机器学习工程师纳米学位毕业项目

Rossmann 销售预测

姬建业

2018/07/14

1. 定义

1.1 项目概览

Rossmann 是欧洲的一家连锁药店。 在这个源自 Kaggle 比赛 Rossman Store Sales 中，我们需要根据 Rossmann 药妆店的信息（比如促销，竞争对手，节假日）以及在过去的销售情况，来预测 Rossmann 未来的销售额。

Rossman 在 7 个欧洲国家经营着 3,000 多家药店目前，Rossmann 商店经理的任务是预先提前六周预测他们的日常销售。商店销售受许多因素的影响，包括促销，竞争，学校和州假日，季节性和地方性。成千上万的个体经理根据他们独特的情况预测销售情况，结果的准确性可能会有很大差异。

项目要求预测 6 周以后德国各地的 1,115 家商店每周销售，通过历史销售数据构建回归模型，预测未来 6 周的销售量。

销售预测是一个常见的场景，在供应链管理，顾客服务和制造业领域有广泛的应用。使用过去的销量预测未来的情形存在很大的挑战,由于经济下滑,员工离职,流行物品的更新，增加的竞争等因素[1], 一个合适的销售预测模型可以帮助企业节省很多成本，以及及时采取措施来应对预测的有利或不利的结果通常的预测算法有神经网络,基于时间序列的 ARIMA，基于数据挖掘等算法。目前比较流行的是基于数据挖掘算法，通过提取特征，使用 xgboost, GBDT 等算法实现预测[2-3]。

1.2 问题描述：

本项目的目的是通过历史上店铺的销量及店铺的其他相关信息，包括竞争对手的开业，促销及位置信息。为了预测店铺未来六周的销售情况，kaggle 提供了以下数据。

本项目的数据集来自 kaggle Rossmann store sales 项目。数据包含训练集，测试集，商店信息。

训练集和测试集数据包括商店 id，销售额，客户数量，每个周的第几天，商店是否营业，州假日（公共假日，复活节，圣诞节），学校假期。其中连续型变量为 Sales，Customers。离散型变量 Open，Promo，StateHoliday，SchoolHoliday。日期型变量 Date。其中 date 可以拆分成年月日，StateHoliday 需要做 onehot 处理。

商店信息包括商店类型，距离最近的竞争对手商店的距离，最接近的竞争对手开放时间的大致年份和月份，商店当天是否正在运营促销，商店的持续和连续促销，促销开始的年份和月份，促销的启动的月份。其中连续型变量为 CompetitionDistance，离散型变量为 StoreType，Assortment，Promo2，PromoInterval，日期型变量为 CompetitionOpenSinceMonth, CompetitionOpenSinceYear, Promo2SinceWeek, Promo2SinceYear。

预期的解决方案主要如下：

1. 挖掘数据，通过分析数据排除异常点，填充空值。通过分析店铺信息将店铺信息合并到 train 信息中。

2. 分析模型，本次问题为回归问题，首先考虑 GBDT 和 xgboost 给出一个基准的模型 baseline。将训练集分为训练集和验证集。通过参数搜索，实现最优参数的选取以达到最好的模型效果。

3. 考虑模型融合，通过模型融合发挥每个模型的最大优势。以实现模型结果的进一步提升。达到得分要求。

1.3 评价指标

本次评价使用 RMSPE，尽可能的降低测试集的 RMSPE，选取 RMSPE 最低的 1 个或几个模型融合的结果。

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

上式中 y_i 是实际销量。y_i-hat 为销量的预测值。

本项目的任务是达到 kaggle 测试集上满足前 10%得分。也就是在 Private Leaderboard 上的 RMSPE 要低于 0.11773

2. 分析

2.1 数据研究

2.1.1 数据概览

该数据集由以下 3 个部分组成：

train.csv - 历史数据，包括销售

test.csv - 不包括销售额的历史数据

sample_submission.csv - 正确格式的示例提交文件

store.csv - 有关商店的补充信息

字段解释:

Id - 代表测试集中（存储，日期）双倍的 Id

Store - 每个商店的唯一 ID（主键）

Sales -营业额（label）

Customers - 某一天的客户数量

Open - 商店是否开放的指标：0＝关闭，1＝开放

StateHoliday - 表示州休假。通常情况下，除了少数例外，所有商店都将在节假日关闭。请注意，所有学校在公共假期和周末都关闭。a＝公众假期，b＝复活节假期，c＝圣诞节，0＝无

SchoolHoliday - 表示（商店，日期）是否因公立学校关闭而受到影响

StoreType - 区分 4 种不同的商店模式：a，b，c，d

Assortment - 描述分类级别：a＝基本的，b＝额外的，c＝扩展的

CompetitionDistance - 距离最近的竞争对手商店的米数

CompetitionOpenSince [Month / Year] - 给出最接近的竞争对手开放时间的近似年份和月份

Promo - 指示商店是否在当天运行促销

Promo2 - Promo2 对于一些商店是持续和连续的促销活动：0 =商店不参与，1 =商店参与促销 2 自[年/周] - 描述商店开始参与促销 2 的年份和日历周

PromoInterval - 描述 Promo2 开始的连续间隔，命名重新开始促销的月份。例如“二月，五月，八月，十一月”是指每一轮开始于该商店任何一年的二月，五月，八月和十一月

2.1.2 数据描述

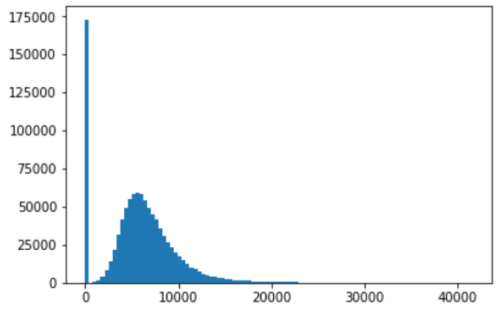
train.csv 训练数据,无空值

Store：值范围 1-1115。共有 1115 家商店

Dayofweek:值范围 1-7，代表周一到周末。

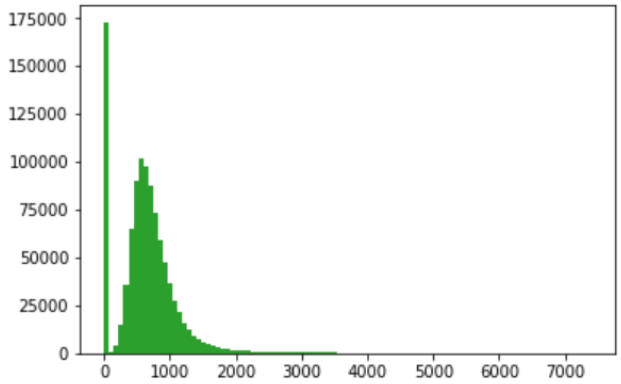
Date：日期型变量：从 2013-01-01 到 2015-07-31。

Sales：连续型数据，平均值 5773.81，最小值 0，最大值 41551 。销量的分布图如下：



可以看出，有大量销量为 0 的情况。通过分析会发现，当 open=0 时，表示商店没有营业。

Customer: 平均值 633.1，最小值 0，最大值 7388。



Open:是否营业：1：844392，0：172817。

Promo:是否促销。 0：62912，1：388080.

Schoolholiday:是否学校假期 0：835488，1：181721 。

Store.csv 商店信息数据

空值分布：

Store	1115 non-null int64
StoreType	1115 non-null object
Assortment	1115 non-null object
CompetitionDistance	1112 non-null float64
CompetitionOpenSinceMonth	761 non-null float64
CompetitionOpenSinceYear	761 non-null float64
Promo2	1115 non-null int64
Promo2SinceWeek	571 non-null float64

Promo2SinceYear 571 non-null float64

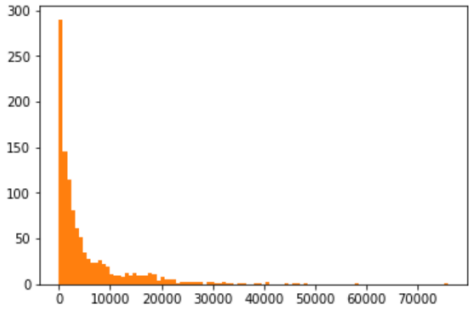
PromoInterval 571 non-null object

其中 CompetitionDistance 有 3 空值，CompetitionOpenSinceMonth, CompetitionOpenSinceMonth, Promo2SinceWeek, Promo2SinceWeek, Promo2SinceWeek 有空值。

Storetype:商店类型。a:602,b:17,c:148，d:348。

Assortment:a:593,b:9,c:513 。

Competitiondistance:有空值，数量较少直接用均值填充。柱状分布图如下。



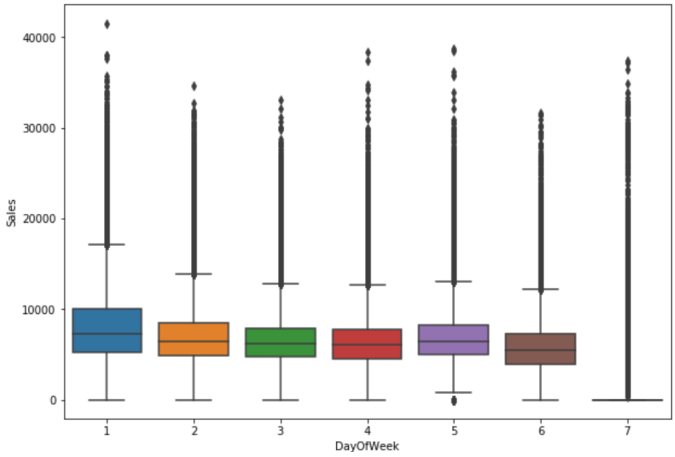
CompetitionOpenSinceMonth 和 CompetitionOpenSinceYear 是竞争者出现的年份和月份，通过抽取几个商店，发现出现竞争者之后商店的销量会出现下滑。后面会进行详细分析。

Promot2:进行促销活动 2，1：571，0：544。

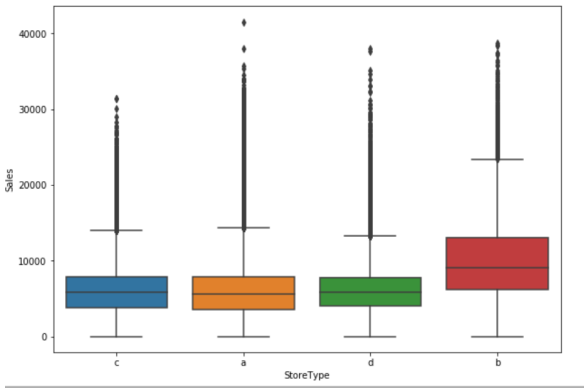
PromoInterval：促销间隔。Jan,Apr,Jul,Oct：335，Feb,May,Aug,Nov：130，Mar,Jun,Sept,Dec：106。

2.2 数据可视化

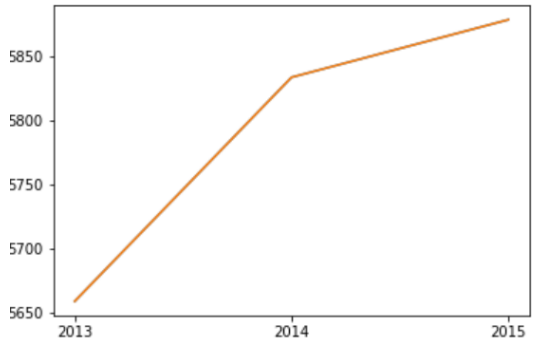
2.2.1 dayofweek 对销量的影响：



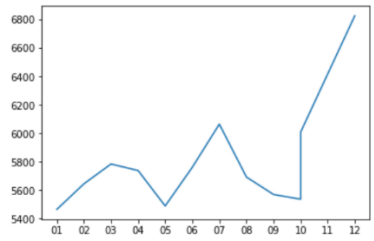
2.2.2 商店类型对销量的影响：



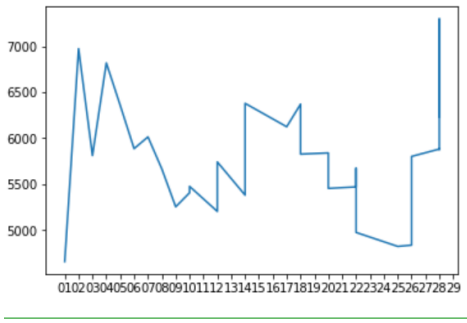
2.2.3 年份对销量影响



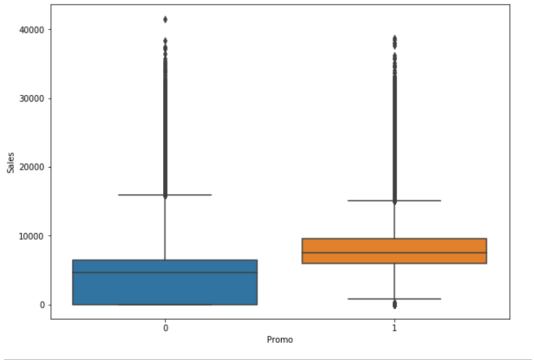
2.2.4 月份对销量影响

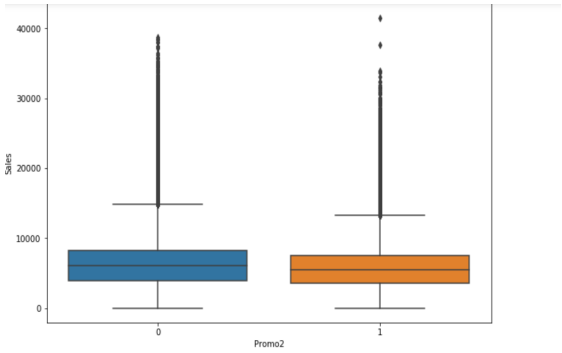


2.2.5 dayof month 对销量的影响



2.5.6 促销活动对销量的影响：





2.3 算法与方法

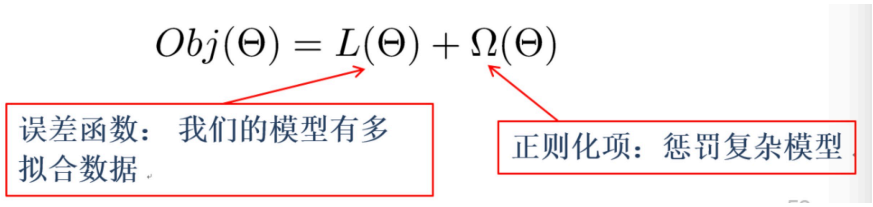
对于数据挖掘问题，常见的方法有基于树的模型和线行回归模型。考虑到本次项目的主要目的是预测销量，销量为连续型变量，且数据的维度较多，使用回归数。

常见的树模型有决策树，随机森林，GBDT, xgboost,lightgbm 等。这些基于树的模型均实在决策树的基础上演变而来，像 GBDT 和随机森林，实际上实在决策树的基础上使用

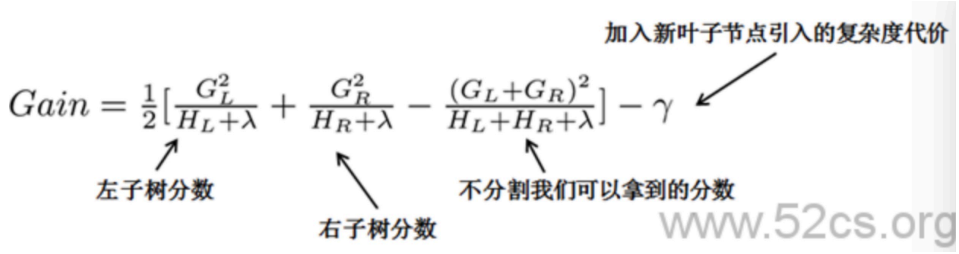
bagging 或 boosting 的思想，通过对多个弱分类器的集成达到一个比较好的效果[4]。

xgboost 是 2014 年 2 月诞生的专注于梯度提升算法的机器学习函数库，此函数库因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[5]。仅在 2015 年，在 Kaggle[2]竞赛中获胜的 29 个算法中，有 17 个使用了 XGBoost 库，而作为对比，近年大热的深度神经网络方法，这一数据则是 11 个。

XGBoost 实现的是一种通用的 Tree Boosting 算法，此算法的一个代表为梯度提升决策树(Gradient Boosting Decision Tree, GBDT)，又名 MART(Multiple Additive Regression Tree)。首先使用训练集和样本真值（即标准答案）训练一棵树，然后使用这棵树预测训练集，得到每个样本的预测值，由于预测值与真值存在偏差，所以二者相减可以得到“残差”。接下来训练第二棵树，此时不再使用真值，而是使用残差作为标准答案。XGBoost 的方法是，将损失函数做泰勒展开到第二阶，使用前两阶作为改进的残差。 XGBoost 使用了一种替代指标，即叶子节点的个数。此外，与许多其他机器学习模型一样，XGBoost 也加入了 L2 正则项，来平滑各叶子节点的预测值。训练每棵树时，不是使用所有特征，而是从中抽取一部分来训练这棵树。这种方法原本是用在随机森林中的，经过试验，使用在 GBDT 中同样有助于效果的提升。在 xgboost 中，也采用了列抽样及行抽样，以达到提升效果的目的。



机器学习就是模型对数据的拟合。对于一组数据，使用过于复杂的模型去拟合，往往会发生过拟合，这时就需要引入正则化项来限制模型复杂度，然而正则化项的选取、正则化系数的设定都是比较随意的，也比较难做到最佳。而如果使用过于简单的模型，由于模型能力有限，很难把握数据中蕴含的规律，导致效果不佳。 Xgboost 通过上式中正则项惩罚模型的参数，防止出现过拟合的情形。



另外 xgboost 通过修正的基尼系数来控制分裂，以达到防止过拟合的目的，通过来保证分裂之后记你系数大于一定值来防止出现分裂过多导致过拟合的情形。

Xgboost 可以说是树模型的集大成者，基本上展现了大部分机器学习模型的训练提升的方法，如集成模型，正则化，剪枝，列抽样等。在很多比赛中都有较为广泛的应用。所以本次项目也将 xgboost 作为首选模型。

Xgboost 模型主要调节的参数如下：

(1) max_depth, min_child_weight 和 gamma，主要用与防止过拟合，减少分裂。

(2) subsample 和 colsample_bytree，主要用于去除噪声，提高模型的鲁棒性。

(3) eta,主要用于控制梯度下降的速度，防止由于速度过快和过慢导致的无法收敛。

2.4 基准测试

在进行 train.csv 与 store.csv 合并后，在没有处理复杂特征的情况下，预测的 rmspe 已经到达 0.22 左右，虽然效果不错，但距离目标 0.117 还有较大的差距，因此想要完成目标还需要考虑更复杂的特征和表现效果更好的模型。

3 方法

3.1 数据预处理

3.1.1 空值填充

1.Competitiondistance 确实值较少，使用均值填充。

2.其他空值表示没有信息，为了区别有值信息，用-1 填充。

3.由于 test.csv 中没有 consumer 信息，根据 consumer 这个信息堆不同的 store 做 groupby，生成每个商店的历史总顾客数。根据 Sales 特征生成每个商店历史总销量。根据 open 信息按照 store 进行 groupby 生成每个商店历史开业数量。之后生成每个商店营业日每天顾客，每天销售额以及每个客户的销售额的平均值。

4.将日期分成年，月，日。

5.将 StoreType,Assortment, StateHoliday,PromoInterval 进行 label-encoding。此处没有用 onehot-encoding 的原因是树模型的分裂根据分裂点左右的基尼系数，使用 onehot-encoding 之后发现对结果的提高并不明显，所以为了减少计算量，使用 label-encoding 。

7,去掉 open=0 的数据，不用于训练。

3.2 实施

在 train.csv 中划分训练集和测试集。训练集。随机抽取 10000 数据作为验证集，其他数据做训练集。训练时取 open>0 的数据进行训练。

3.3 改进

通过调整参数来使模型达到最优的效果,调节的主要参数包括树的深度,学习率,迭代次数,

样本抽样比例和列抽样比例。

	eta	Max_depth	Subsample	colsample_bytree	Valid_rmspe
默认参数	0.3	6	1	1	0.1382
调整后参数	0.01	12	0.8	0.7	0.1006

4.结果

根据 kaggle 上提交的结果，最优的结果 private score 为 0.11711，public score 为 0.10770.

4.1 模型评估与验证

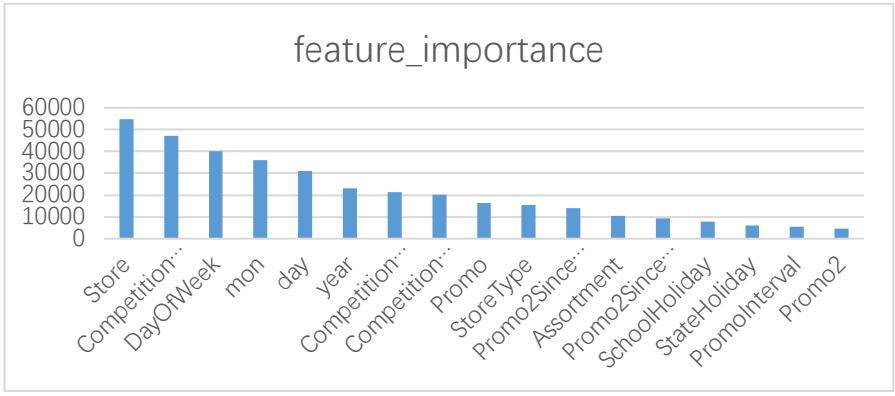
通过合理的组合构造特征及拆解时间特征（将日期分成年，月，日），为训练增加了许多特征（每个商店营业日每天顾客，每天销售额以及每个客户的销售额的平均值），提高了模型的效果。通过对离散型变量进行 labelencoding，很好的解决了模型的输入问题。

使用 xgboost 调整参数，较好的解决了过拟合和欠拟合问题。使得模型得到了良好的表现。

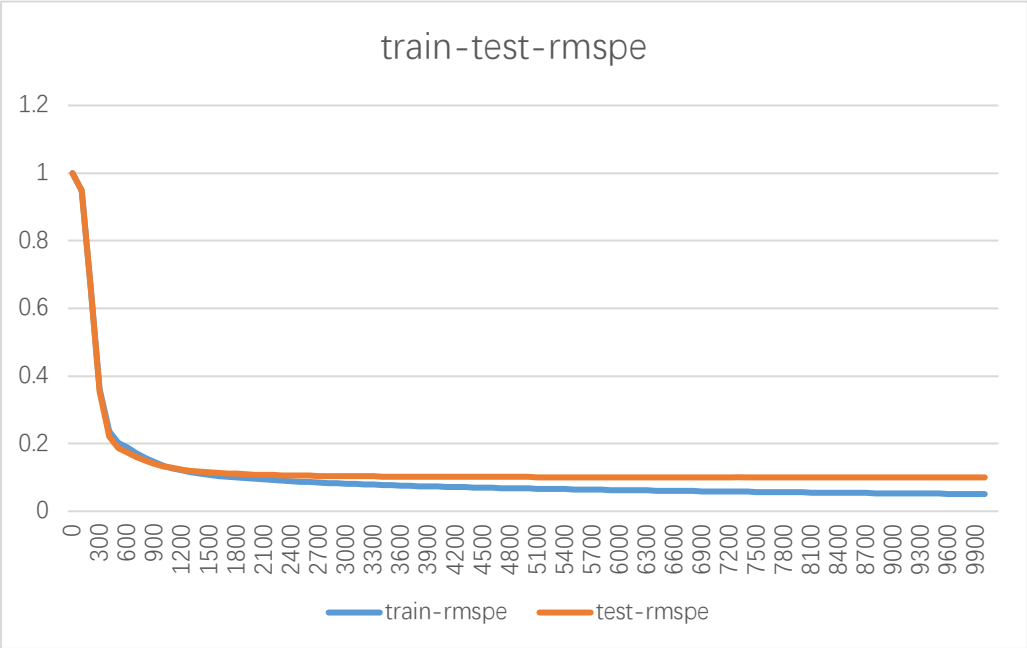
	eta	Max_depth	Subsample	colsample_bytree	Valid_rmspe
默认参数	0.3	6	1	1	0.1382
调整后参数	0.01	12	0.8	0.7	0.1006

5.结论

1.结果可视化



上图展示了特征的重要性排序。可以看到商店 id,竞争者距离，dayofweek 对于模型的结果最重要。



上图为训练集和测试集迭代过程中的 rmspe 的变化,可以看到约 3000 步测试集的 rmspe 已经不太下降了。训练集和测试集的 rmspe 在 0.10 左右，已经达到要求。

3. 思考与改进

特征的挖掘还可以深入在挖掘出更多的与结果相关的特征。

目前的结果只是用一个 xgboost 的模型结果，虽然 xgboost 本身即是一个融合模型，但

主要是树模型，可以考虑深度学习 DNN 和 xgboost 结果合并，这样会对结果带来提高。

由于实际的应用场景为预测未来的销售，将训练集和测试集按时间划分更加合理。

在实施过程中，有使用 lightgbm，虽然速度可以极大提升，但是效果没有 xgboost 好，查看其原因，主要为 lightgbm 算法实施过程中将连续性特征离散化来提高速度，这样并不能保证分裂点事最优的，虽然速度较快，但结果不如 xgboost。

参考文献：

[1] https://money.howstuffworks.com/sales-forecasting3

[2] 郑洪源, 周良, 丁秋林. 神经网络在销售预测中的应用研究[J]. 计算机工程与应用, 2001, 37(24): 30-31.

[3] 刘莹. 基于数据挖掘的商品销售预测分析[J]. 科技通报, 2014, 30(7): 140-143.

[4] https://blog.csdn.net/moledyzhang/article/details/79498520

[5] https://github.com/dmlc/XGBoost