Rossmann 商店销售预测报告

数据挖掘-回归

黄海泰 2019 年 12 月 22 日

1 问题定义

1.1 项目概述

Rossmann 在欧洲 7 个国家经营着 3000 多家药店，商店销售额会受到许多因素的影响，包括促销、竞争、学校、节假日和季节性等。通过历史鉴往知来，商业的中心任务就是扩大销售，从瞬息万变的市场环境灵活经营，用机器学习方法预测销售。对商店的流动性有一定把握，方可通过预测加强计划性，减少盲目性，为商店的稳定运营提供保障。

由于店铺数量多，规模和类型不一，使用机器学习建模可以很好的处理这种复杂情况。

1.2 问题陈述

销售额是一个连续变量，需要用有监督模型进行建模，数据有两大部分，第一部分是店铺类型的介绍：包含商店类型、竞争情况和促销；第二部分是日常运营描述：包含经营时间、节假日和销售额。

这些数据里有些特征是不适合直接加入模型，需要经过一些处理和转换，使到符合建模要求。例如销售额是一个右偏态分布，在建立模型前应该进行对数转换，这样预测结果会更加准确；此外还需要将连续型特征归一化、删除异常个案、填补缺失值。数据所提供的特征还能扩展。像日期这类特征，可以通过加工得到周数、月份、一年中的天数和季节，这些维度能加强模型对季节性的识别，此外还可以额外增加店铺的总销售额、客单价、去年同比增长情况等。

前期数据准备好就可以通过建模来预测未来 6 周的销售额。

1.3 评价指标

评估指标使用均方百分比误差（RMSPE），公式如下：

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

其中 y_i 表示商店单日销售额，yhat_i 表示对应的预测。RMSPE 计算的着重点在误差与基准的差异百分比，这是一种相对误差的计算逻辑，相对于 RMSE 绝对误差计算逻辑来说，可以避免基准值较大从而导致误差更大的情况出现。

2 分析

2.1 数据的探索

2.1.1 数据描述

数据由 Rossmann 提供，分别包含：

- data.csv-包括销售在内的历史数据
- store.csv-有关存储的补充信息

特征描述：

Store	**商店 ID**
StoreType	商店类型
Assortment	分类：a =基本，b =额外，c =扩展
CompetitionDistance	距离最近的竞争对手商店的距离（以米为单位）
CompetitionOpenSinceMonth	竞争对手开放的月份
CompetitionOpenSinceYear	竞争对手开放的年份
Promo2	商店连续促销：0 = 不参与，1 = 参与
Promo2SinceWeek	参与促销开始的周数
Promo2SinceYear	参与促销开始的年份
PromoInterval	促销循环的月份
DayOfWeek	星期
Date	日期
Sales	销售额
Customers	顾客数量
Open	商店是否开业，1 = 打开，0 = 不打开
Promo	商店当天是否促销，1 = 是，0 = 不是
StateHoliday	a =公共假期，b =复活节假期，c =圣诞节，0 =无
SchoolHoliday	学校放假与否，1 = 放假，0 = 不放假

数据展示：

Store	1	2	3	4	5
DayOfWeek	5	5	5	5	5
Date	2015/7/31	2015/7/31	2015/7/31	2015/7/31	2015/7/31
Sales	5263	6064	8314	13995	4822
Open	1	1	1	1	1
Promo	1	1	1	1	1
StateHoliday	0	0	0	0	0
SchoolHoliday	1	1	1	1	1

表一来源于 data 数据集

Store	1	2	3	4	5
StoreType	c	a	a	c	a
Assortment	a	a	a	c	a
CompetitionDistance	1270	570	14130	620	29910
CompetitionOpenSinceMonth	9	11	12	9	4
CompetitionOpenSinceYear	2008	2007	2006	2009	2015
Promo2	0	1	1	0	0
Promo2SinceWeek		13	14		
Promo2SinceYear		2010	2011		
PromoInterval		Jan,Apr,Jul,Oct	Jan,Apr,Jul,Oct		

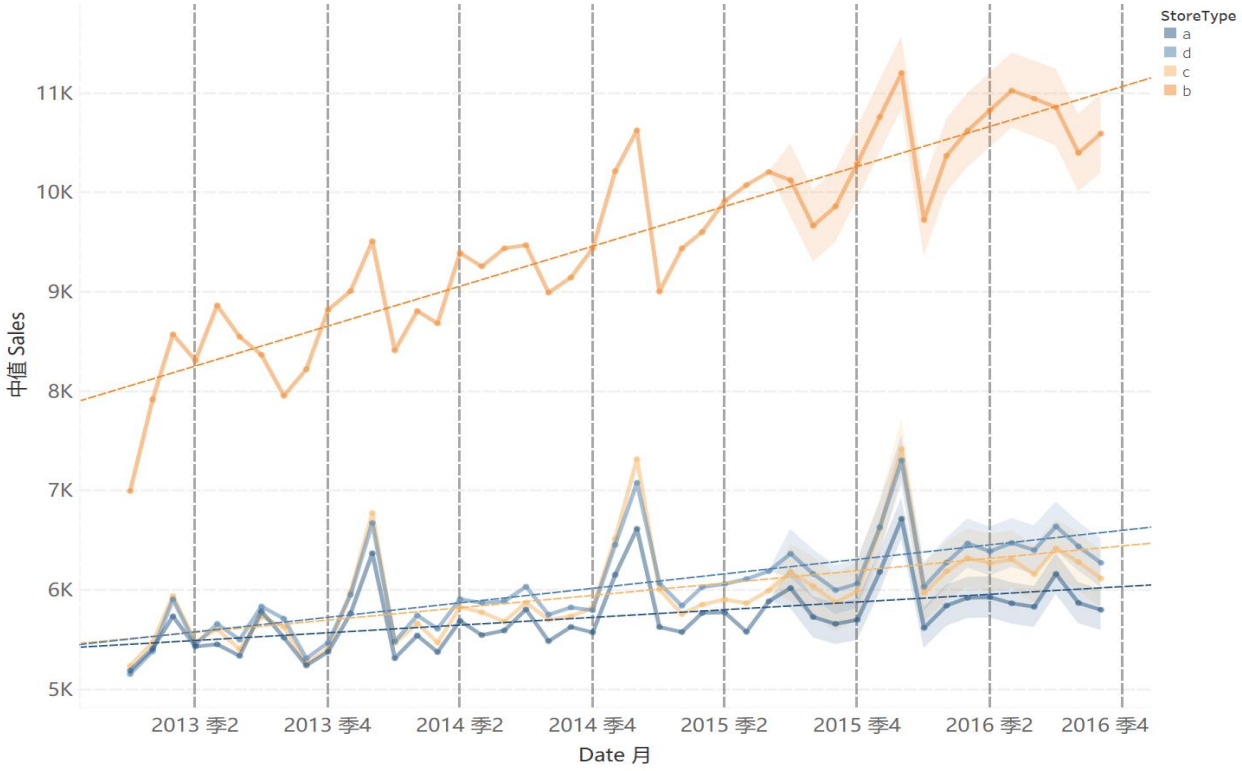
表二来源于 store 数据集

从 store 数据集来看，"StoreType"和"Assortment"是字符型特征，有些模型对输入的特征有要求，对此需要根据模型的要求转换。像线性回归模型要求所有特征都是数值型，而这时候就需要将字符型的特征转换成热编码或数值编码。

Store 表、和 test 表里存在缺失值，其中"CompetitionDistance"目前尚未清楚导致缺失的主要原因，因此可以用中位数、众数或 0 统一填充；通过观察"Open"的缺失一般处于正常营业且没有假期的情况下，因此统一填充数值"1"，表示营业。

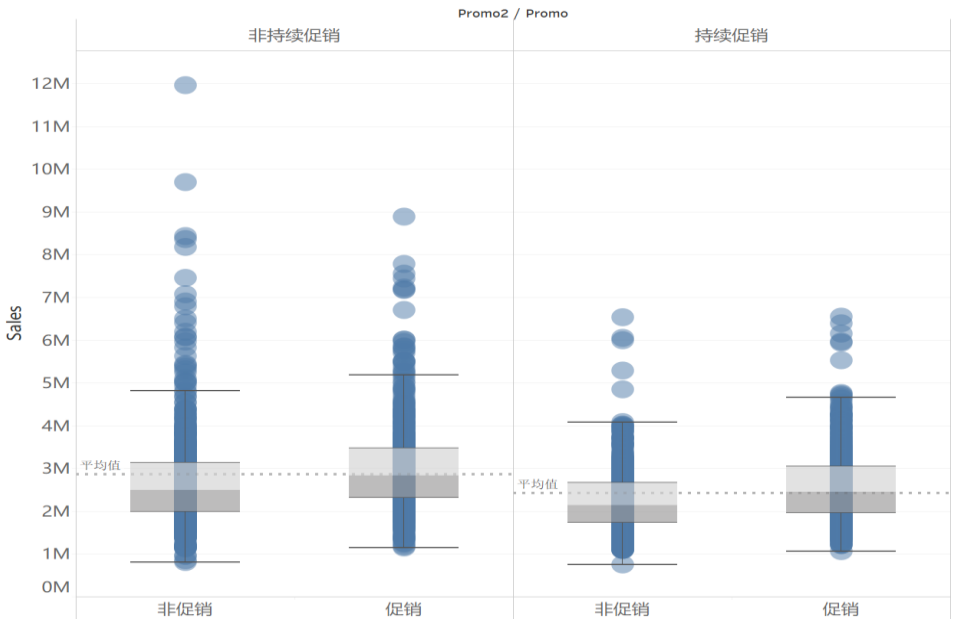
2.2 数据可视化

2.2.1 周期性因素



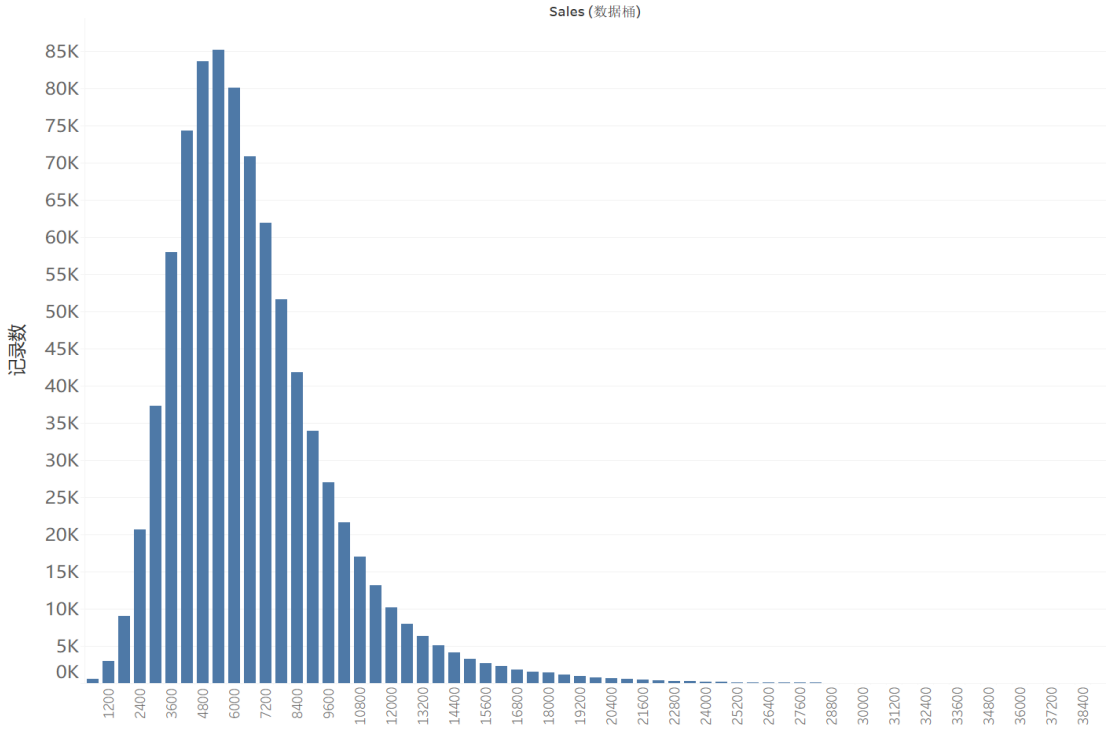
销售额受季节的因素影响较大，从图表可以看出从第四季度开始，销售额上升比较快；另外 b 类型的店铺整体增长趋势比其它类型的商店快。此外，公共节假日也是定期出现，像圣诞节、复活节，顾客都会在前一两天为这些节准备。

2.2.2 促销因素



对于整体而言，有持续促销活动的店铺销售额均值比非持续促销的店铺低。有促销的时候，销售额均值比没有促销的时候高

2.2.3 分布情况



销售额属于右偏分布，对右偏分布的数据进行对数转换，这样有利于模型的拟合。

2.3 算法和技术

Rossmann 商店销售预测属于回归问题，适合回归问题的算法有：线性回归、集成学习等。

线性回归是通过变量与目标变量之间的相互依赖关系来构建，简单的线性回归只适合变量与自变量之间的关系是线性的，对于非线性关系的预测会比较差，对于非线性关系，变量可以通过多项式转换来改善，以此达到更好的拟合效果。

集成学习是一种融合多种模型以达到最优效果的策略，集成学习分为两大类，分别是 Boosting 和 Bagging。Boosting 是运用串行的方式执行训练，各个模型之间有相互依赖关系，每一层训练时候会对上一层级分类器预测错误的样本给予更高权重，以此不断反复；Bagging 训练方式与串行不同，其训练时候各个模型没有强依赖关系，因此模型之间相互独立，最终预测结果是通过子模型"投票"决定

项目会用到集成学习的 Boosting 方法，用到的模型是 Xgboost。编码环境用 Pycharm 和 Anaconda 搭建，编译语音是 Python。Pycharm 编译器是一种 Python IDE，该编译器提供了调试、代码跳转和智能提示等高效率的功能；Anaconda 是一个专门对 Python 包管理的开源软件，其中包含了常用的科学计算包——numpy、Pandas 等。模型用到了 Xgboost_gpu 版本，此版本能够提供 gpu 加速，能够缩短漫长的训练等待过程。

本次项目使用了 Xgboost 模型，Xgboost 模型相对于 GDBT 优点在于添加了正则项：obj(θ)=L(θ)+Ω(θ)，其中 L(θ)是训练损失函数，用于优化模型误差，一般是通过均方误差和 logistic 损失来拟合优化模型，也可以指定损失函数作为优化目标；Ω(θ)是正则项，用于控制模型复杂度，Xgboost 的正则项定义为：

$\Omega(\mathrm{f}) = \lambda \mathrm{T} + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2$，前半部分是 L2 正则，后半部分是叶子数目的正则。

2.4 评估基准

对于回归预测问题，判断的基准是预测与目标之间的差距比率，对于随机预测，误差的范围无限大，因此得出一个基准评分，而基于项目的比赛情况来看，全球参赛选手共 3303 组，第一名的 RMSPE 得分为：0.10021，前 10%的得分为：0.11773，评判标准可以以 0.11773 及格界限为准，从而不断突破。

3 方法

3.1 数据预处理

3.1.1 缺失值填充

数据的缺失情况不严重，其中 test 表格的"Open"是用来表示门店是否开门，参考列表的日期判断，这些缺失都是处于开门状态，因此统一用数值 1 替代。Store 表格的"CompetitionDistance"是竞店的距离，有的门店周围存在竞店但没有竞店距离，因此可以通过竞店距离中位数、众数或 0 替代。

3.1.2 特征工程

日期转换："Date"字段无法直接用于模型训练，因此通过日期特征延展，分别添加对应的年、月、日、日数、周数和季节等，这些周期性指标对训练模型有很大帮助；

热编码：因为"DayOfWeek"，"Season"是连续的数值，会有大小之分，而不同星期或季节的销售额并不是逐级递增，可以通过热编码转换来消除这种关系；

对门店关门的情况判断：根据门店关门的数据来看，门店关门有三种状态：周末、法定假期、装修升级。门店在关闭情况下没有销售额，因此"Open"为 0 的数据将会从训练数据剔除，因此"Open"字段无法提供有效信息训练模型，这时需要新增一列用于判断门店的关门情况，以当天为标准，前面连续关门的次数汇总和是否处于假期前的判断；

新增去年同比数据：根据 Rossmann 销售数据来看，整体处于上升阶段，而且去年同比增长的店铺，今年更有可能持续增长，因此在数据之外统计了各个门店去年同比情况，并加到训练数据里面来；

新增'CompetitionOpen'和'PromoOpen'特征，用于计算某店铺的竞争对手已营业时间和店铺已促销时间，用月为单位表示；

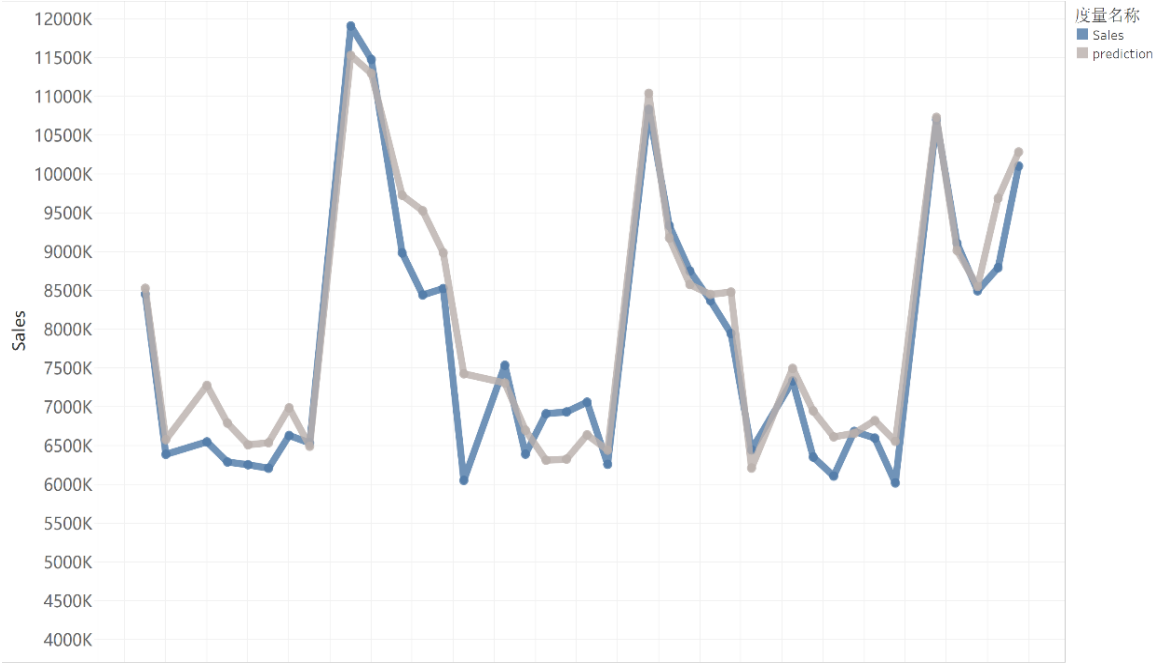
对数转换：对目标字段"Sales"进行对数转换；

多项式转换：由于"Day"，"DayOfYear"与目标不是纯线性关系，而是周期增长趋势，因此可以通过多项式转换来提高非线性拟合。

3.3 完善

由于特征数量比较多，将所有特征放到模型并不会有很好的效果，因此先用周期性特征放到模型进行训练，在初次训练 Private Score 为 0.16265，然后使用步进法逐步添加特征训练，用较大的学习率提高效率，再观察最终提交得到的分数来判断特征是否有效，当评分不再提高时候通过调整模型的学习率以及'max_depth'最大层等参数来提高模型得分。最后通过 5 个不同的随机数种子来训练，最终结果通过加权平均算出，此外为了防止过拟合，用 train 数据集最后 42 天的数据作为测试数据集。

对差异比较大的门店个别分析后发现，销售额处于低位的店铺预测准确性比较差，考虑到不同店铺处于不同成长阶段，有不同的竞争者和促销方式，而模型基于目前的特征还无法完全分别预测不同店铺的销售额，因此不同店铺应该用各自店铺训练的模型进行预测，而不是通过一个模型输出所有店铺。

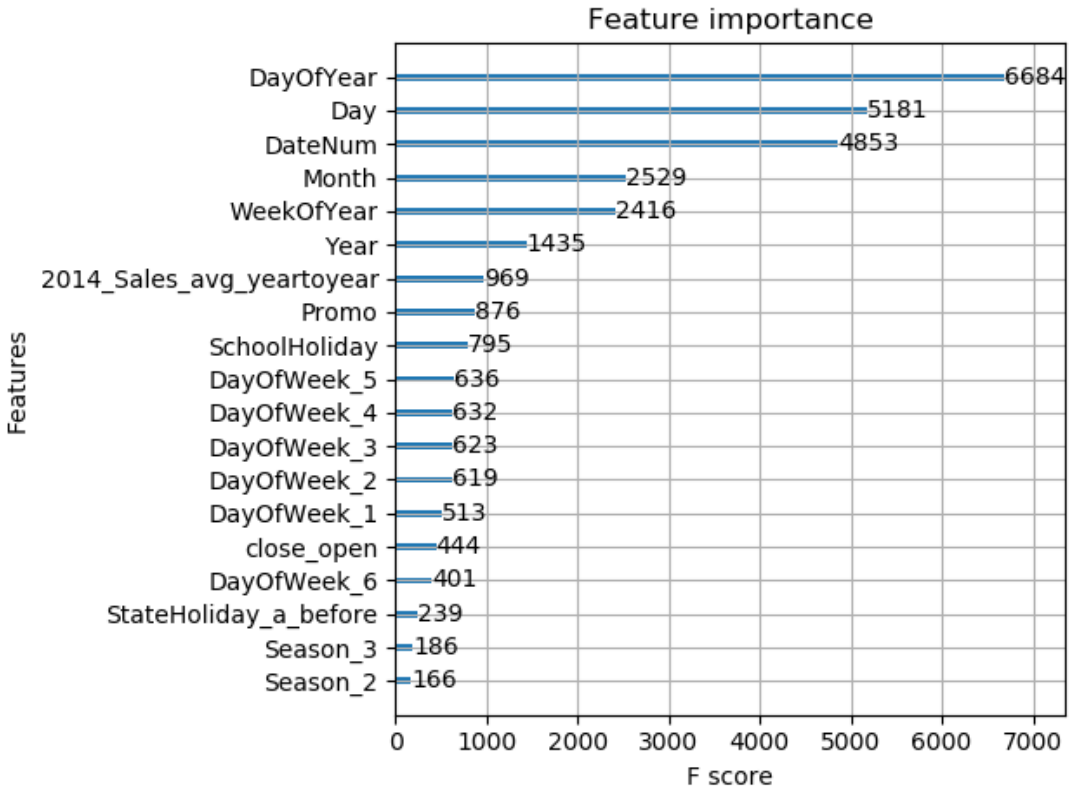


对此应该对每个店铺分开建立模型，这时候应该用到循环将每个店铺列出来，训练的数据来自于一个店铺，因此店铺的固定属性应该剔除，例如：CompetitionDistance、PromoOpen、StoreType、Assortment，重新训练后共有 1115 个模型，每个模型的收敛次数在 450 附近，训练和测试的 RMSPE 分别在 0.03 和 0.085 附近，Private Score 和 Public Score 分别是 0.11223 和 0.10672。

通过多轮训练筛选特征，最终选定一下特征作为输入：

'Promo','Sales','SchoolHoliday','Store','Year','Month','Day','DayOfYear','WeekOfYear','DateNum','DayOf

Week_1','DayOfWeek_2','DayOfWeek_3','DayOfWeek_4','DayOfWeek_5','DayOfWeek_6','StateHoliday_a','Season_2','Season_3','close_open','2014_Sales_avg_yeartoyear', 'StateHoliday_a_before'



4 结果

4.1 模型的评价与验证

模型开始用'max_depth': 10, 'subsample': 0.9, 'colsample_bytree': 0.7,参数时候，预测的结果已经很符合目标的整体趋势，但是对于销售额比较低的情况，预测值大部分情况比实际值高一点，验证集上传后 Private Score 评分为 0.1235。通过各店铺各自建立模型后，Private Score 评分降为 0.11223。

Submission and Description	Private Score	Public Score	Use for Final Score
sample_submission.csv 2 days ago by Heston add submission details	0.11217	0.10638	☐
sample_submission.csv 2 days ago by Heston add submission details	0.11223	0.10672	☐

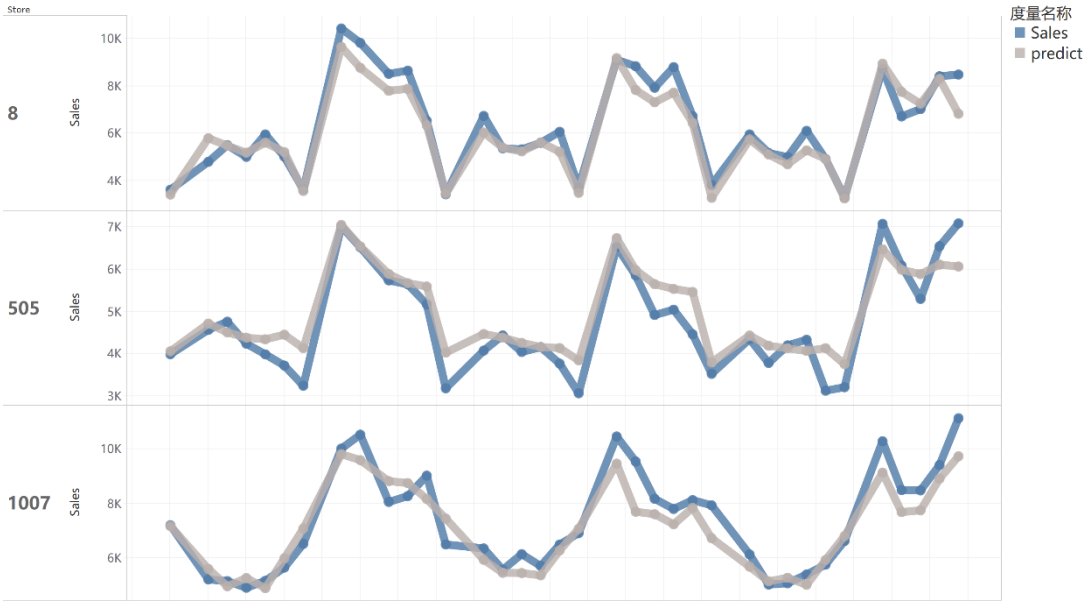
4.2 合理性分析

除了 Xgboost 模型，还尝试了 GDBT，由于 GDBT 每次训练需要等待的时间需要 4 小时左右，有过拟合现象，因此不利于短时间修改参数来优化，而 Xgboost 的训练时间可以控制在 30 分钟左右，且 RMSPE 得分为 0.08，Private Score 在 0.113，过拟合现象没那么严重，所以 Xgboost 可以很好的应付这个项目。

5 项目结论

5.1 结果可视化

经过改进，从随机抽取的三个店铺来看，金额在低点的误差比之前更少了。



5.2 对项目的思考

Rossmann 公司有众多门店，不同门店的类型以及规模大不相同，对于这种情况应该要区分对待，需要为不同门店建立单独的模型来预测；进行特征工程时候应及时对新的特征进行检验，这样会剩下很多特征筛选的时间；项目因为需要筛选 Sales 大于 0 的数据作为训练集时候，这种情况 Open 字段的值只剩下一个唯一值，这样的特征是无法为模型提供有效的信息，这就需要重新考量这个特征的其它含义，如果没有很好利用这些特征，就无法突破更高的分数；模型自带的特征重要性可以作为特征的筛选，但不能完全相信，有些特征重要性比较低删除后会提高整体分数，但有些删除了反而会降低分数，这需要逐步验证。

5.3 需要做出的改进

对模型的参数调整不够完善，此外还可以通过多个模型融合来改善误差。