Rossman Store 销售额预测

潘维维
2019/10/13

模型 Kaggle 的 private score:0.11505(前 5%)

Rossmann 成立于 1972 年，是德国最大的日化用品超市，在 7 个欧洲国家有 3000 多家药店。此次项目需要解决的问题是使用 Rossman3000 多家门店过去 2 年半的相关数据对其未来 1 个半月的销售额进行预测。提供的历史数据集为 2013 年 1 月 1 日到 2015 年 7 月 31 日所有门店的每日销售额，包含客户量、日期、是否促销以及节假日信息。同时提供了 3000 多家门店的 StoreType、Assortment、是否连续促销、促销日期、最近竞争对手距离和竞争门店开设日期。通过以上数据，预测 2015 月 8 月 1 日到 2015 年 9 月 17 日的门店销售情况。

本篇报告会从数据分析、技术方法、结果分析和项目思考及改进四个方面进行阐述，最终获得预测模型，对 Rossman 门店的销售额进行预测。本项目使用的编程语言为 python。

模型评价指标引用了竞赛举办方提供的 Root Mean Square Percentage Error (RMSPE)，公式如下：

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

y_i 为每个商店每天的实际销售额，yhat_i 为对应的预算销售额。此评估指标对于日期型数据预测能稳定的进行评估。同时，销售额为 0 不会对此指标造成影响，可避免因销售额为 0 影响此指标的评估性能。

一、数据分析

1. 数据探索和可视化

1.1 数据集简介

本项目提供了三个数据集：

a. train.csv — 含有销售额的历史数据;

b. test.csv — 和 train.csv 中变量一致，但不含销售额和客户量的历史数据；

c. store.csv — 关于每个商店的一些补充信息；

以上特征变量数据类型有数值型变量，如 Store、Sales、Customers 以及 CompetitionDistance，日期型变量（Date），其他均为分类变量。

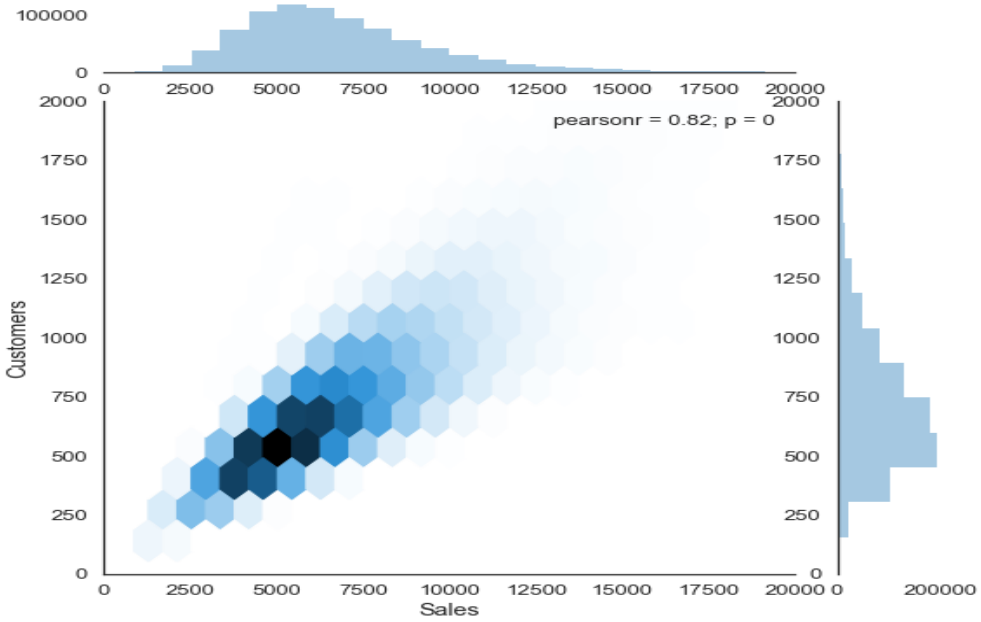
1.2 数据特征探索

本项目的目的是预测 3000 多家门店 1 个半月每日销售额，获取有良好泛化能力的预测模型，然每家门店可能具有其个性化特征，因此，数据特征探索主要从所有门店整体销售额着手，旨在提取其共有特征，而非对每家商店进行单独探索。

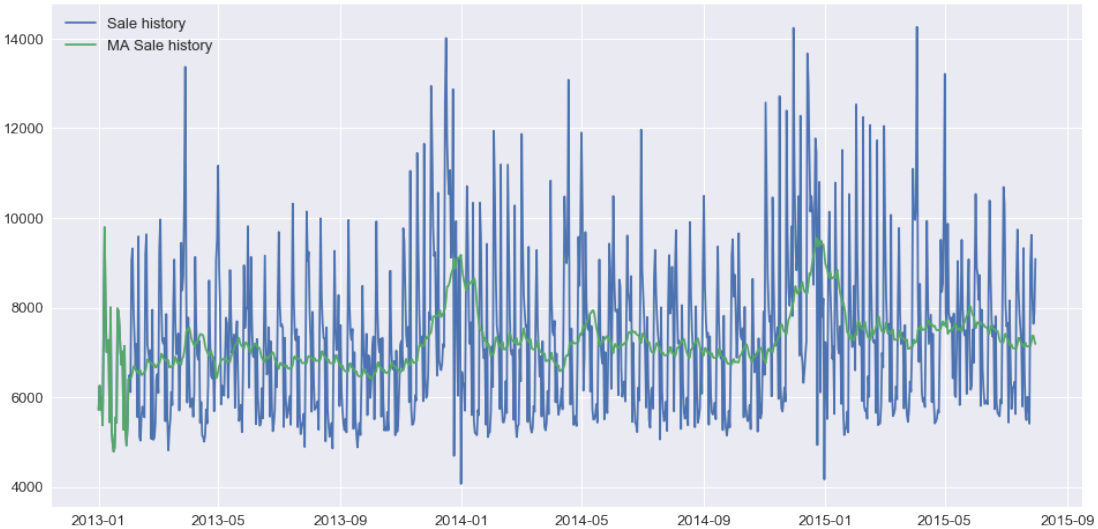
从日常知识推测，门店销售额会受促销、节假日、门店类别、类型和竞争对手等影响。现从数据层面探索 Rossman 门店销售额受以上因素的影响。

1.2.1 销售额整体情况

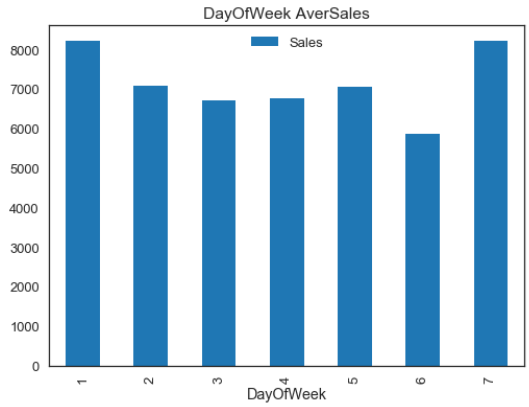
a. 所有门店销售额和偏正态分布，未发现异常值。销售额和客户量呈高度正相关性；

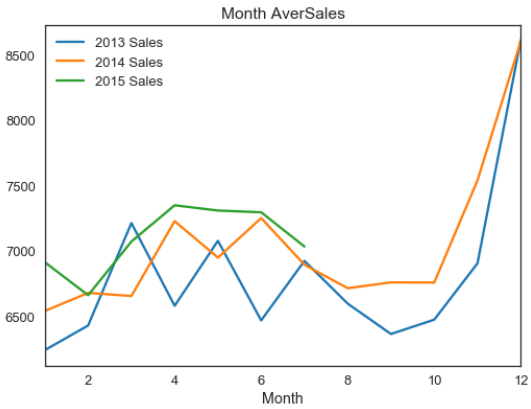


b. 下图为门店日销售额和其移动平均 30 天日销售额关系，相比日销售额趋势，30 天移动平均值波动较为平稳，但其趋势变动和日销售额趋势一致。可考虑增加相关特征变量，进行建模；



c. 门店销售额受星期几影响比较大，同时，月度销售额呈剧烈变动趋势，在 3 月、7 月和 12 月会出现峰值；

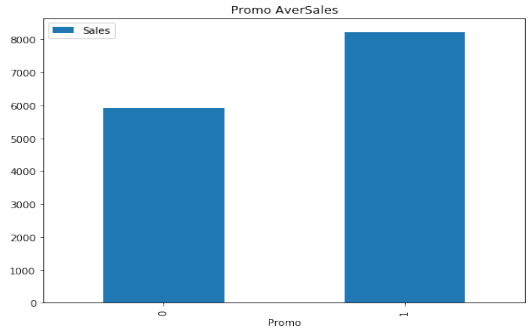


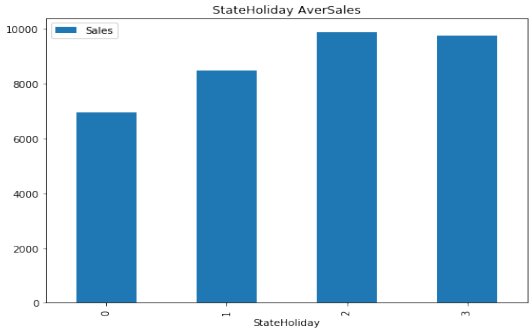


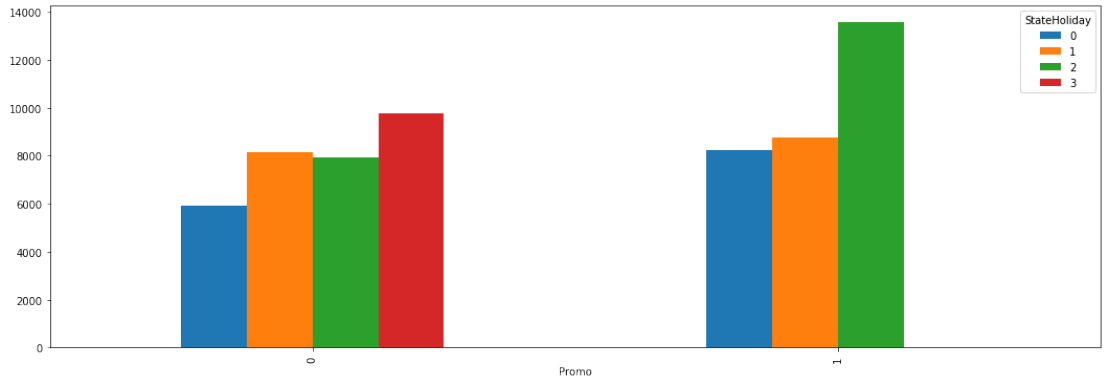
1.2.2 销售额和促销及节假日关系

促销分为短期促销和连续促销

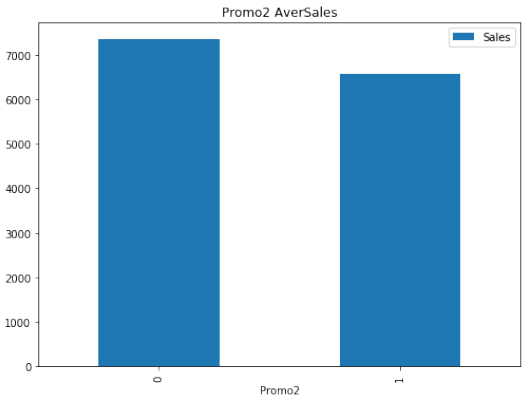
a.销售额受短期促销和节假日影响较为明显，尤其复活节促销时，销售额达到高值。同时，发现圣诞节未参与短期促销，但其销售额仍高于其他假期类型；

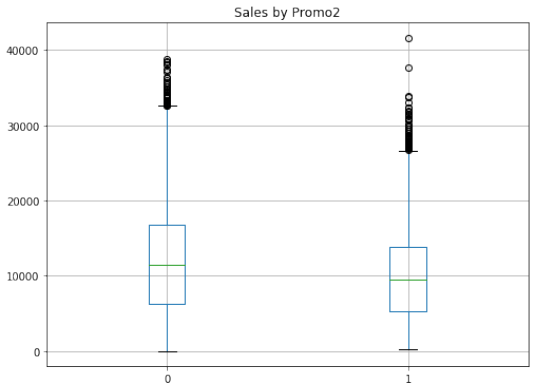




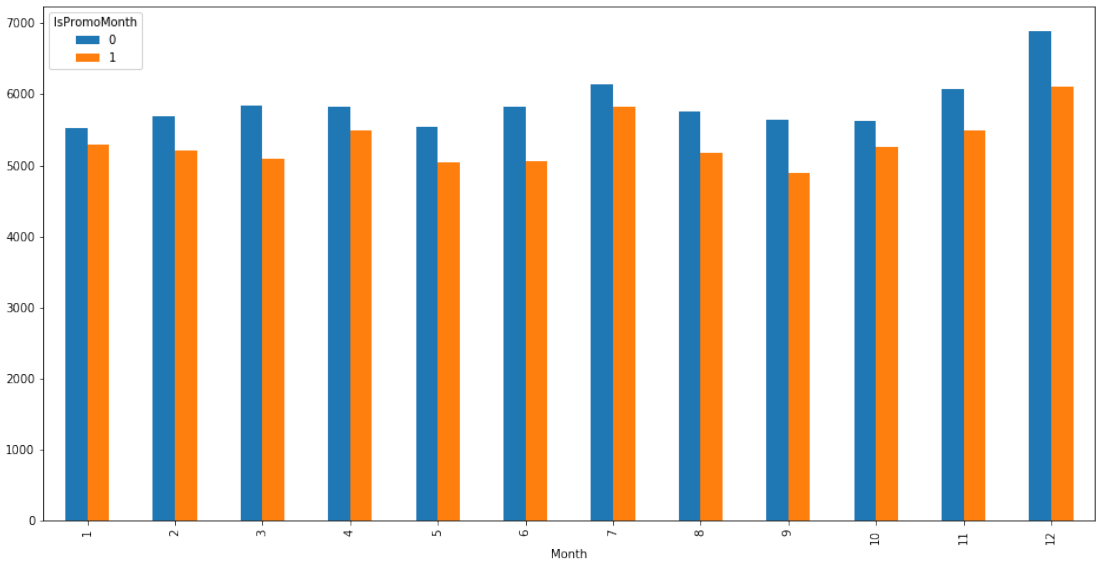


b. 进行连续促销门店的销售额低于不进行连续促销门店的销售额。推测因连续促销的门店本身销售额过低，才进行连续促销；



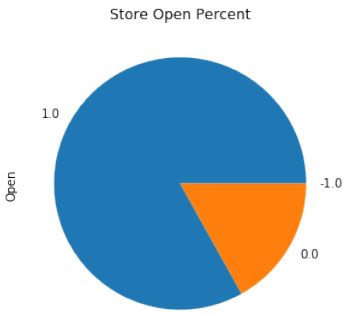


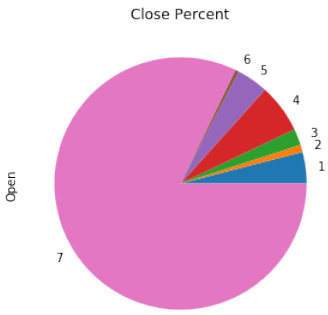
c. 从是否进行连续促销的门店月度销售额进行比较，发现在 7 月和 12 月进行连续促销的门店，整体销售额相比其他月份更高。值得注意是因所需预测数据为 8 和 9 月数据，考虑夏季促销可能对其有更大影响；

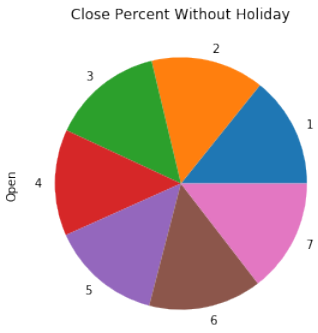


1.2.3 销售额和门店关系

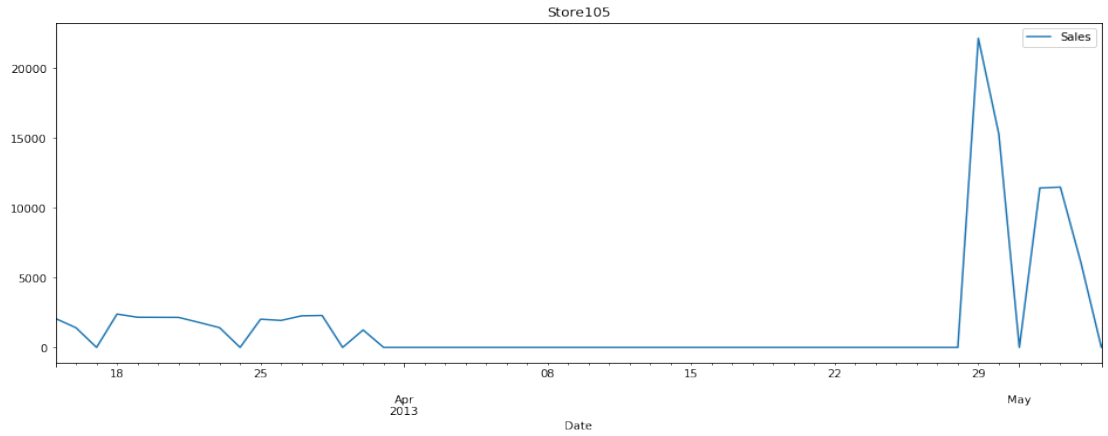
a. 门店关门占比较大，较多是受周日及节日休息影响，但仍存在其他事件引起的关门；



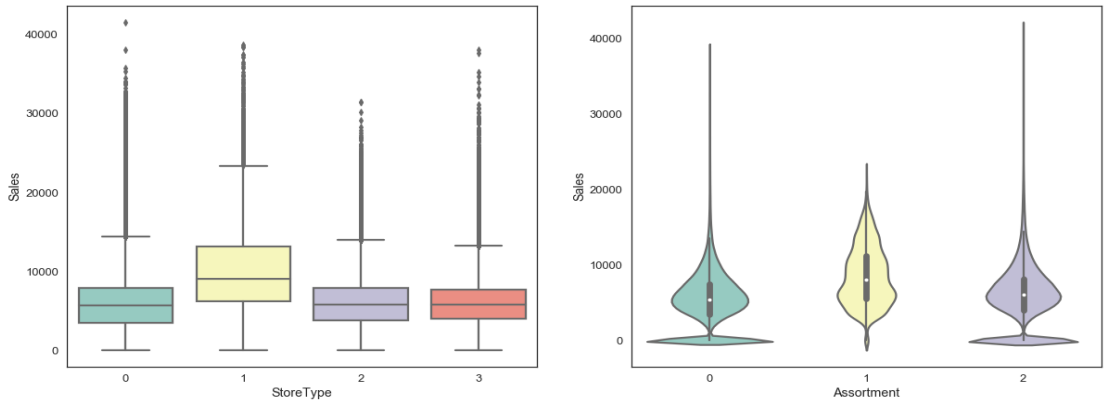




b. 门店 105 在 2013 年 4 月的时候进行了近一个月时间的关闭，但在其门店重新开业后 5 天，销售额有一个激增，把此事件定义为装修效应；



c. 从下图可看到，不同的 StoreType 和 Assortment 对销售额产生一定的影响；



1.3 算法和基准模型

1.3.1 算法和技术

此次需解决的问题为回归问题，然从特征相关性图（见附录**图二**）可看，大部分特征变量和 Sales 变量相关性较小，若单独用基于相关系数的回归模型，效果较差。考虑用的是集成学习技术，通过结合多个模型的决策来提高整体性能。在集成算法中，首先考虑到的是 Boosting 算法中的 Xgboost，此算法是梯度提升算法的高级实现，被广泛推荐。

Xgboost 算法支持线性回归问题。和 GBDT 相比，它的损失函数里加入正则项，

用来控制模型的复杂度。正则项里包含了树的叶子节点个数、每个叶子节点上输出的score 的 L2 模的平方和，从 Bias-variance tradeoff 角度来讲，降低了模型的variance，使学习出来的模型更加简单，防止过拟合。同时，Xgboost 利用泰勒公式进行二阶展开，能更快求解，也优化了分裂点搜索算法，还可以自定义损失函数。另外，其借鉴了随机森林的做法，支持列抽样，不仅能降低过拟合，还能减少计算。鉴于其多重优点，用 Xgboost 作为本项目主要算法。

1.3.2 基准模型

此项目最低要求是进入 Kaggle 前 10%排名，所以本项目我使用的基准模型是在不进行特征选择的情况下，把所有的特征和数据投入 Xgboost 模型中，把其结果作为基准模型结果。此模型使用的 CV 是 10 等分，测试集的 Private Score 为 0.12281，未能进入前 10%，验证集结果如下：

```
Stopping. Best iteration:
[3447]  train-rmse:0.076004     test-rmse:0.090768      train-rmspe:0.082206    test-rmspe:0.103659
```

二、技术方法

1. 数据预处理

此部分主要包括三方面：数据集处理、变量转化和新增变量。

1.1 数据集处理

a.把 train.csv 和 test.csv 数据进行了合并，组成一个数据集；

b.把组合的数据集和 Store.csv 进行左联，组成一个新的数据集；

c.剔除了门店开业时，销售额为 0 的数据；

d.对空缺值进行了-1 填充，便于相关性计算；

1.2 变量转化

a. 把 Sales 变量转为 log 变量，对 CompetitionDistance 进行 MinMaxScaler 标准化处理；

b. 把类别变量 StateHoliday、StoreType、Assortment 转为 Category 变量，便于模型计算；

c. 把 Date 变量转为日期变量，拆分为 Year、Month、Day、DayOfYear、Week 变量。因日期对销售额影响较大，拆分能更好的让模型挑选最适合的时间变量；

d. 把 CompetitionOpenSinceYear 和 CompetitionOpenSinceMonth 两个变量进行组合，转为日期变量，再转化为 int64 格式，新增变量 CompetitionOpenInt，便于模型计算；

e. 把 Promo2SinceYear、Promo2SinceMonth 两个变量同样转为日期变量，再最终转化为 int64 格式，新增变量 Promo2SinceFloat，便于模型计算；

f. 把 PromoInterval 变量进行拆分，最终转为 PromoOpen（对应的日期，连续促销已持续的月份）变量，能更好的让模型对变量进行挑选；

g. 同理新增 CompetitionOpen（对应的日期，竞争门店开业持续的月份）变量；

1.3 新增变量

以下新增变量，均来自前文中数据特征探索。

a. 新增 IsPromoMonth 变量，为对应的销售日期，是否在进行连续促销；

b. 新增 IsSummerPromo 变量，为对应的销售日期，是否在进行暑期促销；

c. 新增 RefurbishEffect 变量，此变量为非节假日和周日停业，超过 4 天，然后再次开业的 5 天，定义为 RefurbishEffect；

d. 新增了 SalesPerDay、CustomersPerDay、SalesPerCustomersPerDay 三个变量，分别为每个门店对应的日均销售额、日均客户量以及日均客单销售额；

最终变量结果见附录（**图一**）。

2. 特征选择

对训练集各特征变量计算相关系数，见附录（**图二**）。因训练模型数据均为开业数据，去掉 Open 变量。因 Week\Month\DayOfYear 关联度很高，去掉 Week 和 DayOfYear 变量。同理，去掉 CustomersPerDay 变量。

通过 XGBboost 模型，进行二次特征选择，特征重要性图见附录（**图三**），在搭建 Stacking 模型时，剔除 IsSummerPromo、RefurbishEffect 特征。

3. 执行过程

3.1 Stacking 模型

因 Xgboost 基准模型，不能达到预期效果，考虑搭建二层 stacking 模型。所使用特征变量如下：

```
#根据XGBoost特征结果，在跑集成模型时，去掉IsSummerPromo,RefurbishEffect特征
feature_xj = ['DayOfWeek','Promo','SchoolHoliday','StateHoliday','Store',
           'Year','Month','Day','StoreType','Assortment','CompetitionDistance','Promo2',
           'CompetitionOpenInt','Promo2SinceFloat','IsPromoMonth',
           'SalesPerDay','SalesPerCustomersPerDay','PromoOpen','CompetitionOpen']
feature_yj = ['SalesLog']
```

模型架构和结果如下：

一层基准模型	**Rmspe**
Random Forest	0.0177689
ExtraTreesRegressor	0.0173214
AdaBoost	0.0299056
XGBoost	0.106261

二层模型(CV=10)	Rmspe
XGBoost	0.078583

然此集成模型虽然在验证集结果较好，但在 Kaggle 测试集得分仍不理想，甚至不及基准模型得分。究其原因可能是训练数据被过于训练，过拟合，导致测试集得分不理想。此模型特征重要性见附录（**图四**）。

3.2 分期间模型

考虑到预测的数据集在 8-9 月，刚好在暑期营销季，且从之前分析中得知，销售额受月份影响较大决定进行季节性分层建模。

分别对数据集 5-9 月数据，1-4 月数据以及 10-12 月数据，建立 Xgboost 模型，再对这三个模型进行加权平均，得到最终集成模型，结果如下：

数据集	**模型**	**验证集 Rmspe**
5、6、7、8、9 月	XGBoost	0.085292
1、2、3、4 月		0.110792
10、11、12 月		0.094005
集成模型		0.11662

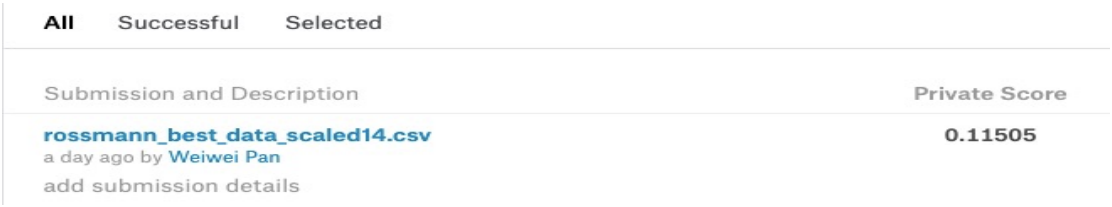
最终集成模型在 kaggle 的 prive_score 为 0.11662，排名 9%左右，符合此项目要求（权重分别为 0.6、0.15、0.25）。

如果分别把期间的三个模型结果单独的在 kaggle 计算测试集 Rmspe，其效果均比较差，然其集成模型效果却比较好，即使在搭配不同的权重情况下，集成模型的结果都较为稳定，原因为集成模型能更充分的利用数据，使模型预测效果变好。

3.3 最终集成模型

从 2.2.2 中的集成模型受到启发，把 2.2.1 得到的 stacking 模型纳入到 2.2.2 中的集成模型中。在调整加权系数的时候，发现比较有趣的现象，对数据集为 5-9 月的暑期效应模型赋予最大权重时，测试集得分会更好。但即使加权系数进行不断调整，整体模型结果均较为稳定，不为出现较大波动。

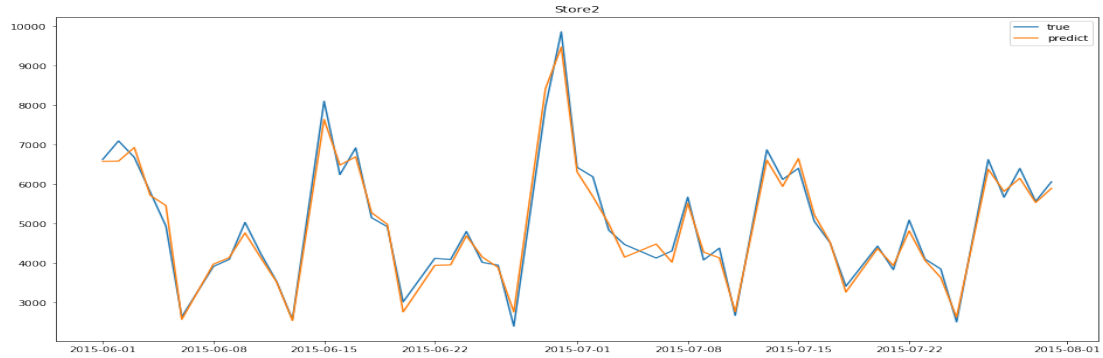
最终集成模型在 kaggle 的得分为 0.11505，进入前 5%，权重分别为 0.25（Stacking）、0.45、0.05 和 0.25。Kaggle 得分如下：

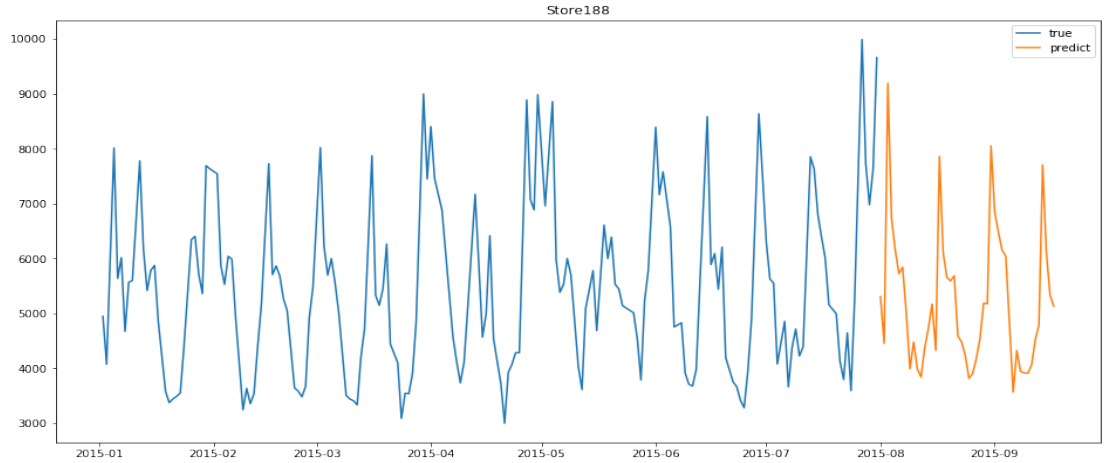


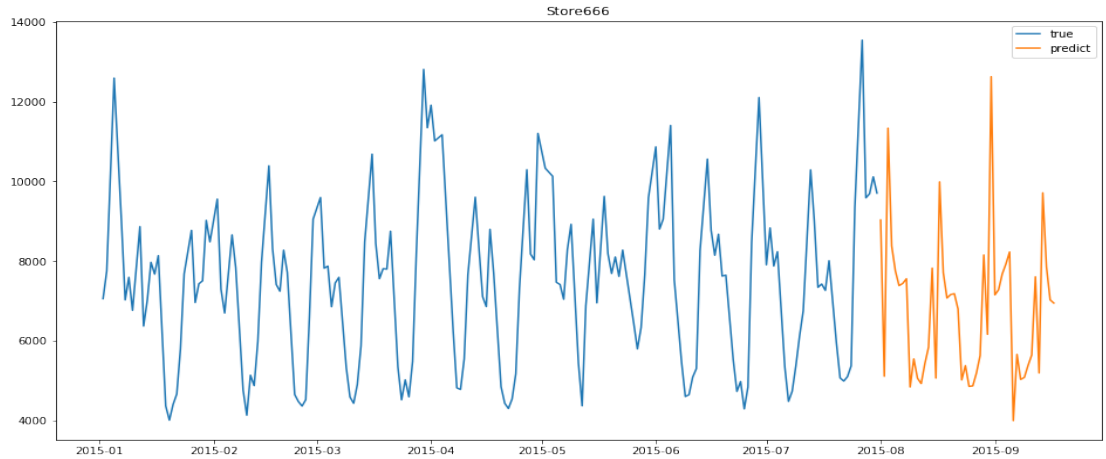
三、结果分析

1. 模型评价及验证

从特征重要性来看（见附录（**图四**）），其结果和前期特征分析结果一致。在进行建模时，均采用了 CV10 份验证，模型的验证得分均较为稳定。同时，Kaggle 测试集 Rmspe 结果，不管是 private 还是 public 得分，也未因为模型的微小调整，出现较大波动。以下为随机挑选的几个门店的预测情况。门店 2 的销售额真实值和预测值基本拟合。门店 188 和门店 666 预测值趋势基本和历史数据趋势一致。







2. 合理性分析

和基准模型相比，最终集成模型的 Rmspe 评估指标得到较为显著改善。最终的集成模型，去除了相关性较大的特征，放大了暑期数据集的影响，放小了其他期间数据集，也综合了部分因 stacking 模型导致的过拟合现象。模型性能更优。

四、项目思考和改进

1. 项目思考

此项目最有趣的地方是对项目进行了分期间数据集的独立建模，再进行加权。从最后的结果来看，暑期数据集对最终的测试结果影响较大。然若只考虑暑期数据集模型，因特征较多，虽然验证集 Rmspe 也较为理想，但因其对训练数据集过度训练，且同

时未对其他数据集物尽其用，导致最终测试结果不理想。而集成模型只选择简单的加权平均，可很好的综合独立模型之间的过度训练情况。

同时，把因过度训练的 stacking 模型加入到集成模型中，赋予权重，为最终模型结果添砖加瓦，显著改善了最终结果。

2. 项目改进

a. 考虑引入天气变量，从 kaggle 讨论群中看，此变量非常有用；

b. 看到第三名的神经网络分享，若能引人神经网络模型，更甚者深度学习模型，模型效果可能会有新的发现；

c. 特征工程是提高模型效用最好的方法，在第一名的分享中，提到装修效应对模型有较好的影响。虽然我有把其加入到模型中，但其特征排名并不靠前。在另一个讨论，若去除门店相关异常值，模型效能会有显著提升。因装修效应，会造成门店出现异常值情况，若把其进行剔除处理，而非当前的变量引入，模型效果可能得到提升；

d. 本次因各种原因，未对模型的参数进行网格搜索，所用参数可能非模型最佳；

文献

1. 第 1 名的参考资料 https://www.kaggle.com/c/rossmann-store-sales/discussion/18024

2. 数据分析项目入门 https://www.cnblogs.com/majimaji/p/10265242.html

3. 如何进入 10% https://dnc1994.com/2016/04/rank-10-percent-in-first-kaggle-competition/

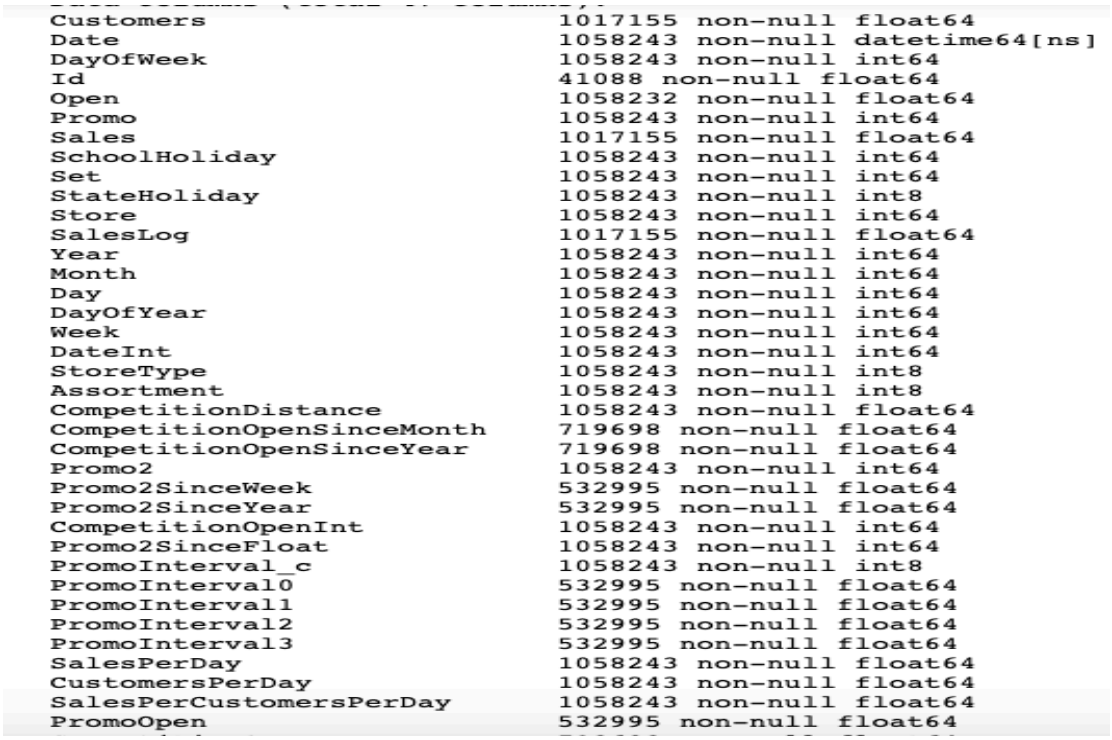
4. 第 72 名的参考资料 https://www.kaggle.com/c/rossmann-store-sales/discussion/17979#latest-167677

5. 从集成到实现集成学习 https://www.jiqizhixin.com/articles/2018-07-28-3

6. 增强树介绍 https://xgboost.readthedocs.io/en/latest/tutorials/model.html

附录

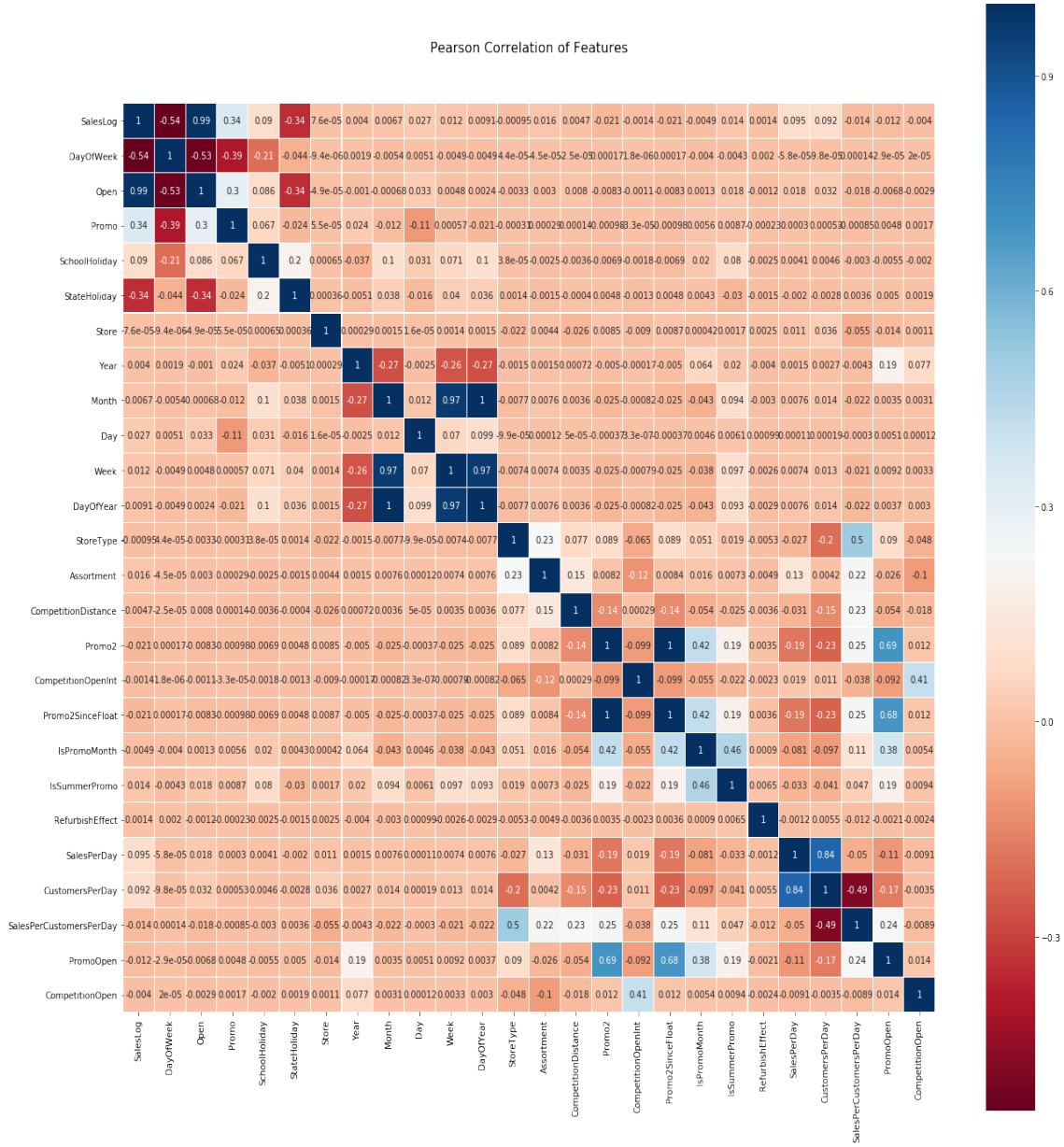
图一：特征变量



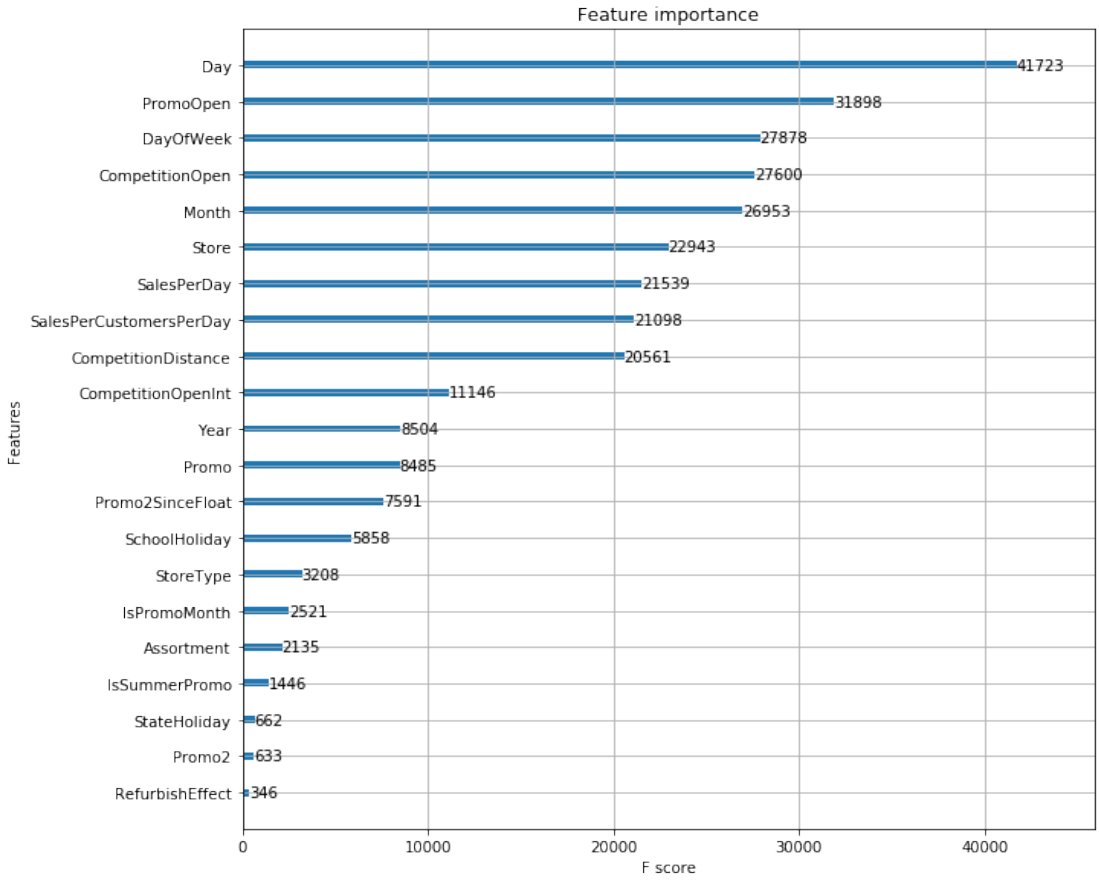
Customers	1017155 non-null float64
Date	1058243 non-null datetime64[ns]
DayOfWeek	1058243 non-null int64
Id	41088 non-null float64
Open	1058232 non-null float64
Promo	1058243 non-null int64
Sales	1017155 non-null float64
SchoolHoliday	1058243 non-null int64
Set	1058243 non-null int64
StateHoliday	1058243 non-null int8
Store	1058243 non-null int64
SalesLog	1017155 non-null float64
Year	1058243 non-null int64
Month	1058243 non-null int64
Day	1058243 non-null int64
DayOfYear	1058243 non-null int64
Week	1058243 non-null int64
DateInt	1058243 non-null int64
StoreType	1058243 non-null int8
Assortment	1058243 non-null int8
CompetitionDistance	1058243 non-null float64
CompetitionOpenSinceMonth	719698 non-null float64
CompetitionOpenSinceYear	719698 non-null float64
Promo2	1058243 non-null int64
Promo2SinceWeek	532995 non-null float64
Promo2SinceYear	532995 non-null float64
CompetitionOpenInt	1058243 non-null int64
Promo2SinceFloat	1058243 non-null int64
PromoInterval_c	1058243 non-null int8
PromoInterval0	532995 non-null float64
PromoInterval1	532995 non-null float64
PromoInterval2	532995 non-null float64
PromoInterval3	532995 non-null float64
SalesPerDay	1058243 non-null float64
CustomersPerDay	1058243 non-null float64
SalesPerCustomersPerDay	1058243 non-null float64
PromoOpen	532995 non-null float64

```
CompetitionOpen                719698 non-null float64
IsPromoMonth                   1058243 non-null int64
IsSummerPromo                  1058243 non-null int64
IsRefurbish                    1058243 non-null int64
RefurbishEffect1               60 non-null float64
RefurbishEffect2               72 non-null float64
RefurbishEffect3               81 non-null float64
RefurbishEffect4               82 non-null float64
RefurbishEffect5               82 non-null float64
RefurbishEffect                1058243 non-null int64
dtypes: datetime64[ns](1), float64(24), int64(18), int8(4)
```

图二：特征相关性



图三：特征选择中 XGBoost 模型特征重量性排名



图四：集成模型特征重要性