1 定义

1.1 项目概述

Rossmann销量预测题目源自Kaggle比赛，Rossmann是欧洲一家连锁药店，比赛目标是根据Rossmann的商店信息如促销信息，竞争对手信息等以及历史销售情况，来预测Rossmann 未来的销售额。　　销量预测作为现代商业重点关注问题之一，是企业在库存、物流、市场营销、资金管 理等多方面的决策依据，因此销量预测的准确性对企业至关重要。销量预测业界主要有三 种销量预测方法：专家预测法、时间序列法、基于机器学习的预测方法 (1) 。为了达到良好 的预测准确度，本文重点关注基于机器学习的销量预测方法，即使用有监督回归模型解决 销量预测问题。

1.2 问题陈述

基于Rossmann的1115家店铺的历史销售数据以及店铺信息，对1115家店铺未来销售额的进行预测，确保预测值尽可能接近未来真实值。基于历史数据的销量预测属于有监督学习中回归问题。
本项目基于历史销量数据以及店铺信息进行数据分析和特征处理，然后选择合适的机器学习算法基于训练数据进行回归建模，通过回归模型评价指标评估模型质量并进行参数或特征调整以提升预测准确度，最终使用优化后的模型对未来Rossmann店铺的销售额度进行预测。

1.3 评价指标

回归模型的评价指标一般有MSE、MAE、R-squared、RSME、RMSPE等，本项目选择RMSPE作为评价指标，RMSPE 的定义如下 (2) ：

$$RMSPE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i - \hat{y}_i}{y_i}\right)^2}$$

其中 y_i 是实际值，y_i-hat 是模型预测值,RMSPE与R-squared相比更能代表预测值相对实际值的误差概念，与RSME相比，RSME体现了所有预测点的平均误差，对预测异常点敏感，而RMSPE计算的是误差率，避免了真实值大小不同对误差率产生的影响。因此本项目计划选择RMSPE作为模型的评价指标，Kaggle官方也使用此指标作为评估标准。

2 分析

2.1 数据探索

2.1.1 数据集概述

本项目包含以下数据集，数据集源于Kaggle官网[3]：

1. train.csv：Rossmann1115家店铺从2013年1月到2015年7月的营业情况，包含每家店铺每天的销售额，以Sales字段表示。此数据集可基于交叉验证拆分为训练集和验证集数据，用于模型的训练。此数据集共包含 1017209 条数据，9 个数据维度，详情如下：
    - Store - 店铺标识，数据格式为整型，无数据缺失。
    - DayOfWeek - 当前日期是本周第几天，数据格式为整型，无数据缺失。
    - Date - 当前日期，精确到日，数据格式为字符串型，无数据缺失。
    - Sales - 某店铺当前日期的销量值，数据格式为整型，无数据缺失。
    - Customers - 某店铺当前日期的客户数量，数据格式为整型，无数据缺失。
    - Open - 某店铺当前日期是否开业标识，数据格式为整型，无数据缺失。
    - Promo - 某店铺当前日期是否促销标识，数据格式为整型，无数据缺失。
    - StateHoliday - 当前日期是否为某类型StateHoliday标识，数据格式为字符串型， 无数据缺失。
    - SchoolHoliday --当前日期是否为SchoolHoliday标识，数据格式为整型，无数据 缺失。
2. test.csv ： Rossmannde1115家店铺在 2015年8月至9月的营业情况，和train.csv相比，不包含每家店铺每天的销售额和顾客数量，销售额Sales是模型的预测字段，和train.csv相比新增的ID字段用于Kaggle上预测结果提交。此数据集可以用于最终训练好的模型预测。
3. store.csv：Rossmannde1115家店铺的补充信息，如竞争对手情况、促销活动情况等。此数据集可以和train.csv以及 test.csv 分别合并，用于提取模型训练或和预测的特征。
    - Store - 店铺标识，数据格式为整型，无数据缺失。
    - StoreType - 店铺类型，数据格式为字符串型，无数据缺失。
    - Assortment - 店铺某种分类类型，数据格式为字符串型，无数据缺失。
    - CompetitionDistance - 与竞品店铺间隔距离，数据格式为字符型，有数据缺 失。
    - CompetitionOpenSinceMonth - 竞品店铺开业月份，数据格式为字符型，有数 据缺失。
    - CompetitionOpenSinceYear - 竞品店铺开业年份，数据格式为字符型，有数据 缺失。
    - Promo2 - 店铺是否持续促销的标识，数据格式为整型，有数据缺失。
    - Promo2SinceWeek - 店铺持续促销开始日期（当年第几周），数据格式为浮点 型，有数据缺失。
    - Promo2SinceYear - 店铺持续促销开始年份，数据格式为浮点型，有数据缺 失。
    - PromoInterval - 店铺持续促销的间隔月份，数据格式为字符串型，有数据缺 失。

2.1.2 数据可视化

1. 数据的分布
    - 通过数值型数据分布情况看出Sales值主要集中在500-1000之间，Sales、Customer 存在很多 0 值，可能和Open字段有关，也可能是异常值，预处理中需要注意。CompetitionDistance分布非正态，中位数和均值相差较大，在数据预处理中根据需要进行特征缩放；

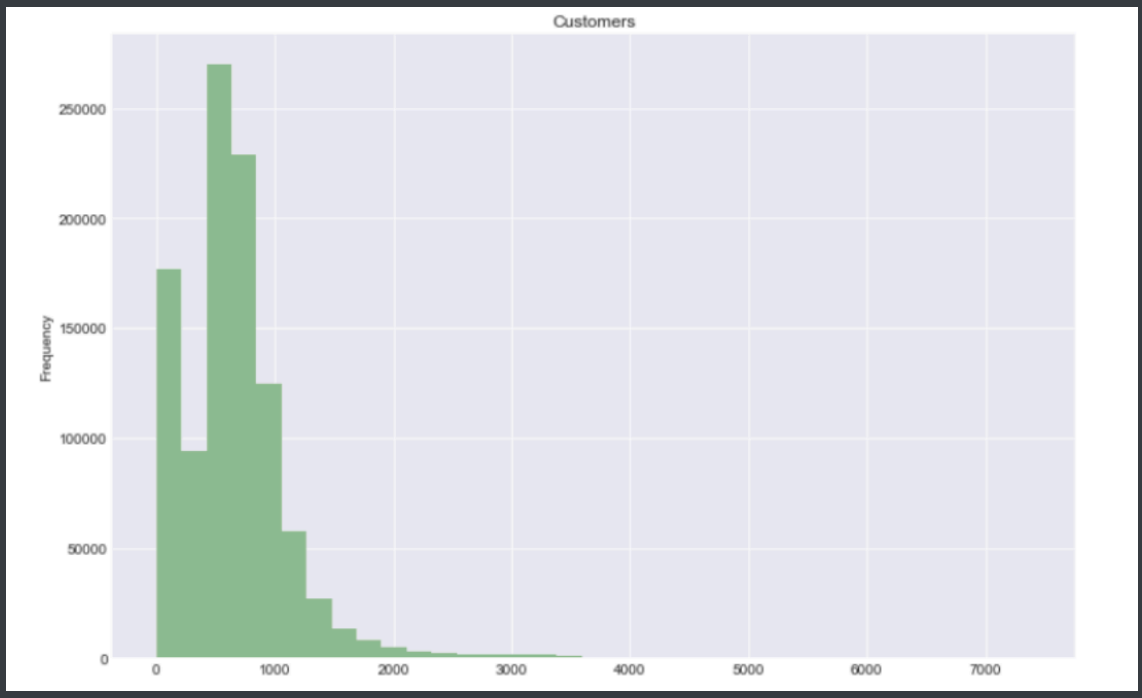


图1 Custormers分布直方图

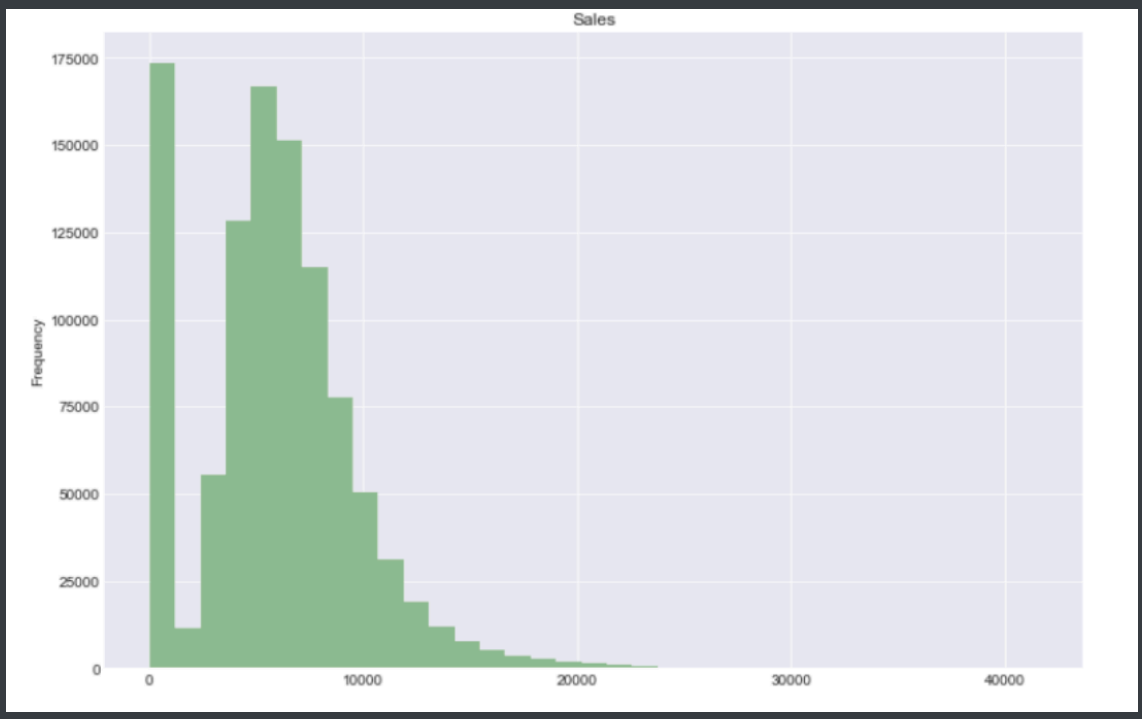


图2 Sales分布直方图

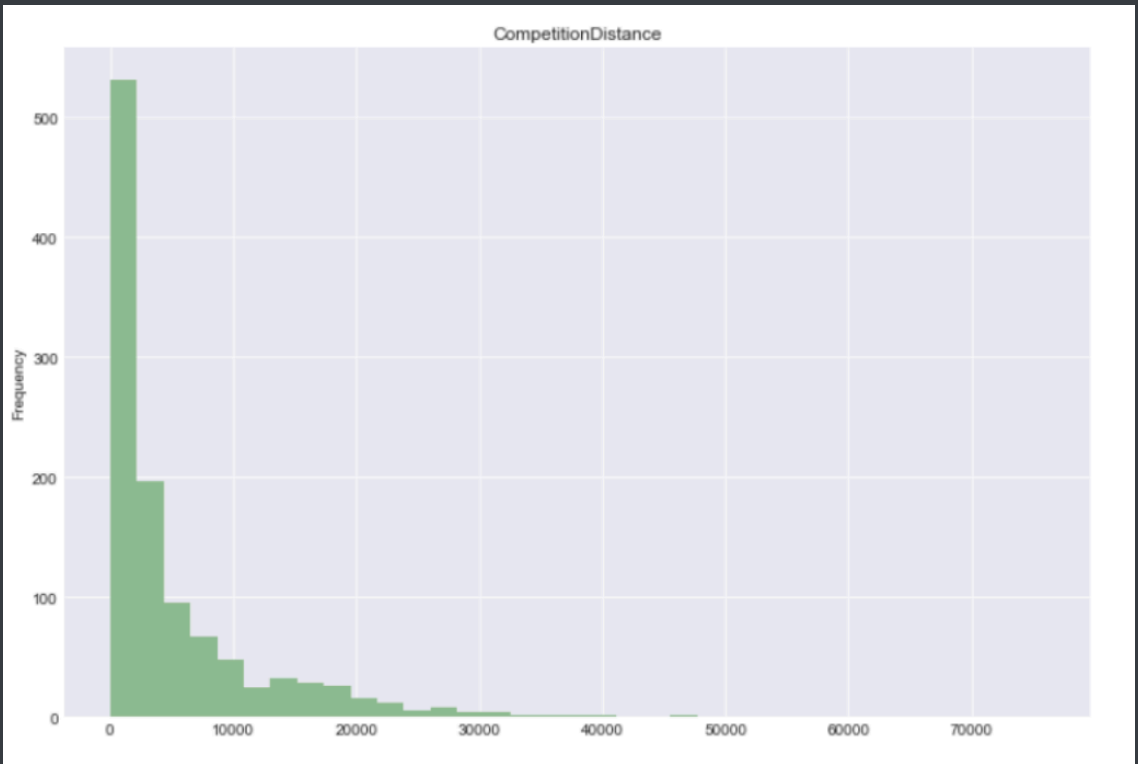


图3 CompetitionDistance分布直方图

- 发现商店大多数在周一至周六开业，其中周六有一小部分商店停业，绝大多数商店在周末不开业。

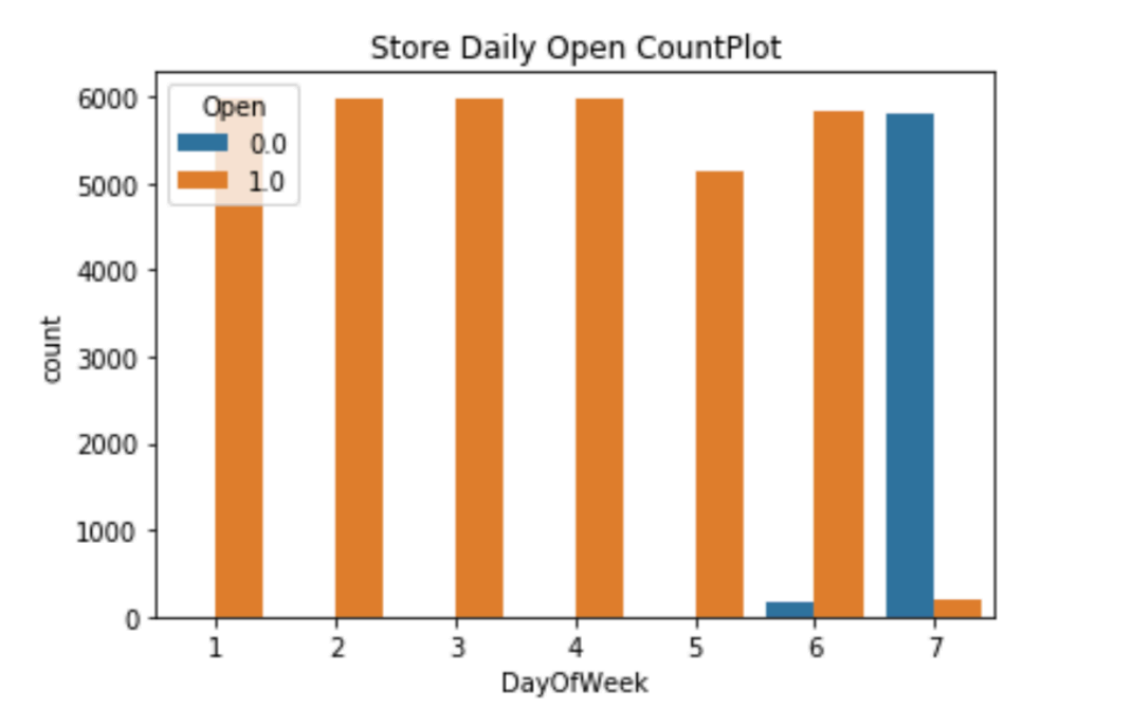


图4 商店一个星期内开业分布直方图

- 通过非数值型分布分析看出train.csv中Date时间范围是从2013年1月到2015 年 7 月，后续将Date分割为年、月、日的数值型数据。StateHoliday字段应该有4个分类，分布统计显示5个，在预处理时需要对两个0进行合并。分类类型的非数值型字段在预处理中根据需求通过one-hot编码转化为数值型数据。
- 通过分析销售额的skewness，发现原始的sales没有呈现出正太分布，其skewness为1.60。然后对其进行log(1+sales)处理之后，数据呈现出了正太分布，其skewness为0.11。

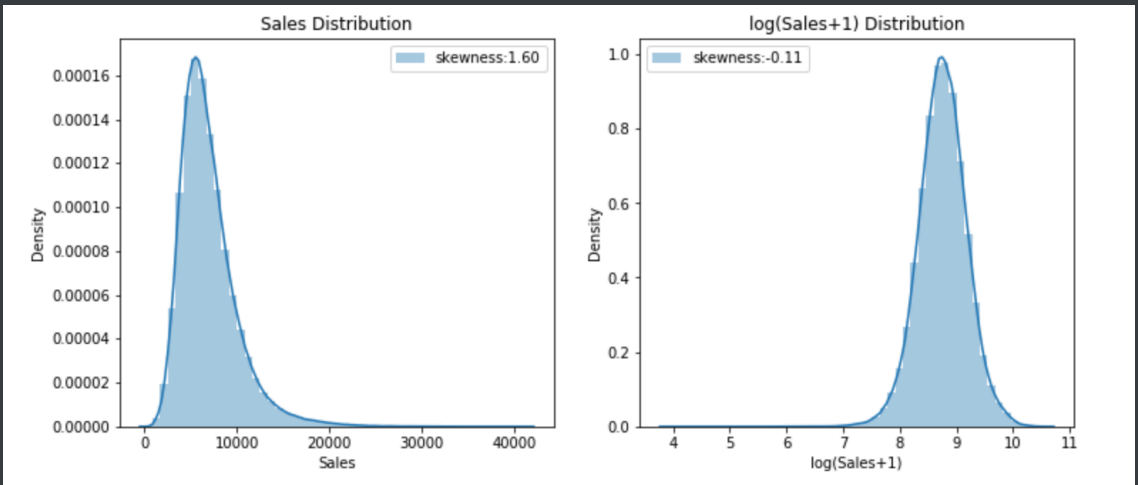


图5 Sales skewness图

2. 数据相关性

- 通过对数据进行简单的相关性分析，可以看出，Open、Promo、Customer三个维度与Sales的相关性较大，符合通常认知，即店铺是否营业，店铺是否开展营销活动，到店顾客数对店铺销量影响较大。

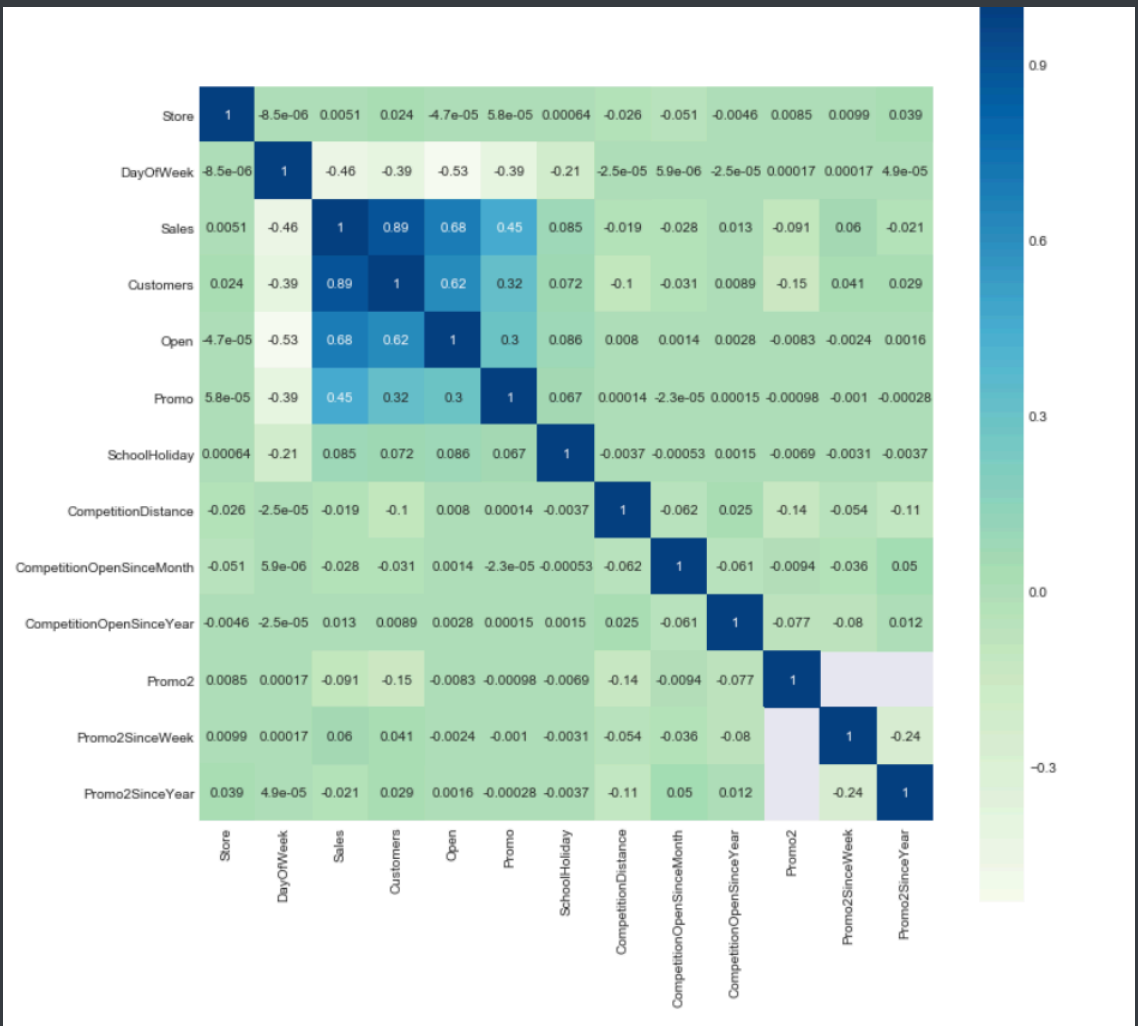


图6 相关系数矩阵

2.2 算法和技术

2.2.1 回归算法

销量预测算法的本质是建立回归模型，目前主流回归算法可以分为三类：线性/多项式回归、决策树、神经网络等，其中线性/多项式回归方法建模速度快，可解释性强，但难以良好表达高度复杂数据，回归树/回归森林建模效果好，可解释性强，但是容易导致过拟合，神经网络可以表达复杂的非线性关系，且不需要过多关注特征工程，但计算量大、可解释性差、不适合小规模数据[4]。 **线性/多项式回归**：线性回归是利用称为线性回归方程的最小平方函数对一个或多个自变量和因变量之间关系进行建模的一种回归分析。非线性的多项式回归则将输入变量进行一系列非线性组合以建立与输出之间的关系。训练回归算法模型一般使用随机梯度下降法。线性回归以及多项式回归建模迅速，且可解释性强，在金融风控、征信等直接面向客户领域有广泛应用，但是不适合数据特征间具有相关性或高度复杂的数据场景。 **神经网络**：由一系列神经元节点分层次连接而成，数据的特征通过输入层被逐级传递到网络中，形成多个特征的线性组合，每个特征会与网络中的权重相互作用。随后神经元对线性组合进行非线性变化，这使得神经网络模型具有对多特征复杂的非线性表征能力，神经网络通常使用随机梯度下降法和反向传播法训练，数据越多，训练结果越好。神经网络不需要我们进行复杂的特征工程，但是训练过程计算量大，模型可解释性差。 **回归树/回归森林**：决策树是通过遍历树的分支并根据节点的决策选择下一个分支的模型。树型感知利用训练数据作为数据，根据最适合的特征进行拆分，并不断进行循环指导训练数据被分到一类中去。建立树的过程中需要将分离建立在最纯粹的子节点上，从而在分离特征的情况下保持分离数目尽可能的小。纯粹性是来源于信息增益的概念，它表示对于一个未曾谋面的样本需要多大的信息量才能将它正确的分类。实际上通过比较熵或者分类所需信息的数量来定义。而随机森林则是决策树的简单集合，输入矢量通过多个决策树的处理，最终的对于回归需要对输出数据取平均、对于分类则引入投票机制来决定分类结果。决策树可以拟合复杂的非线性关系，模型易理解，但是决策树很容易过拟合。

2.2.2 技术

本项目选用XGBoost算法进行模型训练。XGBoost是2014年陈天奇博士发布的梯度提升算法机器学习函数库，自诞生后因为其优良的学习效果以及高效的训练速度而获得广泛的关注。 XGBoost是GBDT（Gradient boosting Decision Tree）算法的扩展和改进。boosting 思想是迭代生多个弱模型，然后将每个弱模型的预测结果相加生成强模型，Gradient Boosting是一种实现Boosting的方法，即每次建立模型，是在上一次建立模型损失函数的梯度下降方向。GBDT 是基于CART回归树的Gradient Boosting算法，在GBDT的迭代中，假设前一轮迭代得到的强学习器是 ft−1(x), 损失函数是L(y,ft−1(x)), 本轮迭代的目标是找到一个CART回归树模型的弱学习器 ht(x)，让本轮的损失函数L(y,ft(x)=L(y,ft−1(x)+ht(x))最小，本轮损失的用损失函数的负梯度来拟合[5]。XGBoost是GB思想的高效实现，和GBDT的主要区别有：

- XGBoost中基学习器可以是CART或线性分类器。
- XGBoost在目标函数中添加正则化项，用于控制模型复杂度，训练出模型更加简 单，防止过拟合。
- 传统的GBDT在优化的时候只用到一阶导数信息，XGBoost则对损失函数进行了二阶泰勒展开，将损失函数从平方损失推广到二阶可导的损失。

XGBoost的原理

XGB的目标函数如下：

$$Obj^{(t)} = \sum_{i=1}^{n} l(y_i, \hat{y}_i^{(t-1)} + f_t(x_t)) + \Omega(f_t) + constant$$

用泰勒展开来近似原来的目标

同时定义：$g_i = \partial \hat{y}^{(t-1)} l(y_i, \hat{y}_i^{(t-1)}), h_i = \partial^2 \hat{y}^{(t-1)} l(y_i, \hat{y}_i^{(t-1)})$

则目标函数变为如下

$$Obj^{(t)} \approx \sum_{i=1}^{n} [l(y_i, \hat{y}_i^{(t-1)} + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)] + \Omega(f_t) + constant$$

对于f的定义做细化，把树拆分为结构部分q和叶子权重部分w。结构函数q把输入映射到叶子的索引号上面去，而w给定了每个索引号对应的叶子分数是什么。

$$f_t(x) = \omega_q(x), \omega \in R^T, q : R^d \to \{1, 2, 3, \ldots, T\}$$

定义这个复杂度包含了一棵树里面节点的个数，以及每个树叶子节点上面输出分数的L2模平方。

$$\Omega(f_t) = \gamma T + \frac{1}{2} \lambda \sum_{j=1}^{T} \omega_j^2$$

在这种新的定义下，我们可以把目标函数进行如下改写，其中I被定义为每个叶子上面样本集合 $I_j = \{i | q(x_i) = j\}$,g是一阶导数，h是二阶导数。

$$Obj^{(t)} \approx \sum_{j=1}^{T} [(\sum_{i \in I_j} g_i) \omega i + \frac{1}{2} (\sum_{i \in I_j} h_i + \lambda) \omega_j^2] + \gamma T$$

这一个目标函数办好了T个相互独立的单变量二次函数。我们可以定义

$$G_j = \sum_{i \in I_j} g_i, H_j = \sum_{i \in I_j} h_i$$

最终的公式可以简化为

$$Obj^{(t)} \approx \sum_{j=1}^{T} [G_j \omega_j + \frac{1}{2} (H_j + \lambda) \omega_j^2] + \gamma T$$

通过对ω_j求导等于0，可以得到

$$\omega_j^* = -\frac{G_j}{H_j + \lambda}$$

然后把ω_j代入得到最优解：

$$Obj^{(t)} \approx -\frac{1}{2} \sum_{j=1}^{T} \frac{G_j^2}{H_j + \lambda} + \gamma T$$

XGBoost中比较重要的参数如下：

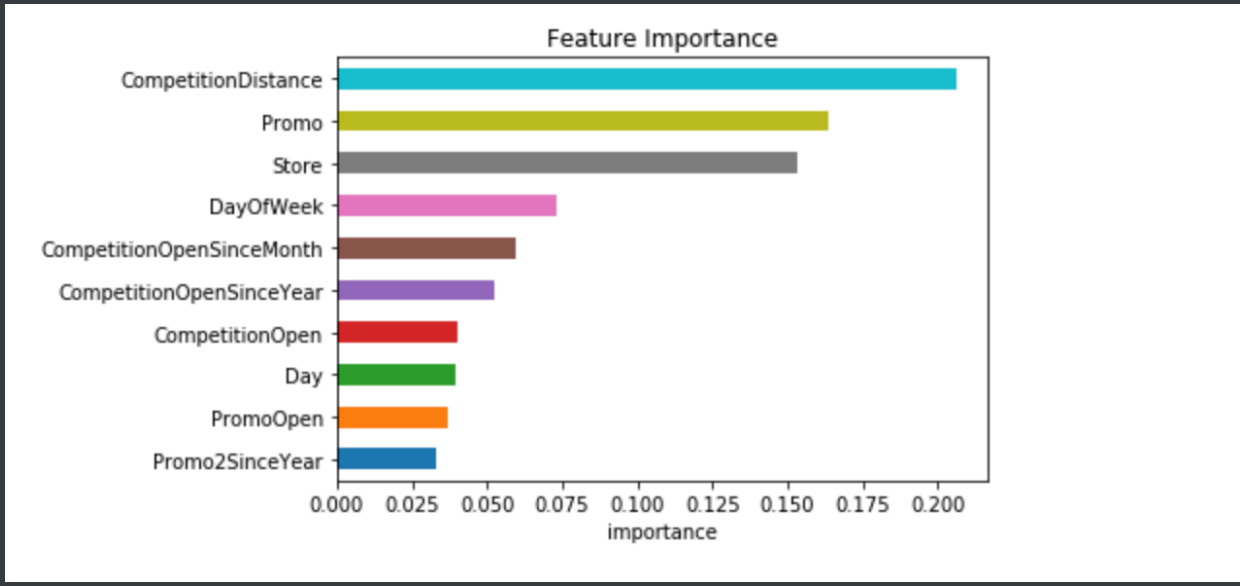
1. objective [default=reg:linear] 定义学习任务及相应的学习目标，可选的目标函数如下：
    - “reg:linear” - 线性回归
    - reg:logistic” - 逻辑回归
    - “binary:logistic” - 二分类的逻辑回归问题，输出为概率
    - binary:logitraw” - 二分类的逻辑回归问题，输出的结果为wTx
    - “count:poisson” - 计数问题的poisson回归，输出结果为poisson分布。 在poisson回归中，max_delta_step的缺省值为0.7。(used to safeguard optimization)
    - “multi:softmax” - 让XGBoost采用softmax目标函数处理多分类问题，同时需要设置参数num_class（类别个数）
    - multi:softprob” - 和softmax一样，但是输出的是ndata * nclass的向量，可以将该向量reshape成ndata行nclass列的矩阵。没行数据表示样本所属于每个类别的概率
    - “rank:pairwise” - set XGBoost to do ranking task by minimizing the pairwise loss
2. 'eval_metric' The choices are listed below，评估指标:
    - "rmse": root mean square error
    - "logloss" negative log-likelihood
    - "error": Binary classification error rate. It is calculated as #(wrong cases)/#(all cases). For the predictions, the evaluation will regard the instances with prediction value larger than 0.5 as positive instances, and the others as negative instances.
    - "merror": Multiclass classification error rate. It is calculated as #(wrong cases)/#(all cases).
    - "mlogloss": Multiclass logloss
    - "auc": Area under the curve for ranking evaluation.
    - "ndcg": Normalized Discounted Cumulative Gain
    - "map": Mean average precision
    - "ndcg@n", "map@n": n can be assigned as an integer to cut off the top positions in the lists for evaluation.
    - "ndcg-","map-","ndcg@n-","map@n-": In XGBoost, NDCG and MAP will evaluate the score of a list without any positive samples as 1. By adding "-" in the evaluation metric XGBoost will evaluate these score as 0 to be consistent under some conditions.
3. lambda [default=0] L2正则的惩罚系数
4. alpha [default=0] L1正则的惩罚系数
5. lambda_bias 在偏执上的L2正则。缺省值为0（在L1上没有偏置项的正则，因为L1时偏置不重要）
6. eta [default=0.3] 为了防止过拟合，更新过程中用到的收缩步长。在每次提升计算之后，算法会直接获得新特征的权重。 eta通过缩减特征的权重使提升计算过程更加保守。缺省值为0.3 。取值范围为：[0,1]
7. max_depth [default=6] 数的最大深度。缺省值为6 ，取值范围为：[1,∞]
8. min_child_weight [default=1] 孩子节点中最小的样本权重和。如果一个叶子节点的样本权重和小于min_child_weight则拆分过程结束。在现行回归模型中，这个参数是指建立每个模型所需要的最小样本数。该成熟越大算法越conservative。取值范围为: [0,∞]

2.3 基准模型和指标

基准模型使用随机森林模型，n_estimators设置为15。其RMSPE分数为0.1607。

在Kaggle此比赛中，参加的队伍共3303支，选取Private Leaderboard榜以排名 TOP 20%对应得分 0.12022，即 RMSPE 误差小于 0.12022，作为基准阈值。如果时间允许，项目 挑战目标是进入TOP10%，RMSPE 误差小于 0.11774。

接着通过训练号的随机森林模型来画出10个最重要的特征。发现其主要的特征为CompetitionDistance、Promo、Store、DayOfWeek等。这与上面的数据初步分析得到的特征重要性有一定出入。



随机森林中最重要的10个特征

3 实现

3.1 数据预处理

- 数据合并和筛选：将Store数据以‘Store’作为key分别合并至 rain和test数据集中；分析发现Open值为0时，Sales值一定为0，因此将Open值为0的数据先删除，将Open值虽然为1但是Sales为不大于0的数据作为异常值删除。
- 数据缺失值补全：分析发现CompetitionDistance、PromoInterval等字段缺失值较多，不适合直接删除，因此将所有缺失值以0填充；test数据中Open的缺失值表明未知店铺知否开业，先按照Open 为1填充。
- 非数值型数据格式转化：将日期Date字段中包含的信息年、月、日、本年第几周分别提取，数据格式为整型；将StateHoliday、StoreType、Assortment三个分类变量进行onehot编码；将

PrompInterval转化为1-12个字段，每个字段分别表示该商店当月是否属于营销月，数据格式为整型；

- 数据缩放：CompetitionDistance数值分布非正态，中位数和均值相差较大，使用自然对数的方式log处理; Sales数值根据Kaggle论坛经验也同样进行log处理，并且保持其他条件不变，测试发现销量值是否进行log处理对结果有非常大的影响。

3.2 特征提取

特征一定程度上决定了模型的上限，因此特征提取至关重要，本项目共选择38个特征，详情如下：

1. 原始集中存在的特征：Store、DayOfWeek、Promo、SchoolHoliday、CompetitionOpenSinceMonth、Promo2、Promo2SinceWeek
2. 数据预处理产生的特征：
   - 日期处理生成特征：year、month、day、weekofyear；
   - One-hot 编码生成特征：StateHoliday_a、StateHoliday_b、StateHoliday_c、StoreType_b、StoreType_c、StoreType_d、Assortment_b、Assortment_c、PromoInterval_1、PromoInterval_4、PromoInterval_7、PromoInterval_10、PromoInterval_2、PromoInterval_5、PromoInterval_8、PromoInterval_11、PromoInterval_3、PromoInterval_6、PromoInterval_3、PromoInterval_6、PromoInterval_9、PromoInterval_12；
3. 数据缩放生成的特征：CompetitionDistance_log；
4. 新生成特征：
   - CompetitionOpen：表示竞品店铺开业至今的时间长度，以月为单位；
   - PromoOpen：表示当前店铺营销开始至今的时间长度，以月为单位；
   - IsPromoMonth：表示当前月是否是营销月

3.3 模型训练

1. 建立初始模型　　本项目选用XGBoost模型进行训练，首先将最后4个星期的数据作为验证数据，其余做为训练数据，使用初始参数 params={ max_depth=10, eta = 0.03, subsample = 0.9, colsample_bytree = 0.7}进行训练，训练结果为 RMSPE：0.1238。
2. RMSPE修正　　通过对比0.98-1.02之间，间隔为0.005的修正系数的预测值的RMSPE，发现0.995系数的RMSPE其值最小，因此最后在设置结果值时需要乘以0.995的系数。

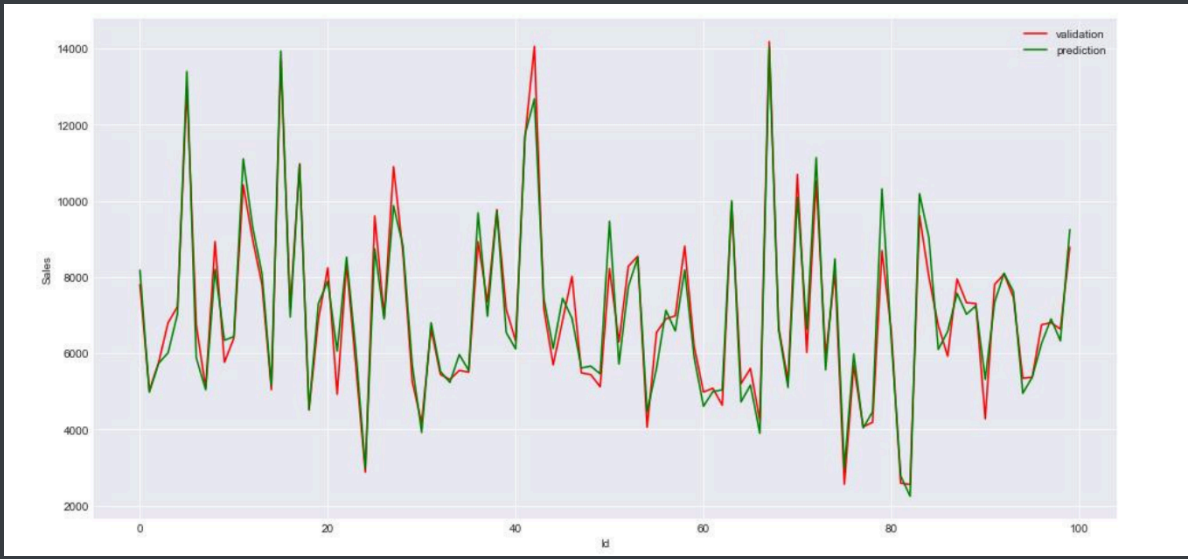
4 结果

4.1 结果可视化

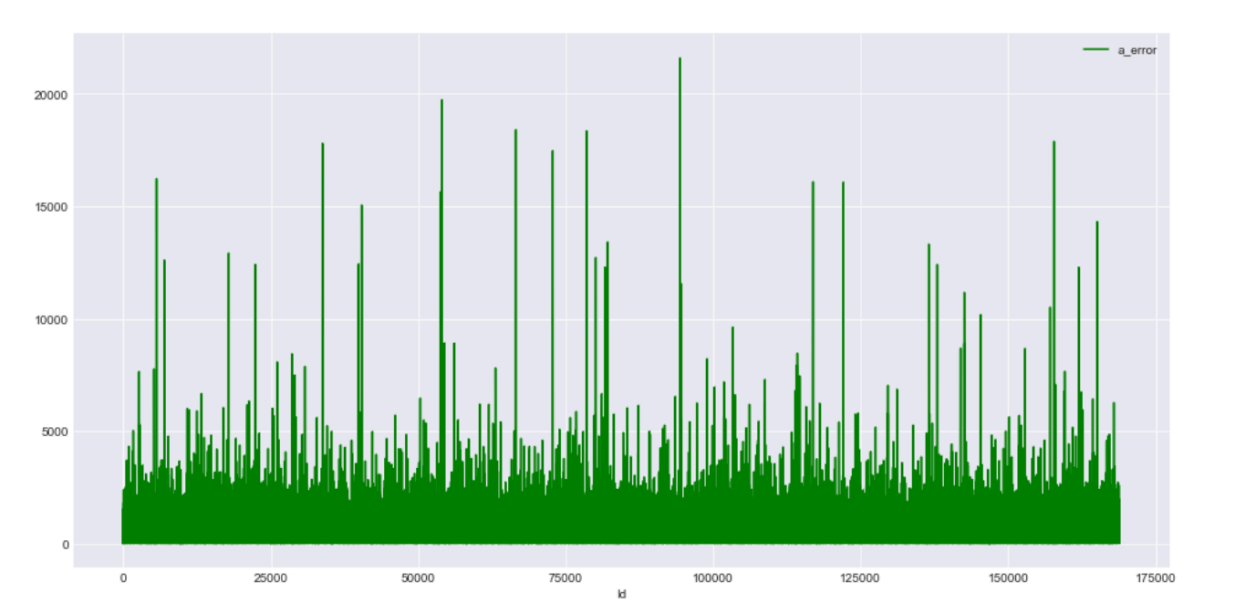
模型在验证集表现结果可以从以下几方面进行可视化分析：

- 由于验证集数据量较大，为了方便观察，随机选取连续 100 个观测点，观察预测值和真实值的对比情况
- 观察模型在验证集上预测值和真实值的绝对误差：|预测值-真实值|
- 观察模型在验证集上预测值和真实值的相对误差：|（预测值-真实值）/真实值|

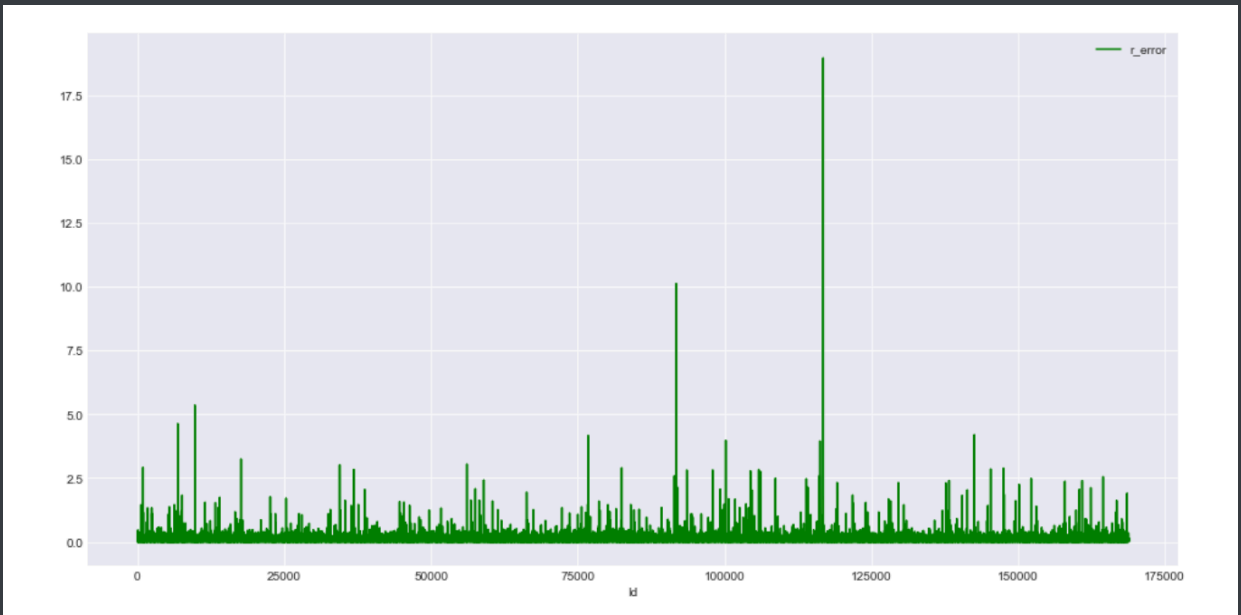
1. 随机选取连续100个观测点，观察预测值和真实值的对比情况



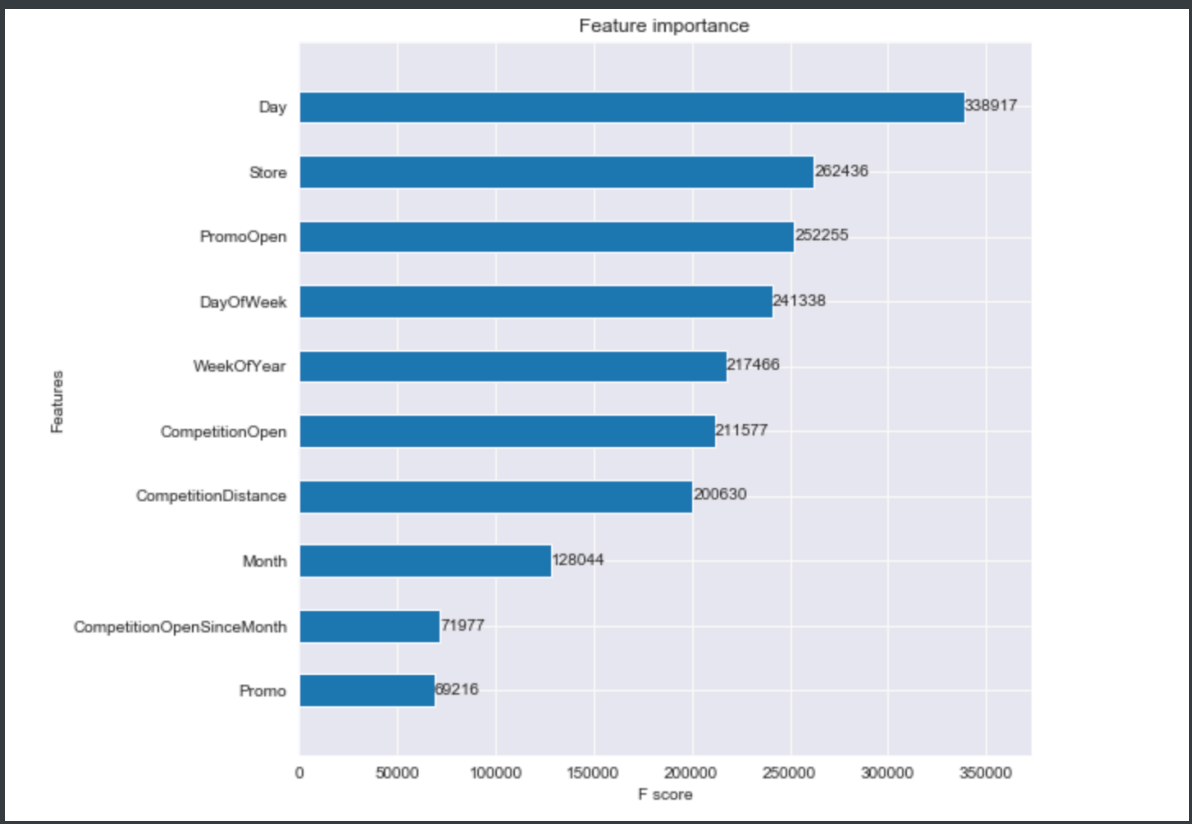
2. 预测值和真实值的绝对误差 从图中可以看出，大部分数据点的预测值与真实值的绝对误差在5000范围内，预测值与真实值相差最大达20000以上。



3. 预测值和真实值的相对误差 从图中可以看出，在验证集上，预测值和真实值的相对误差大部分在0.1以下，只有极少部分的数据点，预测相对误差较大。总体来看，预测结果较为可靠。



4. 特征重要性 通过plot xgboost中的10个最重要的特征，发现其影响比较大的几个分别是Day、Store、PromoOpen、DayOfweek等。



4.2 模型评价和结论

选取模型训练和优化过程中，几个关键结果如下：

模型迭代	Private Score	Public Score
随机森林模型	0.19342	0.18315
XGBoost模型	0.11695	0.10380

从上表可以看出，XGBoost比随机森林模型在Private Score和Public Score上都高。下一步可以通过模型融合的方法来进一步提高分数。

4.3 项目总结

本项目通过数据预处理、特征提取、模型建立以及模型优化等步骤进行1115家店铺的销量预测，最终结果达到0.12356。在数据预处理部分，根据查阅资料，XGBoost模型需要对分类数据进行One-hot编码，已达到CART树中处理离散特征的方式一致，但是在实际应用中，是否进行One-hot编码对结果影响并不大。在特征提取部分，发现增加特征对于训练集的训练分数提升并不明显，但是对测试集有显著的提升，说明当前训练模型有一定过拟合，也表明特征生成和提取对结果非常重要，应该投入更多精力在特征构建方面。

4.4 后续改进

在特征提取部分，可以考虑生成更多特征，Kaggle大赛第一名选手给出了特征生成的建议，例如计算前两年、前一年、前六个月、前三个月的销量中位数、均值、调和平均值、方差、偏差、峰度、10%分位数、90%分位数等，计算营销活动第几天、节假日第几天、暑假第几天、14天营销周期的第几天、本周节假日数量、上周节假日数量、下周节假日数量等，计算销量和顾客数等一些比值数据[7]。 另外可以通过模型融合来进一步提升分数。

参考文献

1.基于机器学习方法对销售预测的研究 2.回归预测评估指标 3.Rossmann Store Sales 4.三种回归算法及其优缺点 5.一步一步理解GB、GBDT、xgboost 6.Wang-Shuo/Kaggle-Rossman-Store-Sales 7.Gert:“Model documentation 1st place”, 8.XGboost数据比赛实战之调参篇(完整流程)