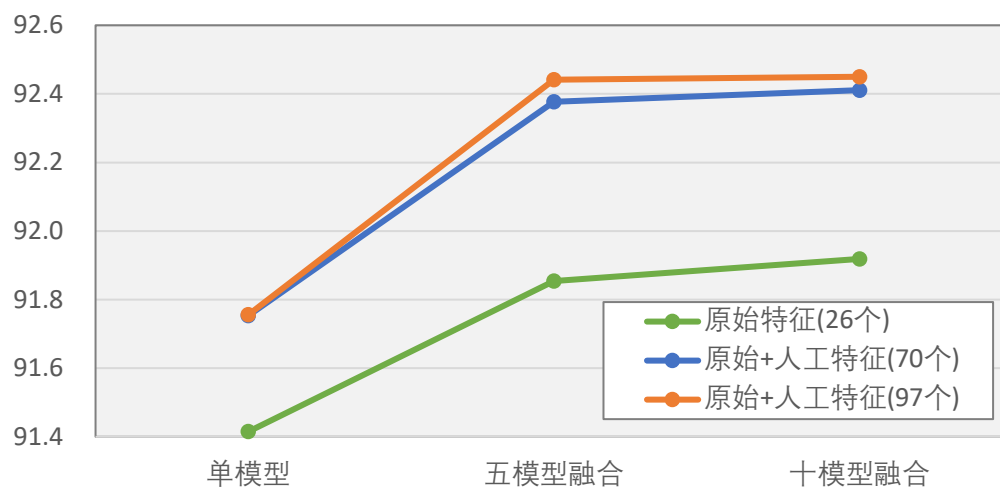


2、通过 K 折交叉验证获得的多模型比简单随机拆分的单模型相比，多模型训练效果更好，通过使用了 5 折交叉验证和 10 折交叉验证验证对比了 5 个模型融合和 10 个模型融合进行训练的结果，本次作业采用了简单易实现的投票方案，当模型较少时，产生不相同结果频率相同而无法正确选择预测结果的情况，因此 10 个模型融合的线下评测脚本得分均高于 5 个模型融合的线下评测脚本。

模型数量与训练得分关系分析图

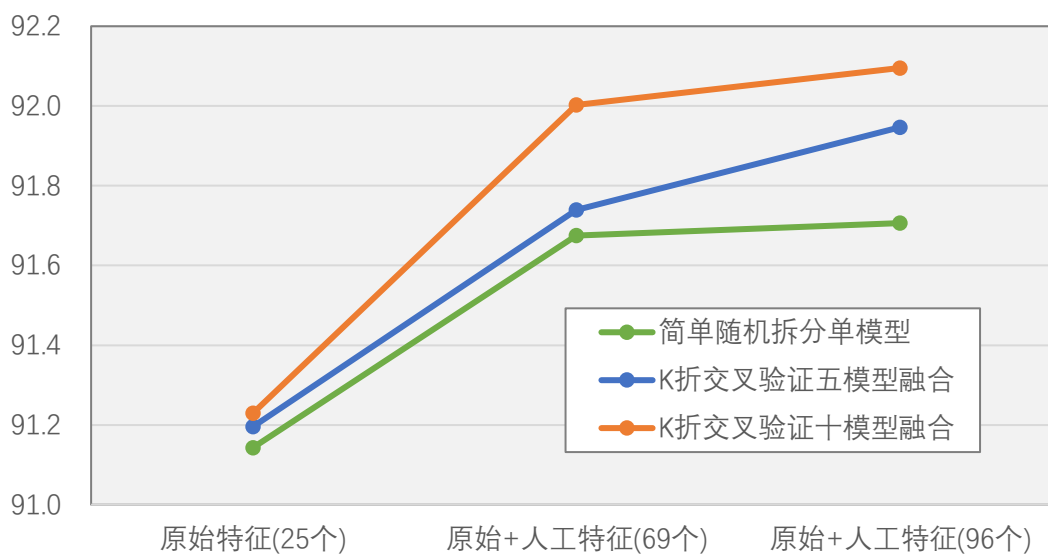


模型数量与线下评测脚本得分关系分析图



3、通过对已有的用户属性(如个人基本信息、用户画像信息等)、终端属性(如终端品牌等)、业务属性、消费习惯及偏好进行分析,合理利用关联规则、业务特征对原始特征进行深入挖掘,建立人工特征,可以有效提高训练准确率,本次作业分别使用了原始特征 (25 个)、原始+人工特征 (69 个) 和原始+人工特征 (96 个) 建立特征矩阵进行训练,发现增加人工特征后训练的模型具有更高的准确率。

工程特征与训练得分关系对比图



特征工程与线下测试脚本得分对比图

