
# 第二届人工智能创新驱动赛（驱动赛道）

# 策划书

| | |
|---|---|
| 队伍名 | GPU8.0 队 |
| 队长 | 邓语苏 |
| 队员 | 李亚轩　汤慧婷 |
| 选题 | A 题 |

# 目录

# 一　选题介绍

## 1.1　题目选择

驱动赛道 A 题：MarTech Challenge 点击反欺诈预测

## 1.2　题目简介

广告欺诈是数字营销需要面临的重要挑战之一，点击会欺诈浪费广告主大量金钱，同时对点击数据会产生误导作用。本次比赛提供了约 50 万次点击数据。特别注意：我们对数据进行了模拟生成，对某些特征含义进行了隐藏，并进行了脱敏处理。

MarTech 技术已经被广泛应用于商业广告分析与挖掘中，在搜索广告，信息流广告，营销预测，反欺诈发现，商品购买预测，智能创意生成中有广泛的应用。三个 Track 分别结合 MarTech 领域中的 3 个重要环节展开，分别包括用户购买预测（商业决策），点击反欺诈（风险控制）以及智能创意生成（创意物料）。

点击欺诈预测适用于各种信息流广告投放，banner 广告投放，以及百度网盟平台，帮助商家鉴别点击欺诈，锁定精准真实用户。测试集中提供了会话 sid 及该会话的各维度特征值，基于训练集得出的模型进行预测，判断该会话 sid 是否为作弊行为。

## 1.3　数据集简介

在本题目中提供了训练集 train.csv 文件以及测试集 test1.csv 文件，数据集中提供了会话 sid 以及基于会话的各维度的特征值。训练集数据为 50 万条，测试集数据为 15 万条，数据较大数据量的数据集。

具体的数据集维度如下：

| | A | B | C |
|---|---|---|---|
| 1 | 字段 | 类型 | 说明 |
| 2 | sid | string | 样本id/请求会话sid |
| 3 | package | string | 媒体信息，包名（已加密） |
| 4 | version | string | 媒体信息，app版本 |
| 5 | android_id | string | 媒体信息，对外广告位ID（已加密） |
| 6 | media_id | string | 媒体信息，对外媒体ID（已加密） |
| 7 | apptype | int | 媒体信息，app所属分类 |
| 8 | timestamp | bigint | 请求到达服务时间，单位ms |
| 9 | location | int | 用户地理位置编码（精确到城市） |
| 10 | fea_hash | int | 用户特征编码（具体物理含义略去） |

图 1: 数据集

| | | | |
|---|---|---|---|
| 11 | fea1_hash | int | 用户特征编码（具体物理含义略去） |
| 12 | cus_type | int | 用户特征编码（具体物理含义略去） |
| 13 | ntt | int | 网络类型 0-未知, 1-有线网, 2-WIFI, 3-蜂窝网络未知, 4-2G, 5-3G, 6–4G |
| 14 | carrier | string | 设备使用的运营商 0-未知, 46000-移动, 46001-联通, 46003-电信 |
| 15 | os | string | 操作系统，默认为android |
| 16 | osv | string | 操作系统版本 |
| 17 | lan | string | 设备采用的语言，默认为中文 |
| 18 | dev_height | int | 设备高 |
| 19 | dev_width | int | 设备宽 |
| 20 | dev_ppi | int | 屏幕分辨率 |

图 2: -续

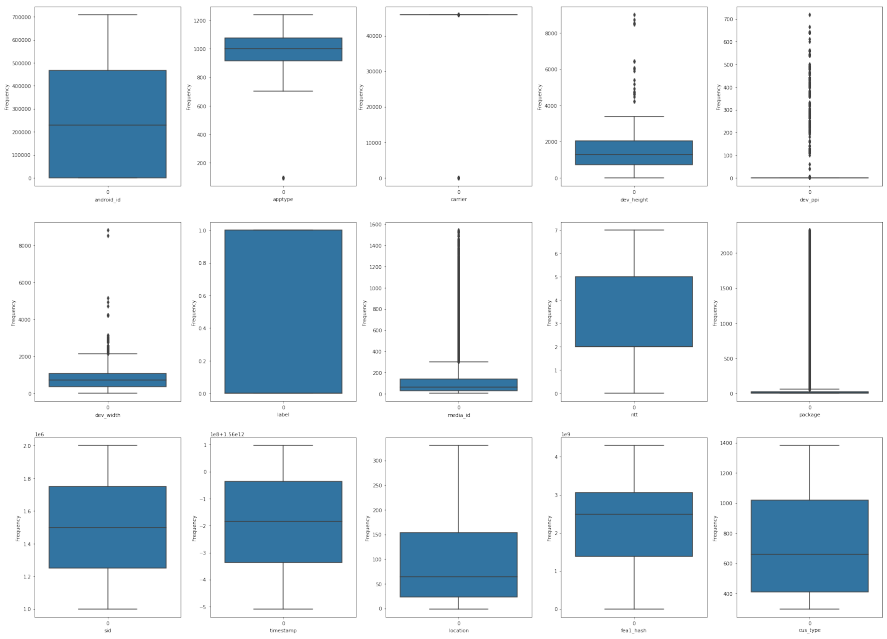
# 二　设计思路

## 2.1　数据集分析

监督式学习的核心是利用已知输入和相应的输出（标签）之间的关系来训练一个模型，以预测新的输入的输出。它的关键在于使用带有标签的训练数据来学习模型的映射函数。这些标签表示了输入数据的真实目标或期望输出。因此对于本题在使用监督式学习的机器学习算法时，输入特征和相应的目标标签通过使用这些已知的输入-输出对，算法通过学习数据中的模式和关联来构建一个模型，而在目的标签已知的情况下，需要尽可能的利用所给的属性值来选择合适的特征。

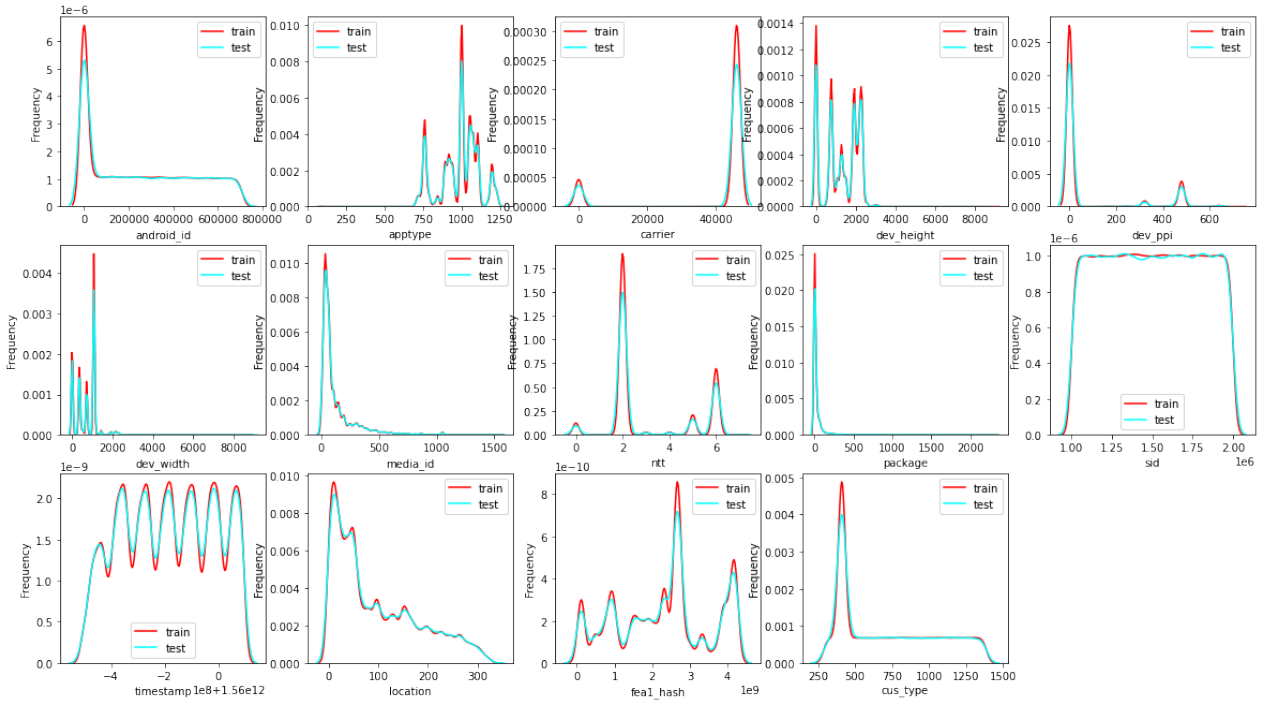
我们可以利用 python 自带的库函数来进行一些初步的数据分析，分析结果如下

1. 对数值型变量绘制箱线图

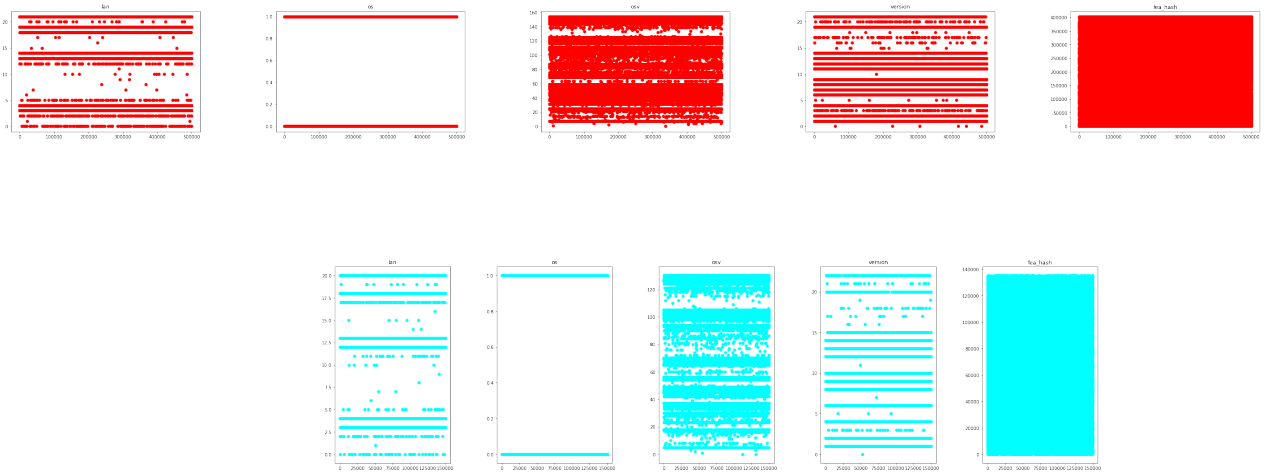


观察图表可以看出 carrier 的取值比较集中，而 dev_ppi 的取值偏于离散，若对 label 值影响不大可以考虑删除。dev_height 和 dev_width 中包含了一定的异常值，需要进行一定的数据预处理。

2. 查看训练集与测试集数值变量分布

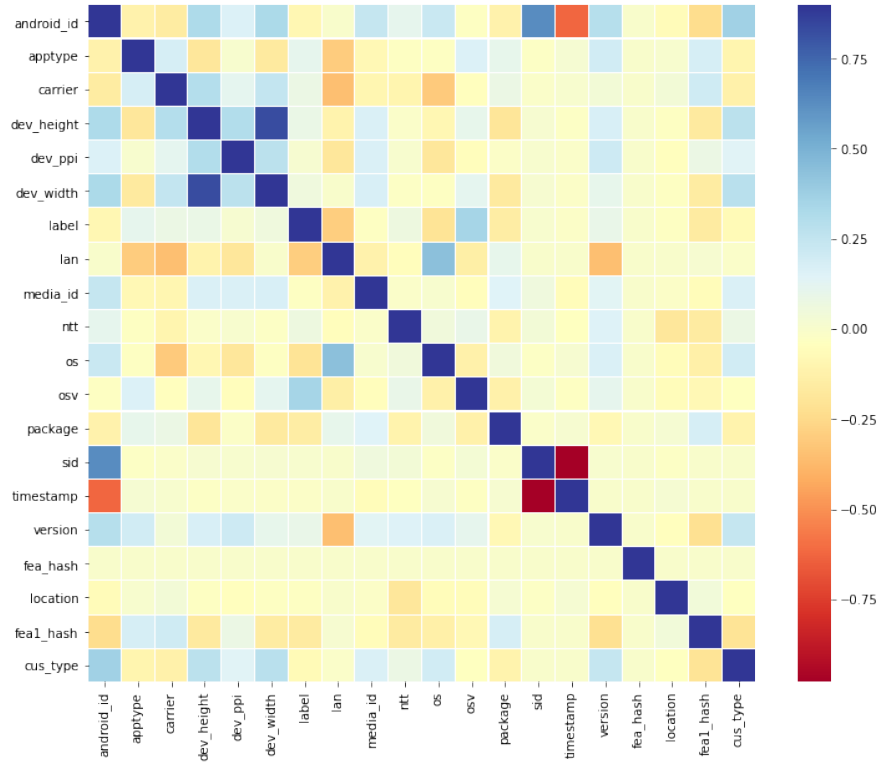



3. 查看分类变量的分布



可以看出 osv 取值单一，对 label 值的预测没有贡献，可以考虑删除。fea_hash 多为唯一值，可以考虑截断其长度。

4. 绘制协方差矩阵热力图




## 2.2 模型选择

本题要求基于训练集得出的模型进行预测，并判断测试集中的会话 sid 是否为作弊行为，这种预测实际上是一种二分类的任务。而在机器学习的各种算法中，有很多算法都能进行二分类的处理，如支持向量机 (SVM),K 最近邻算法 (K-Nearest Neighbors，KNN), 逻辑回归等，集成学习算法如随机森林算法，XGBoost 算法,CatBoost 法,LightGBM 算法等，而其中 CatBoost 和 XGBoost、LightGBM 是基于 GBDT 框架下的三种优秀算法。

通过对其原理分析，我们可以发现 GBDT 框架下的集成算法较为适合用于解决此题，因其准确率相对其余的机器学习算法而言更加准确，且本题中的特征数量较多，而基于 GBDT 框架下的集成算法对于多特征的处理具有较大的优势。**因此对于本题，我们选择使用 GBDT 框架下的集成算法来进行求解，即将 CatBoost 和 XGBoost、LightGBM 三种算法用投票方法进行集成以进一步提升模型的准确率**。

## 2.3 模型评价

评估指标 Accuracy 如下

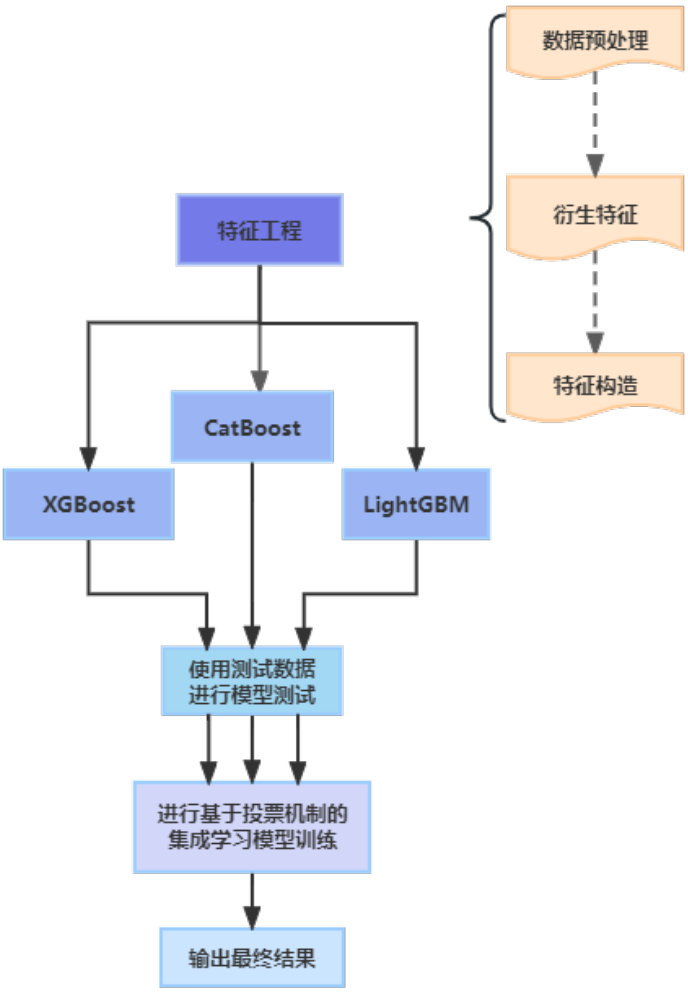
| 实际值 \ 预测值 | 1 | 0 |
|---|---|---|
| 1 | 真正（TP） | 假负（FN） |
| 0 | 假正（FP） | 真负（TN） |

$$acc = \frac{TP + TN}{TP + FP + FN + TN}$$

则当真正 (TP) 的样本数量以及真负 (TN) 的样本数量占总样本数量的比例越高，准确率越高，模型的表现越好。

# 三　技术路径

## 3.1　技术路径简介




### 3.1.1　环境配置

- paddle 框架 2.2.2
- python3.7

## 3.2 特征工程

- **CatBoost 部分**

  类似于 lgb 创建新特征、转换数值特征，如果值为非数字类型，则将其设置为 0，否则将其转换为整数。

  处理时间戳，根据'time' 列的不同格式，计算每个格式的出现频率，并将频率作为新的特征列添加进数据。再进行频率编码，将指定列的值转换为它们在数据集中出现的频率。**频率编码**适用于将分类特征转换为数值特征。

  进行交叉的唯一值数量的 **nunique 编码**。它计算指定列与其他列的交叉组合中的唯一值数量，并将结果作为新的特征添加进数据。

  最后将制定类别的变量进行两两组合，生成**交叉特征编码**。

- **LightGBM 部分**

  为了提供更丰富和有用的信息，我们选择对原始特征进行变换、组合或衍生创建新特征。

```
data['size'] = (np.sqrt(data['dev_height']**2 + data['dev_width'] ** 2) / 2.54) / 1000
#计算并创建名为"size"的新列。它根据"dev_height"和"dev_width"两列的值计算出设备的对角线尺寸，并转换为以米为单位。
data['ratio'] = data['dev_height'] / data['dev_width']
#计算并创建名为"ratio"的新列。它根据"dev_height"和"dev_width"两列的值计算出设备的高宽比。
data['px'] = data['dev_ppi'] * data['size']
#计算并创建名为"px"的新列。它根据"dev_ppi"和"size"两列的值计算出设备的像素数量。
data['mj'] = data['dev_height'] * data['dev_width']
#建名为"mj"的新列。它根据"dev_height"和"dev_width"两列的值计算出设备的面积。
```

  首先将字符串类型特征列转换为整数类型，方便后续的模型训练和处理，对特征值进行映射编码，同时**对缺失值使用进行填充**。

  计算每个类别特征的**计数编码**。通过计算每个类别出现的频次，将其转化为数值型特征，提供给模型使用。再创建交叉特征，其中包括对类别特征的计数统计和计算比例偏好。

  最后为数据集**添加目标特征的滑动平均值**，并进行数据划分、从数据集中提取特征并准备训练集和测试集的数据。

- **XGBoost 部分**

  首先移除了无关的列和索引列，将剩余的列作为特征。

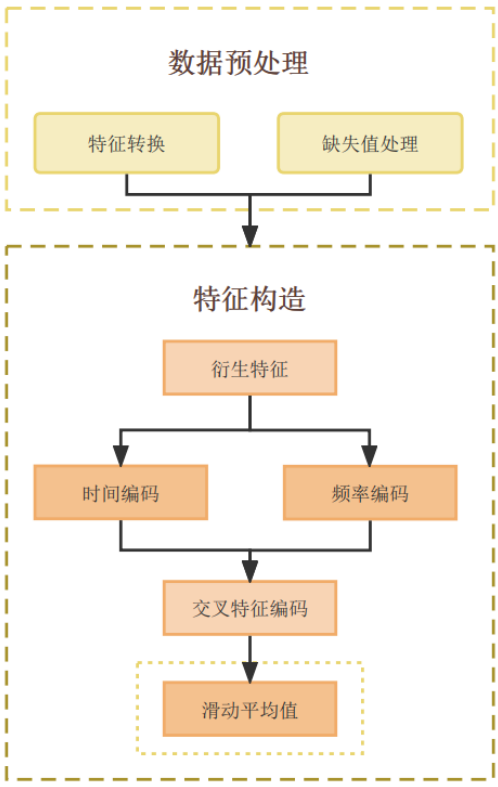
  对于训练集 train 以及测试集 test，在原有的 19 个属性并非都为特征。如下图中所示的 os 属性，通过观察两个数据集，可以发现该属性不存在空缺值，且在两个数据集中该属性的值均为 Android 或为 android，则很明显该属性不能够作为一个特征值来看待，而是需要将其剔除后再进行进一步的处理。

| os |
| --- |
| Android |
| Android |
| android |
| android |
| Android |
| Android |
| Android |

类似于 lgb 处理了缺失值并构造了新的特征和将不同取值映射为数字编码。

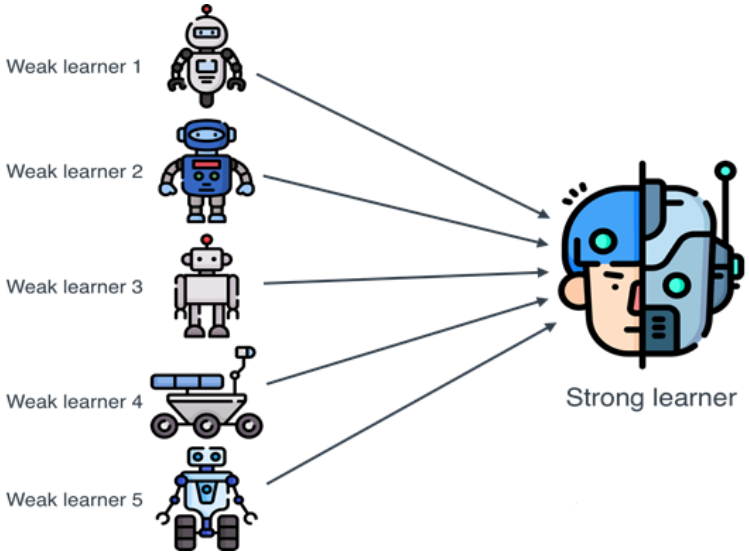
最后将**时间戳**转换为正确的时间格式，并提取了年份、月份、日期、星期几、小时和分钟的信息，将这些信息存储在相应的新列中。

总的来说特征工程涉及了**缺失值处理**、**特征构造**、**特征转换和衍生特征**等操作，以提取数据中的有用信息，为后续的建模任务做准备。



### 3.3 集成模型训练

**集成模型（Ensemble Model）**是指通过将多个独立模型的预测结果进行结合，以达到更好的整体性能的一种机器学习方法。集成模型的目标是通过组合多个弱分类器或回归器，来创建一个强大的模型，能够在预测任务中取得更好的结果。



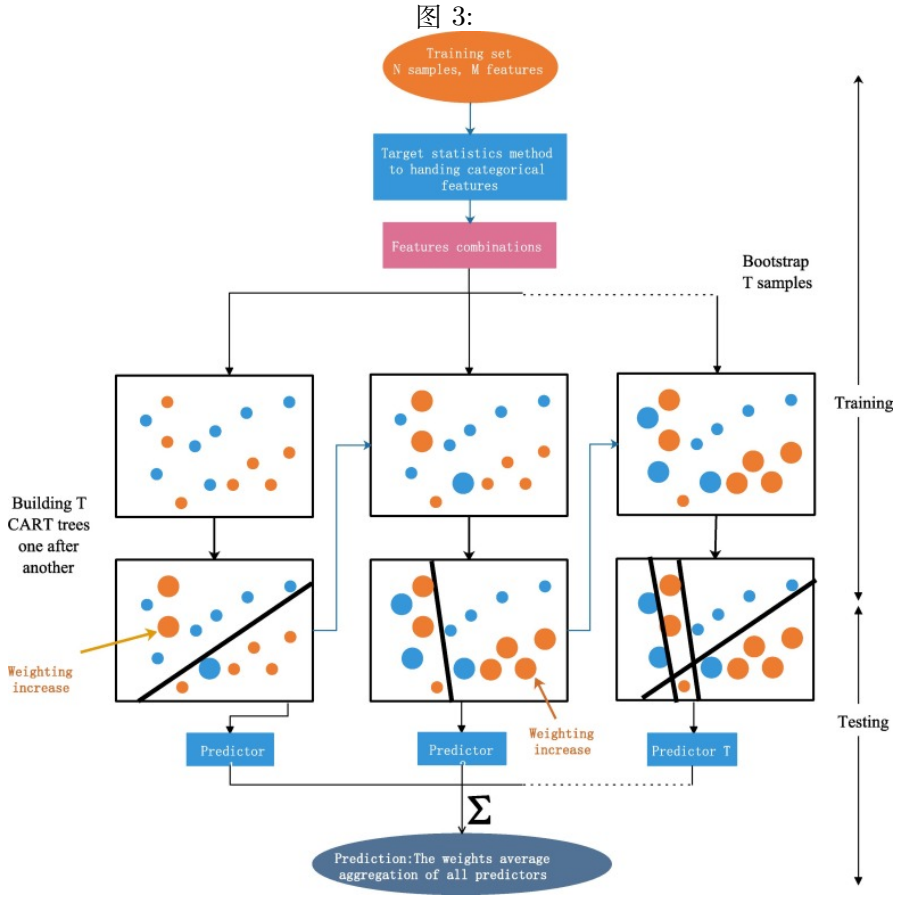

在“2018 创客中国-国家电投大数据及智能应用创新创业大赛”中的光伏发电量预测的高分方案中，其将 XGBoost、LightGBM,LSTM 进行模型融合。在参赛过程中的提交中发现，树模型（XGBoost 和 LightGBM）以及 LSTM 单模型的学习能力都较强，在对几个模型进行线性融合之后，预测能力进一步增强。融合模型取得了了比赛中最好的成绩。受到光伏发电量预测的高分方案的启发。我们选取 XGBoost、CatBoost、LightGBM 进行基于投票方法的集成模型训练。

#### 3.3.1 CatBoost 介绍

**CatBoost（Categorical Boosting）**是一种梯度提升框架，用于解决分类和回归问题。它在处理分类变量（离散特征）方面具有出色的性能，并且能够处理数值特征。CatBoost 使用了一种特殊的技术，称为有序目标转换（Ordered Target Transformation），来处理分类特征。它将分类特征的值排序，并使用目标变量的统计信息（例如平均目标值）来替代原始值。这种转换使得分类特征能够直接参与梯度提升过程，而不需要进行独热编码等昂贵的转换操作。

CatBoost 的核心算法是梯度提升决策树（Gradient Boosting Decision Tree，简称 GBDT）。GBDT 是一种集成学习方法，通过训练一系列决策树模型来逐步提升预测的准确性。

图 3:




CatBoost 在训练过程中采用了类似于其他梯度提升框架的技术，如梯度提升和反向传播。但它引入了两个主要的改进：[1]

1. 对称叶子分数：CatBoost 使用对称梯度来计算叶子分数，而不仅仅使用正向梯度。这样可以更准确地估计叶子的权重，进而提高模型的性能。

2. 随机排列：CatBoost 通过随机排列特征值的顺序来减小过拟合。它会在每次迭代中随机选择一部分特征，并对这些特征的值进行排序。这种随机化可以降低树模型对特定特征排序的依赖性，使得模型更具鲁棒性。

除了上述改进，CatBoost 还提供了其他一些功能，如自动处理缺失值、自动特征缩放、直接支持类别特征等。它还支持多种损失函数和评估指标的选择，以及模型的解释性和可视化。

综上所述，CatBoost 是一种强大的梯度提升框架，通过处理分类特征和引入改进的算法技术，能够在分类问题上取得优秀的性能。

由于 CatBoost 参数较多，运行耗时长，占用内存大，在具体实现时，靠人工试验最优参数是不现实的、性价比低的。可以使用网格搜索 (Grid Search)，或者随机搜索 (Randomized Search) 寻找最优参数。

### 3.3.2 LightGBM 介绍

**LightGBM** 是一个高效的梯度提升决策树（Gradient Boosting Decision Tree）框架，它是由微软开发的机器学习库。它在训练和预测速度上具有优势，并且能够处理大规模数据集。

以下是 LightGBM 实现的一些具体细节：[2]

1. 基于直方图的特征离散化：LightGBM 将连续特征离散化为多个离散的值，然后构建直方图来表示每个特征。这样可以减少内存的使用和计算量，并提高训练速度。

2. 按层生长的决策树：LightGBM 采用了按层生长的策略来构建决策树。在每一层中，它首先选择具有最大增益的分裂特征，然后根据该特征的取值建立子节点。这种策略减少了冗余分裂，提高了树的构建效率。

3. Leaf-wise 生长策略：LightGBM 使用 Leaf-wise 的生长策略，即每次选择当前树中具有最大增益的叶节点进行分裂。这种策略可以更快地增加树的深度，提高模型的学习能力。

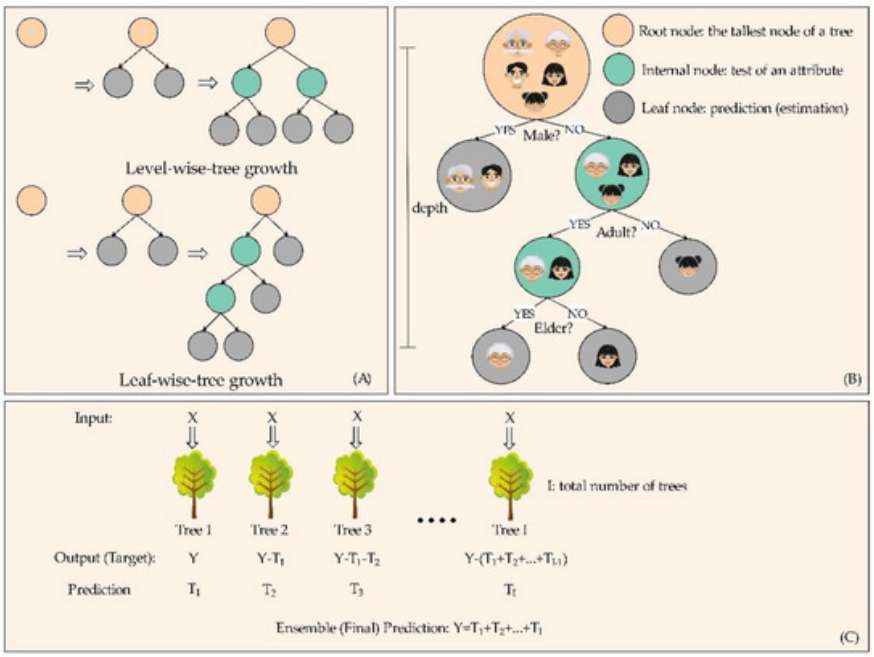
4. 直方图做差加速：LightGBM 使用直方图做差的技术来加速特征的更新。它维护了一个全局直方图和每个叶节点的局部直方图，在计算分裂增益时通过差分操作避免重复计算。

5. 特征并行化：LightGBM 支持特征并行化，可以并行计算不同特征的直方图和分裂增益，加快训练速度。

6. 带深度限制的决策树生长：为了控制模型的复杂度和防止过拟合，LightGBM 引入了最大深度和叶节点数量的限制。

7. 提前停止策略：LightGBM 支持提前停止策略，当验证集上的损失函数不再改善时，可以提前停止训练，防止过拟合。

图 4:



总的来说 LightGBM 采用了基于直方图的算法和高效的并行计算，能够处理大规模数据集和高维特征，具有较快的训练速度和预测速度；同时 LightGBM 使用了基于直方图的数据存储和压缩技术，可以有效地减少内存占用，使得能够处理大规模数据集；最后 LightGBM 提供了特征重要性评估的功能，可以帮助选择对模型影响较大的特征，提高模型的泛化能力。因此运用 LightGBM 框架对本次大规模的数据进行模型训练和预测能得到较好的结果。

### 3.3.3 XGBoost 介绍

**XGBoost (Extreme Gradient Boosting)** 是一种基于梯度提升算法的集成学习模型，被广泛应用于机器学习和数据科学领域。它以高效性能和准确性而闻名，结合了梯度提升算法和决策树模型的优势且 XGBoost 使用了一系列优化技术，如近似算法、稀疏感知算法和并行计算等，使得模型训练和预测速度极快，能够处理大规模数据集。相对于常规的决策树算法，其改进如下：[3]

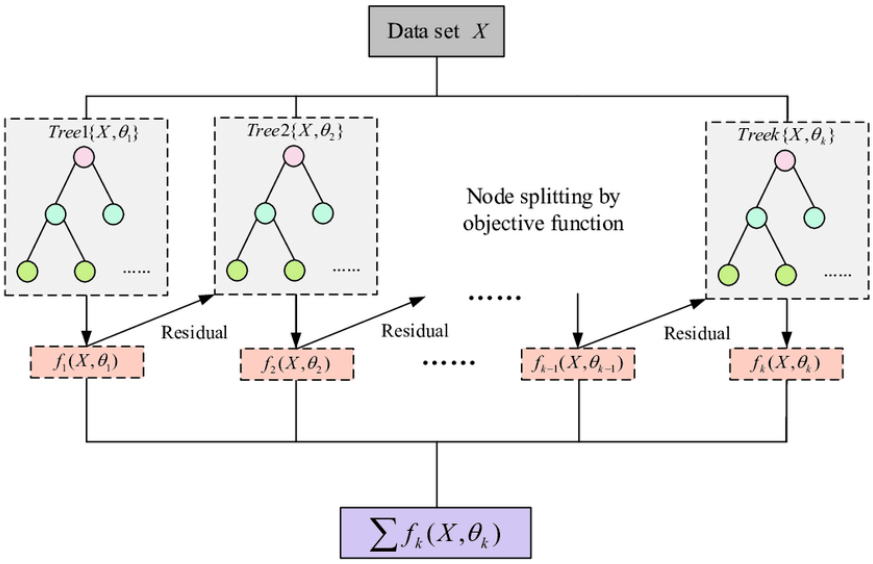
1. 准确性：XGBoost 在损失函数中引入了正则化项和二阶导数信息，能够更好地拟合复杂的数据模式，提供高准确性的预测。

2. 灵活性：XGBoost 支持多种损失函数和目标函数的定义，适用于分类、回归和排名等任务。同时，它提供了丰富的超参数选项，可以灵活调整模型的复杂度和性能。

除了上述改进，XGBoost 还增添了一些其他的功能，如能够自动处理缺失值和稀疏数据，还支持特征重要性评估和特征选择等功能，帮助用户进行特征工程和模型优化，并且在训练过程中采用了正则化和早停策略，可以有效避免过拟合。

而在 XGBoost 的模型训练过程中，其训练的核心是当建立第 t 棵树时，如何找到叶子节点的最优切分点，因为要衡量所建立的树的优劣性，在理论状态下可以列举所有可能的树并挑选其中最优的树。但在现实的训练过程中，由于其实施的复杂性过大，所以这种方法是不可取的，只能尽量一次优化树的一个层次。具体来说是将一片叶子分成两片，并得到分割后的量化分数，找到其最优的分割点以求得最优的树

图 5:



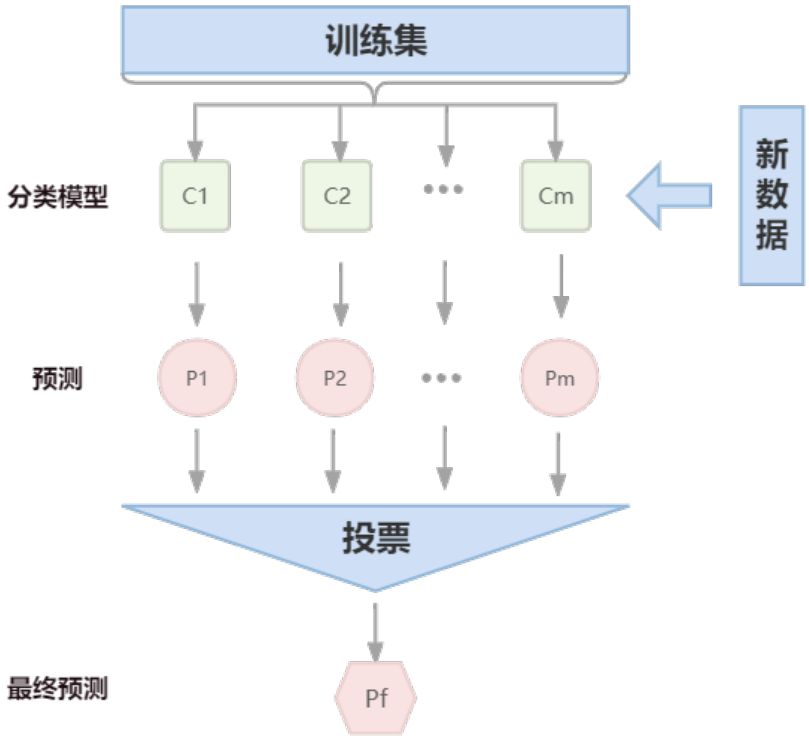
而为了做到有效分割，将所有的实例按照一定顺序排列，然后遍历所有的可能的实例序列计算在不同分割位点的得分从而求得最优的分割位点。

### 3.3.4 集成学习方法–投票

在本模型中，选择使用投票方法中的硬投票进行模型融合。

**硬投票（Hard Voting）**是集成学习 [4] 中一种常见的投票策略，用于集成多个独立模型的预测结果以确定最终的预测结果。在硬投票中，每个独立模型对样本进行预测，并根据多数票的原则来确定最终的预测类别。

图 6:




将 XGBoost、CatBoost 和 LightGBM 这三种模型进行投票改进模型的主要原因是通过集成多个模型的预测结果，可以融合它们的优势，减少个别模型的弱点，从而提高整体模型的性能。以下是一些可能的改进方面：

1. 模型鲁棒性：不同的梯度提升框架在处理数据和学习过程中有着不同的偏好和策略。通过将它们组合在一起，可以减少个别模型的局限性，提高模型的鲁棒性。如果某个模型在某些数据或特征上表现较差，其他模型可以弥补其不足。

2. 偏差-方差权衡：集成多个模型可以在偏差和方差之间进行权衡。每个模型都有其自身的偏差和方差特性。通过集成，可以根据问题的需求和数据集的特点调整不同模型的权重，以获得更好的偏差-方差平衡。

3. 提高泛化能力：通过结合多个独立模型的预测结果，集成模型能够在预测新数据时获得更好的性

能。通过模型之间的协同作用，集成模型能够捕捉更广泛的数据特征，从而提高对未知数据的泛化能力。

4. 增加模型的稳定性：由于集成模型基于多个独立模型的预测结果，它对于数据的变化和噪声具有更好的稳定性。即使某个模型受到干扰或噪声影响，其他模型的预测结果可以提供平衡和修正，从而提高整体模型的稳定性。

# 四　模型优缺点

## 4.1　模型优点

1. 所选用的三种模型 CatBoost、XGBoost 和 LightGBM 都是基于梯度提升决策树（GBDT）算法的框架，GBDT 具有训练效果好、不易过拟合等优点。GBDT 在实际应用中被广泛使用，特别是在分类和回归问题中。它在处理结构化数据和非线性关系方面表现优秀，并且对特征工程的要求相对较低。

2. 选用三个框架都是 GBDT 算法的优秀实现，它们在模型性能、速度和功能方面都有不同的优势和特点。其中 CatBoost 有很好的处理大量类别特征的能力，避免了用独热编码可能造成的维度灾难和大量计算资源的消耗，更适用于本题。

3. 使用了集成模型，通过将多个独立模型的预测结果进行结合，可以融合它们的优势，抵消个别模型的弱点，增加模型的鲁棒性，提高了泛化能力，从而提高整体模型的性能。

## 4.2　模型缺点

1. 集成模型方法使用的是粗糙的硬投票方法，模型融合的最优系数未知。

2. 模型可解释性较差

# 参考文献

[1] Anna Veronika Dorogush, Andrey Gulin, Gleb Gusev, Nikita Kazeev, Liudmila Ostroumova Prokhorenkova, and Aleksandr Vorobev. Fighting biases with dynamic boosting. *CoRR*, abs/1706.09516, 2017.

[2] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30, 2017.

[3] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. *CoRR*, abs/1603.02754, 2016.

[4] Martin Sewell. Ensemble learning. *RN*, 11(02):1–34, 2008.