机器学习工程师纳米学位毕业项目

Forecast Rossmann Store Sales

陈文冬

2018 年 9 月 5 日

1. 定义

1.1. 项目概述

这是一个来自 Kaggle 竞赛的项目，本项目要求对 Rossmann Store 门店未来 6 周每天的销量做出预测。Rossmann 是一家德国药妆店与日用品超市品牌，在欧洲 7 国拥有超过 3 千家门店，门店的经理希望通过预测未来 6 周每天的销量情况，来帮助他们更好的管理门店的经营，将注意力更好的聚焦在营销和团队管理上[i]。项目给定了该企业的历史数据，既 1115 家门店的历史销售数据，数据范围从 2013 年 1 月到 2015 年 7 月，除此之外，还提供了各门店对应的门店类型、促销状况、竞争对手状况等辅助信息。项目源提供的数据集与相应解释如下所示：

- Train.csv，训练集，包含了 1115 家门店从 2013 年 1 月到 2015 年 7 月的历史销售数据，数据量共计 1,017,209。
- Test.csv，测试集，包含了需要进行预测的门店、日期信息，数据量共计 41,088。
- sample_submission.csv，项目提交格式模板。
- store.csv，门店静态属性信息，包含了门店类型、促销时间、竞争对手开业、竞争对手距离等辅助信息，数据量共计 1115。

1.2. 问题说明

本项目要求对测试集中给定的门店未来 6 周每天的销量做出预测。针对这一目标，制定解决方案如下：

1) 数据预处理。将历史数据与门店数据相关联，得到初步数据集，使得涵盖了各个门店历史销量、门店属性、竞争信息等，形成原始多维数据。
2) 数据清洗。对原始多维数据进行二次加工，观察并处理缺失值，为后续探索性数据分析做准备。
3) 基础特征工程。在数据清洗的基础上，构建基础特征工程，主要包含时间类、竞争对手类、短期长期促销类、节假日类特征的构造。
4) 探索性数据分析与深度特征工程。基于基础特征工程，对影响销量的诸多因素进行探索数据分析，挖掘其中的规律性，并生成相应的新特征。
5) 开发基线模型。首先，基于历史数据划分数据集，形成训练集、验证集和测试集，构造基

线模型对训练集进行建模，用验证集进行参数优化，用测试集进行评估，对预测效果形成初步认识。

6) 开发正式模型。以基线模型阈值为参照，借助复杂算法开发正式模型，在基线的基础上提升预测效果。

7) 模型评估与优化。通过特征优化、参数优化、模型优化等方法，进一步改进模型效果。

1.3．指标

根据项目要求，本项目选择 RMSPE 为评估指标，具体计算方法为：

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

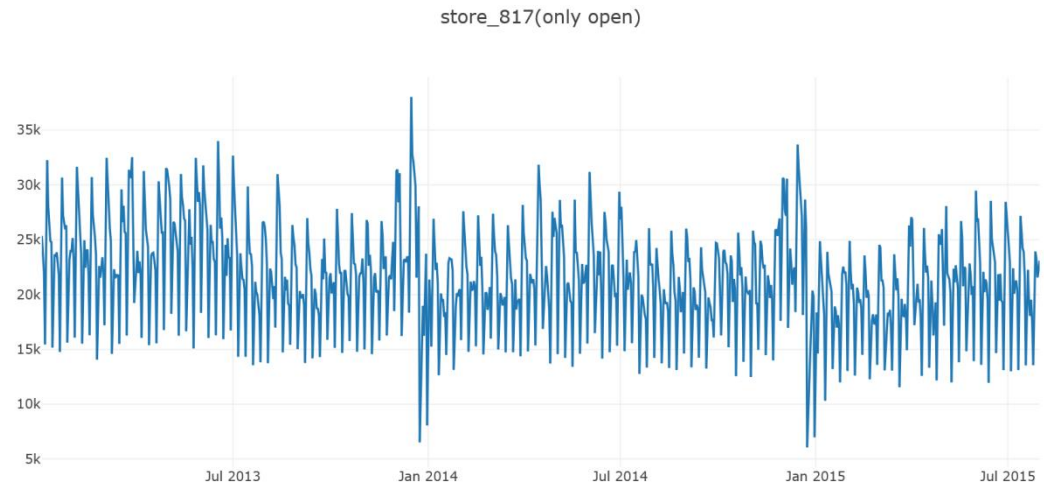
其中，$\hat{y}_i$ 是每个样本的预测值，y_i 是实际值，n 代表样本量，选用 RMSPE 的好处在于无论店铺的实际销量是高是低，RMSPE 都能给出一个合理的评估。

2．分析

2.1．数据研究与探索性可视化

通过数据预处理、清洗、基础特征的构造，形成特征集合，并根据探索性分析研究各个特征与销量的关系，根据需要新增相应的深度特征：

1) 店铺分析（Store）。历史数据给定了 1115 家店铺的过往销量，分析发现店铺不同，销量存在巨大差异。其中，历史平均销量最好的店铺 ID 是 817（平均每天销量达 21757.5）

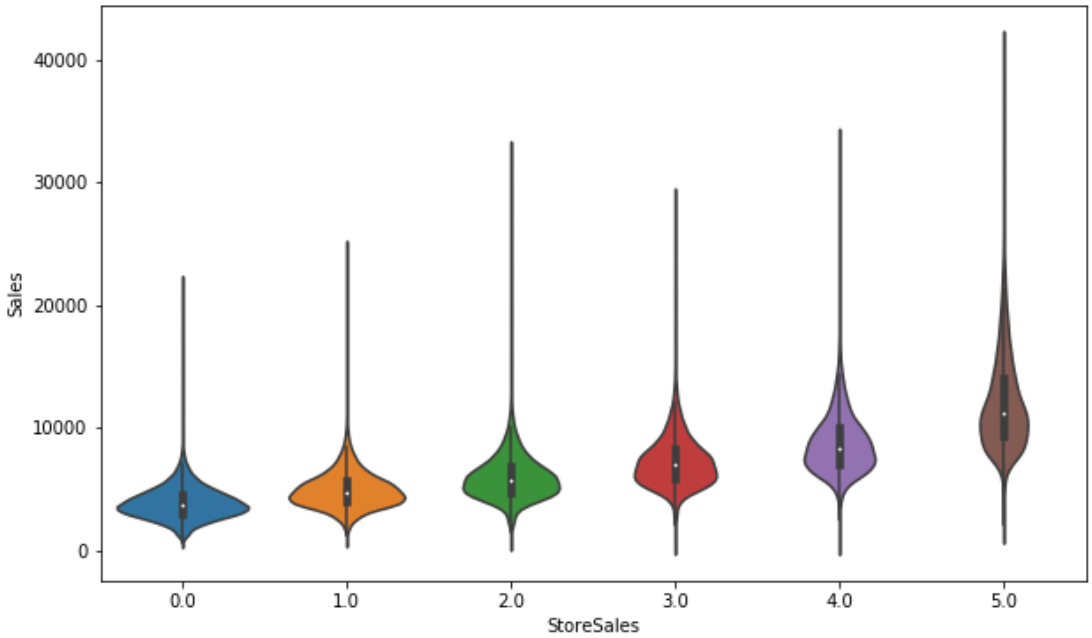


销量最差的店铺 ID 是 307（平均每天销量为 2703.74）

位于平均水平的店铺 ID 是 670（平均每天销量达 6589），且门店销量从 2014 年 7 月到 2015 年 1 月存在断档。

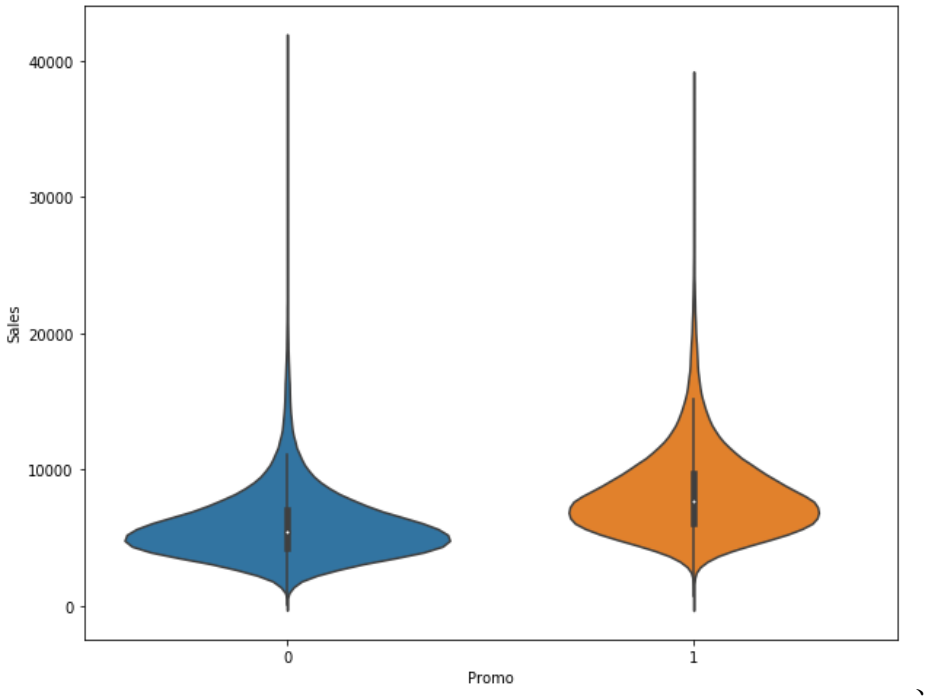
思考：构造新特征（StoreSales），以每家店铺历史均值为样本形成分布，按 10%、25%、50%、

75%、90%分位点进行切分，形成 6 类规模的店铺，通过验证发现 6 类规模的店铺销量呈现明显的不同，从第 0 到 5 类平均销量依次为 3905、4942、5947、7225、8711、12048，且方差逐步增大。



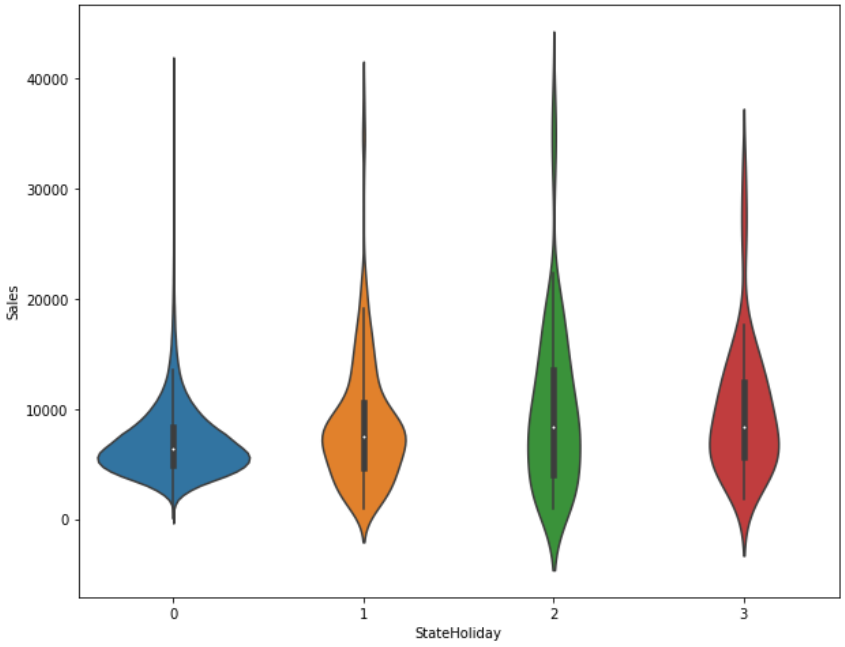
6 类不同销量规模店铺的平均销量分布情况

2) 短期促销分析（Promo）。有无短期促销对店铺的销量有显著性影响，分析发现没有短期促销时的平均销量为 5929，而有短期促销平均销量是 8228。



短期促销的平均销量分布对比

3) 州假（StateHoliday）。州假对销量存在显著的影响，分析发现，没有州假（0）时的平均销量为 6953，PublicHoliday（1）的平均销量是 8487，EasterHoliday（2）的平均销量是 9887，Christmas（3）的平均销量是 9743。



不同州假销量分布对比

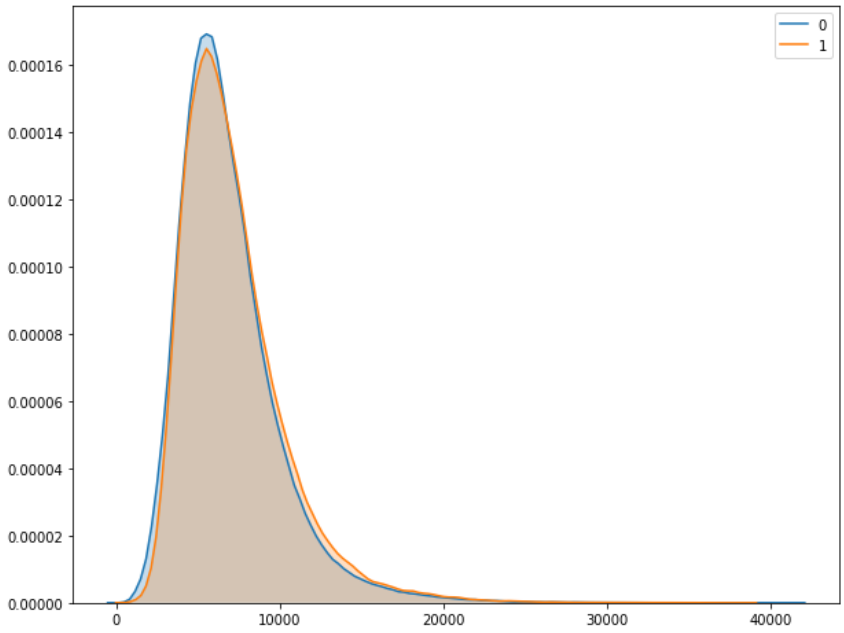
通过方差分析进一步发现除了 EasterHoliday 和 Christmas 之间差异不明显以外，其余组间均拒绝原假设，存在显著性差异（pvalue <0.05）。

Multiple Comparison of Means - Tukey HSD, FWER=0.05

group1	group2	meandiff	lower	upper	reject
0	1	1533.511	1230.7697	1836.2522	True
0	2	2933.9294	2271.8257	3596.0332	True
0	3	2789.7862	1843.6323	3735.9402	True
1	2	1400.4185	672.4879	2128.349	True
1	3	1256.2753	262.9431	2249.6075	True
2	3	-144.1432	-1298.889	1010.6027	False

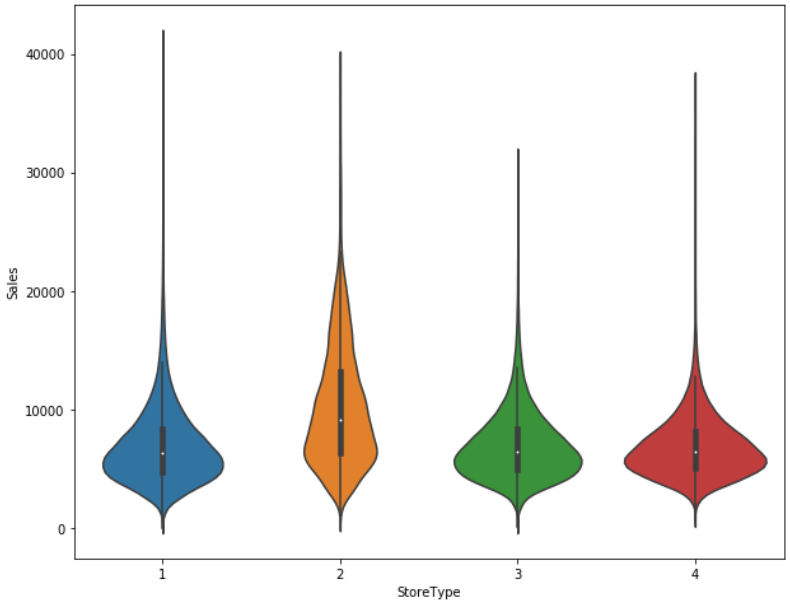
StateHoliday 方差分析-Tukey 检验

4) 学校假期（SchoolHoliday）。从整体数据来看，学校假期对店铺的销量有一定影响，但并不明显，学校不放假（0）时平均销量为 6897，放假（1）时为 7200。



SchoolHoliday 概率密度分布图

5) 店铺类型（StoreType）。从整体数据来看，店铺类型不同，销量也体现出相应的差异，其中店铺类型 2 的平均销量最高（10233），店铺类型 4 最低（6822），店铺类型 1 是 6925，店铺类型 3 是 6933，1 和 3 的销量差异不够显著，其余均呈现显著性差异。



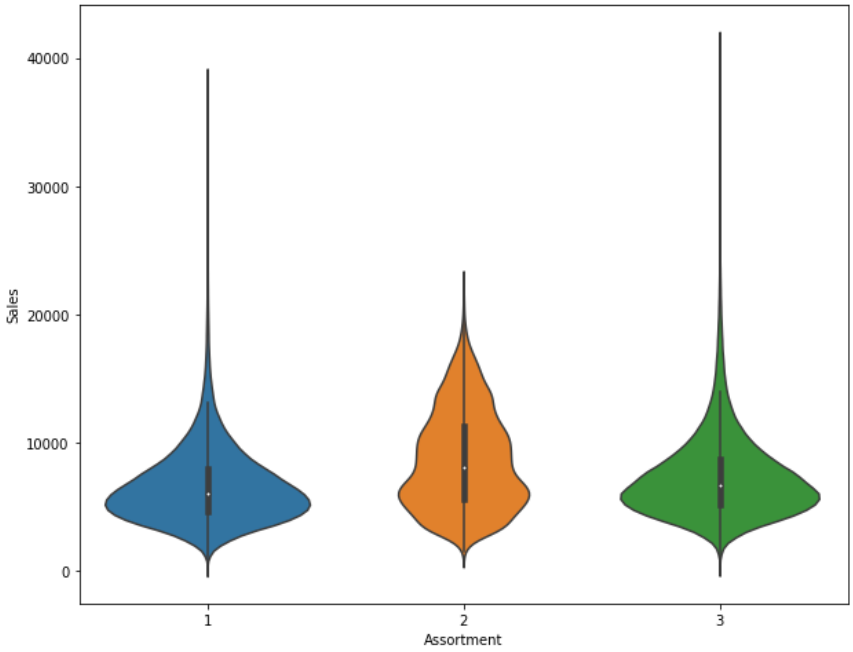
不同 StoreType 的销量分布对比

Multiple Comparison of Means - Tukey HSD,FWER=0.05

group1	group2	meandiff	lower	upper	reject
1	2	3307.6822	3243.3717	3371.9926	True
1	3	7.4284	-18.7837	33.6406	False
1	4	-103.3979	-122.806	-83.9899	True
2	3	-3300.2537	-3367.7117	-3232.7958	True
2	4	-3411.0801	-3476.1967	-3345.9635	True
3	4	-110.8264	-138.9584	-82.6943	True

StoreType 方差分析-Tukey 检验

6) 店铺种类（Assortment）。从整体看，店铺种类不同，销量也体现出相应的差异，其中店铺种类 2 的平均销量最高（8642），店铺种类 1 最低（6621），店铺种类 3 为 7300。



不同 Assortment 的销量分布对比

Multiple Comparison of Means - Tukey HSD,FWER=0.05

group1	group2	meandiff	lower	upper	reject
1	2	2020.9805	1940.5525	2101.4086	True
1	3	679.3205	663.4945	695.1465	True
2	3	-1341.66	-1422.1879	-1261.1322	True

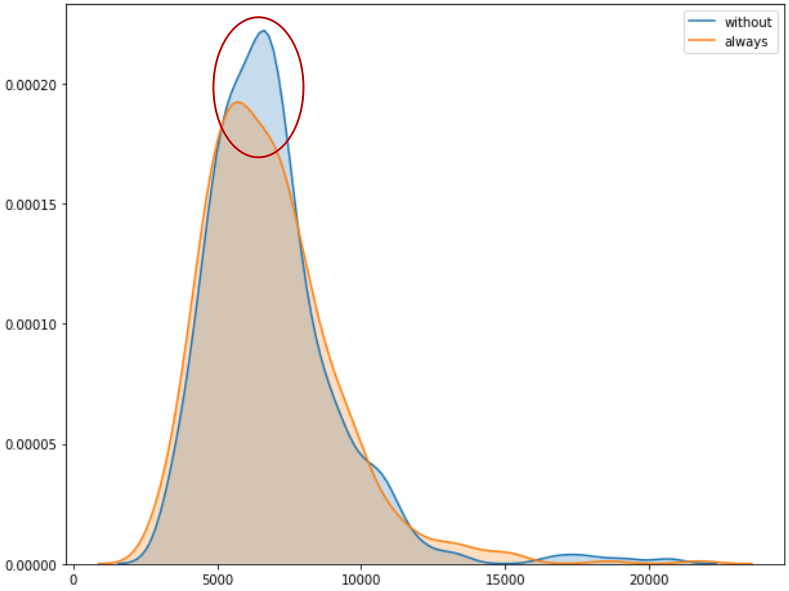
Assortment 方差分析-Tukey 检验

7) 竞争对手是否营业（CompetitionOpen）。在数据预处理中事先定义每家店铺的竞争对手当天是否处于营业状态，是则 1 否则 0。分析发现在我们需要分析的 1115 家店铺中有 357 家店自始至终都没有竞争对手，而有 570 家店自始至终都存在竞争对手，另有 188 家店原先

没有竞争对手，而后竞争对手进入商圈。

思考：竞争对手的存在对销量有无影响？

1、 对这 357 家店（自始至终没有竞争对手）和 570 家店（自始至终都存在竞争对手）进行两独立样本 t-test 发现，没有显著性差异（pvalue 为 0.6312 > 0.05），前者平均销量 6902，后者 6827，两类门店销量数据都存在一定的右偏和尖峰，且无竞争对手的门店右偏和尖峰更严重，说明无竞争对手门店的销量存在更大的极值。但可以认为竞争对手并不是影响这两类店铺销量的主要因素。



从无竞争对手的店铺与始终有竞争对手的店铺销量概率密度图

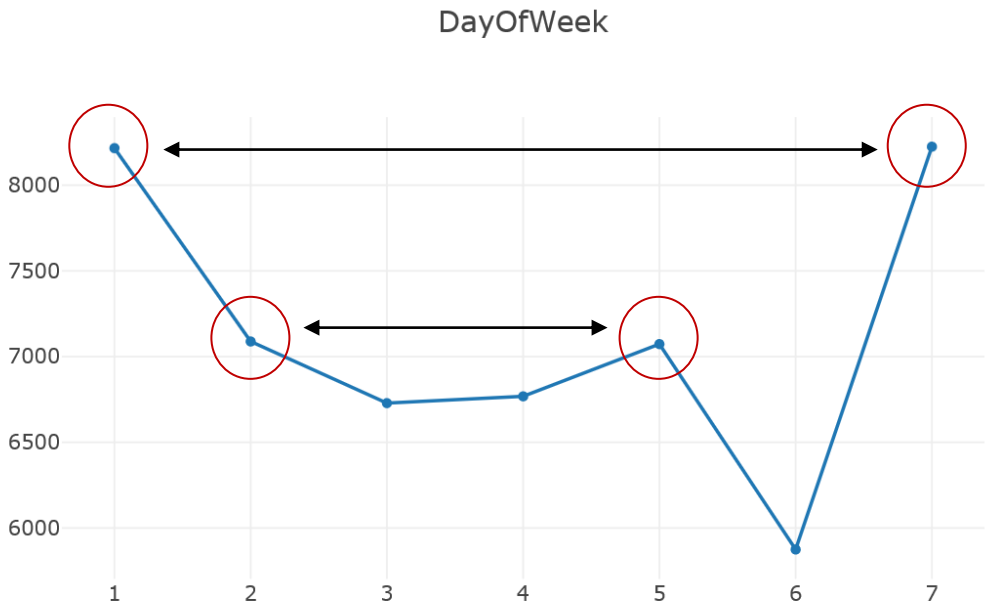
2、 对 188 家门店（竞争对手后期进入）进行配对样本 t-test 发现，竞争对手进入的前后销量存在显著性差异（pvalue 为 0.00044653 < 0.05）,竞争对手进入前平均销量 7409，而进入后平均 7185，出现了一定的下滑，说明竞争对手的进入对这一批店铺的销量存在显著的影响。

因此，构建新特征（CompetitionState）：自始至终都没有竞争对手的门店（0）、自始至终都有竞争对手的门店（1）、此前没有后来引入竞争对手的门店（2）。

8) 竞争对手距离（CompetitionDistance）。通过计算每个门店的销量均值和竞争对手距离之间的关系，发现二者不存在显著的线性相关（pearson 相关系数为-0.055），因此考虑将其离散化，通过 GMM 算法对竞争对手距离进行数据探索，获得聚类数为 2 时轮廓系数达到最大（0.5682），根据聚类中心确定类别边界为 350 米和 4000 米，创建新特征 CompetitionDistanceStr，即小于 350 米，350 到 4000 米，4000 米以上。

9) 周因素（DayOfWeek）。从整体数据来看，周因素对销量存在显著性影响，但是要注意，

周一和周日、周二和周五的差异性不明显，未能拒绝原假设。



周平均销量折线图

Multiple Comparison of Means - Tukey HSD, FWER=0.05

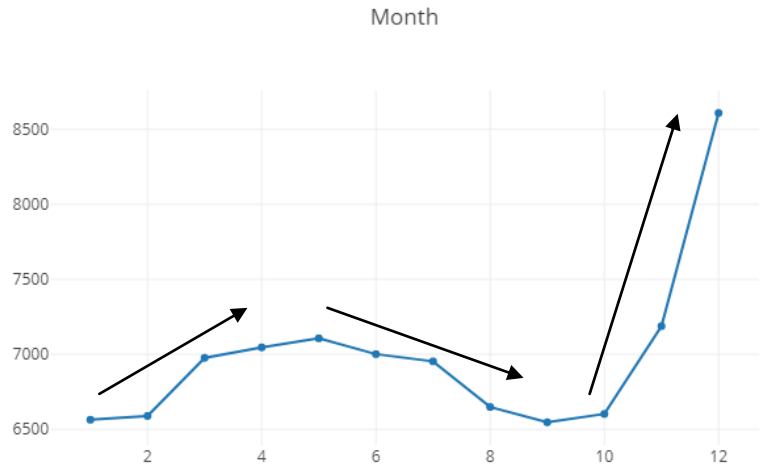
group1	group2	meandiff	lower	upper	reject
1	2	-1127.8432	-1161.4678	-1094.2186	True
1	3	-1487.4656	-1521.2077	-1453.7235	True
1	4	-1448.0373	-1482.2264	-1413.8481	True
1	5	-1143.2181	-1177.1566	-1109.2796	True
1	6	-2341.1673	-2374.7864	-2307.5482	True
1	7	8.4716	-142.2356	159.1789	False
2	3	-359.6224	-392.9816	-326.2633	True
2	4	-320.1941	-354.0054	-286.3828	True
2	5	-15.375	-48.9328	18.1829	False
2	6	-1213.3242	-1246.5589	-1180.0894	True
2	7	1136.3148	985.6929	1286.9368	True
3	4	39.4283	5.5002	73.3564	True
3	5	344.2475	310.5719	377.923	True
3	6	-853.7017	-887.0553	-820.3482	True
3	7	1495.9372	1345.289	1646.5855	True
4	5	304.8192	270.6957	338.9426	True
4	6	-893.13	-926.9358	-859.3243	True
4	7	1456.5089	1305.7599	1607.2579	True
5	6	-1197.9492	-1231.5015	-1164.3969	True
5	7	1151.6898	1000.9974	1302.3821	True
6	7	2349.639	2199.0182	2500.2597	True

DayOfWeek 方差分析-Tukey 检验

10) 年因素（Year）。从整体上看，不同年份销量存在微弱差异，整体呈上升趋势，其中 2013 年日均销量 6814，2014 年日均 7026，2015 年日均 7088。

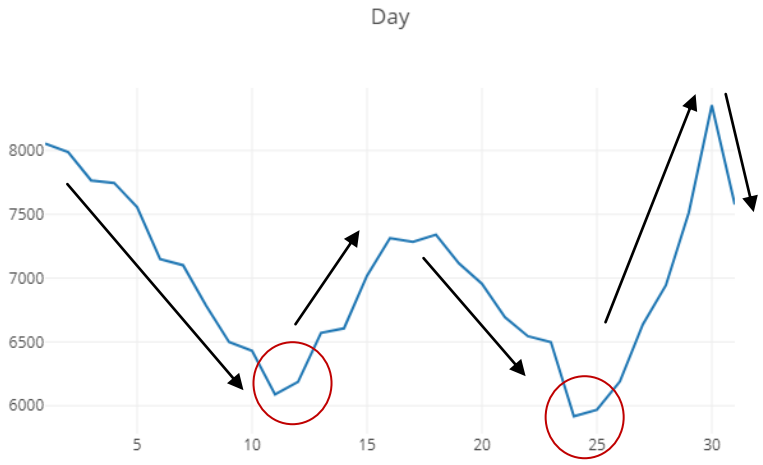
11) 月因素（Month）。从整体上看，不同月份销量存在显著差异，从 2 月份到 5 月份销量呈

现增长趋势，6-9 月出现下滑，10-12 月开始飙升。



月平均销量折线图

12)　日因素（Day）。从整体上看，月中不同的天销量存在显著差异，每月 11、12、24、25、26 号普遍是销量低谷，从 1 号到 11 销量逐渐下滑，11 号到 18 号开始上升，18 号到 25 号又出现下滑，25 号到月底又出现上升。整体呈现“W”型变化。

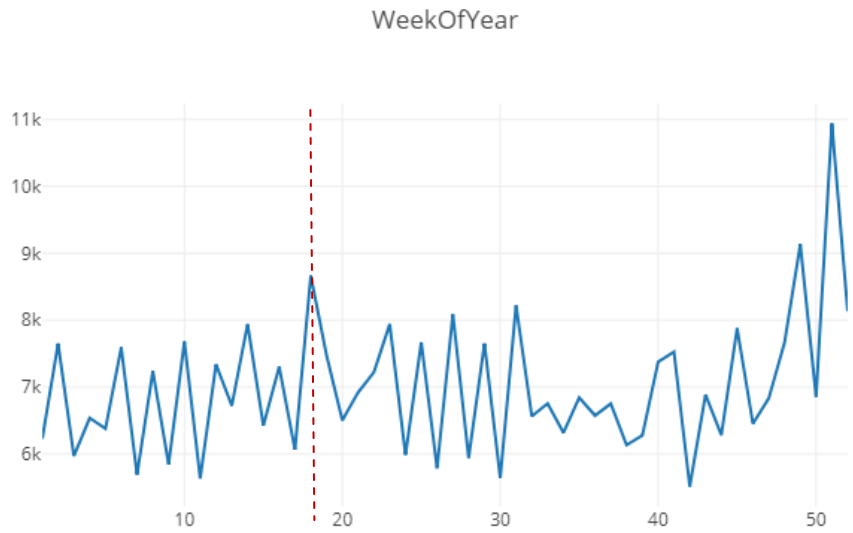


日平均销量折线图

思考：根据日因素构建具体极值日的离散化特征（DayStr），如 1,2,3,16,17,18,30 定义为异常高，11,12,24,25,26 定义为异常低。

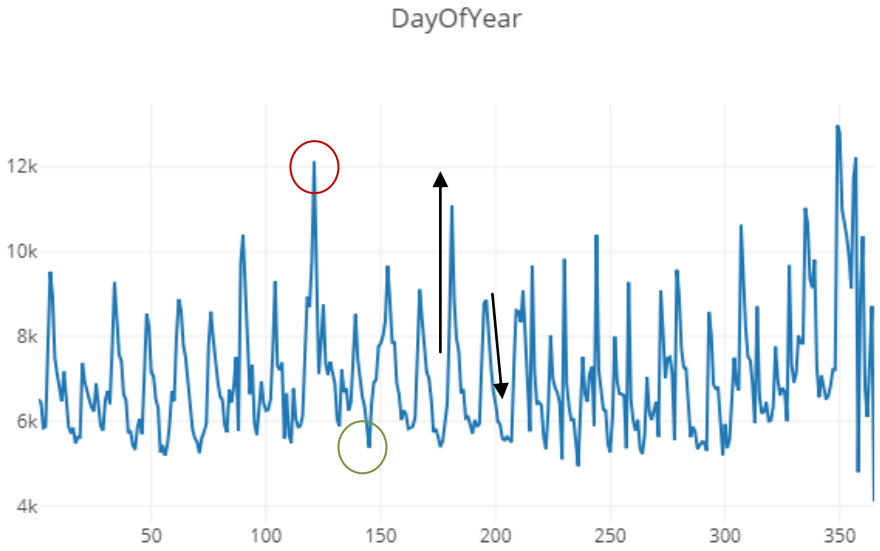
13)　年中周因素（WeekOfYear）。从整体上看，不同周数销量存在显著差异，在第 18 周以前，偶数周的销量普遍高于奇数周；而在 19 周以后，奇数周的销量普遍高于偶数周。思考：根据这

一分布特性定义新特征，18 周以前且为偶数周（0），18 周以前且为奇数周（1），19 周以后且为偶数周（2），19 周以后且为奇数周（3）



年中周平均销量折线图，红色虚线为第 18 周所处位置

14)　年中日（DayOfYear）。从整体上看，部分天存在高销量，部分天处于低谷；有些天前后出现飙升，而有些出现巨大落差。因此思考，新增四类日期数据，分别为“高峰日期”、“低谷日期”、“飙升日期”、“陡降日期”。“高峰”、“低谷”考虑极值，具体为越过 0.5 倍四分卫极差定义为高峰或低谷；“飙升”、“陡降”考虑前后两天斜率变化，具体为斜率越过 1 倍四分卫极差定义为飙升或陡降。



年中日平均销量折线图，红圈代表“高峰”，绿圈代表“低谷”，箭头分别代表“飙升”和“陡降”

15) 旬因素（Tenday）。新增构建每月上中下三旬。从整体上看,每月的上中下三旬销量存在显著性差异。上旬日均 7286，中旬 6844，下旬 6755，方差分析差异显著。

Multiple Comparison of Means - Tukey HSD,FWER=0.05

group1	group2	meandiff	lower	upper	reject
0	1	-442.0912	-461.5732	-422.6092	True
0	2	-530.9528	-550.3565	-511.5492	True
1	2	-88.8616	-108.0138	-69.7095	True

Tenday 方差分析-Tukey 检验

16) 明日闭店且今日是周六（WillClosedTomorrow_TodayIsSat）、明日闭店且今日不是周六（WillClosedTomorrow_TodayIsNotSat）、昨日闭店且今日是周一（WasClosedYesterday_TodayIsMon）、昨日闭店且今日不是周一（WasClosedYesterday_TodayIsNotMon）。分析中发现这四类特殊的日期下平均销量较为异常，不是异常高就是异常低。因此考虑单独标记作为新特征。

WillClosedTomorrow_TodayIsSat	Sales
0	7179.393519
1	5836.286592

WillClosedTomorrow_TodayIsNotSat	Sales
0	6937.021605
1	7734.233285

WasClosedYesterday_TodayIsMon	Sales
0	6727.049829
1	8169.791028

WasClosedYesterday_TodayIsNotMon	Sales
0	6934.039577
1	7603.970607

四类特殊日期下平均销量对比（0：否，1：是）

2.2．算法与方法

充分考虑给定的数据实情，在本案中，自变量为日期、促销、竞争、节假日等因素，因变量为销量，因此考虑有监督的机器学习算法进行建模。

基准模型选择线性回归（Linear Regression），线性回归是处理回归问题的基本模型形态，以 $\hat{y}_i = \hat{w}_0 + \sum_{j=1}^{p} x_{ij}\hat{w}_j$ 为打分函数（scoring function），并通常以 $L(y_i, \hat{y}_i) = \frac{1}{2n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$ 为目标函数（objective function），通过对参数 w 求一阶导数并令导数为零求解最优解，并将预测特征作用于打分函数得到预测值 y。

正式模型选择 XGBoost，XGBoost 本质上是一个优化了的分布式梯度增强库，它在 Gradient Boosting 框架下实现机器学习算法。XGBoost 提供了并行提升树的方法（GBDT，GBM），可以快速准确地解决许多数据科学问题[ii]。XGBoost 目标函数的推导过程如下所示[iii]：

$$
\begin{aligned}
&Obj = \sum_{i=1}^{n} L\left(y_i, \hat{y}_i^t\right) + \sum_{k=1}^{t}\Omega\left(f_k\right) \\
&\Rightarrow \sum_{i=1}^{n} L\left(y_i, \hat{y}_i^{t-1} + f_t\left(x_i\right)\right) + \Omega\left(f_t\right) \\
&\approx \sum_{i=1}^{n} L\left[y_i, \hat{y}_i^{t-1} + \left(\hat{y}_i^{t-1}\right)' f_t\left(x_i\right) + \frac{1}{2}\left(\hat{y}_i^{t-1}\right)'' f_t^{\,2}\left(x_i\right)\right] + \Omega\left(f_t\right) \\
&\Rightarrow \sum_{i=1}^{n}\left[L\left(y_i, \hat{y}_i^{t-1}\right) + \frac{\partial L}{\partial \hat{y}_i^{t-1}}\cdot f_t\left(x_i\right) + \frac{1}{2}\cdot\frac{\partial^2 L}{\partial\left(\hat{y}_i^{t-1}\right)^2}\cdot f_t^{\,2}\left(x_i\right)\right] + \Omega\left(f_t\right) \\
&\Rightarrow \sum_{i=1}^{n}\left[L\left(y_i, \hat{y}_i^{t-1}\right) + g_i f_t\left(x_i\right) + \frac{1}{2}h_i f_t^{\,2}\left(x_i\right)\right] + \Omega\left(f_t\right) \\
&\Rightarrow \sum_{i=1}^{n}\left[C + g_i f_t\left(x_i\right) + \frac{1}{2}h_i f_t^{\,2}\left(x_i\right)\right] + \Omega\left(f_t\right) \\
&\Rightarrow \sum_{i=1}^{n}\left[g_i f_t\left(x_i\right) + \frac{1}{2}h_i f_t^{\,2}\left(x_i\right)\right] + \Omega\left(f_t\right) \\
&\Rightarrow \sum_{i=1}^{n}\left[g_i w_{q(x_i)} + \frac{1}{2}h_i w_{q(x_i)}^2\right] + \gamma T + \frac{\lambda}{2}\sum_{j=1}^{T} w_j^2 \\
&\Rightarrow \sum_{j=1}^{T}\left[\left(\sum_{i\in I_j} g_i\right) w_j + \frac{1}{2}\left(\sum_{i\in I_j} h_i + \lambda\right) w_j^2\right] + \gamma T \\
&\Rightarrow \sum_{j=1}^{T}\left[G_j w_j + \frac{1}{2}\left(H_j + \lambda\right) w_j^2\right] + \gamma T
\end{aligned}
$$

目标函数由损失函数 L（lost function）和正则化项 Omega（regularization）组成，通过在函数空间中应用泰勒二阶展开式最终把目标函数构造为这样一个求和的形式：每棵决策树每个叶子节点（j）一阶导的求和项（G）与每个叶子节点的得分（w）的乘积，加上二阶导的求和项（H）与正则化超参数（lambda）与得分平方的乘积，再加上正则化超参数（gamma）与叶子总数（T）的乘积。打分函数通过对目标函数求一阶导数令导数为零推导出 w，从而更新目标函数的形式，推导过程如下：

$$\frac{\partial Obj}{\partial w_j} = G_j + \left(H_j + \lambda\right) w_j = 0$$

$$\Rightarrow w_j = -\frac{G_j}{H_j + \lambda} = w_{q(x_i)} = f_t\left(x_i\right)$$

$$\Rightarrow \arg\min_{G,H} Obj = -\frac{1}{2}\sum_{j=1}^{T}\frac{G_j^2}{H_j + \lambda} + \gamma T$$

由于在现实中，可能的决策树结构是无穷的，因此实际上不可能枚举所有可能的树结构，所以通常采用贪心策略来生成决策树的每个节点，算法计算的理论步骤如下：

1、 确定损失函数的形式，回归问题选择最小二乘损失函数，二分类问题选择对数损失函数，多分类问题选择交叉熵损失函数；

2、 通过贪心策略生成新的决策树，计算每个叶子节点对应的预测值；

3、 把新生成的决策树添加到模型中；

4、 重复步骤 2-3，直到每个叶子节点只剩下一个样本，或达到具体超参数的限制条件训练终止。

在本案中，所用到的特征类型有连续性、离散型，其中离散型包含二值离散型和多属性离散型，在模型训练时连续性特征采用加 1 对数变换、二值离散型采用 LabelBinarizer 变换、多属性离散型采用 OneHotEncoder 变换。正如 XGBoost 的缔造者陈天奇所说，在离散型特征的属性量在[10,100]之间时，OneHotEncoder 变换往往能发现特征之间的交互效应[iv]。

2.3. 基准测试

运用线性回归进行基准测试，数据集划分与测试结果如下：

训练集：2013-01-01 ~ 2015-06-14，数据量：79.8 万，RMSPE：0.3033

验证集：2015-06-15 ~ 2015-07-31，数据量：4.6 万，RMSPE：0.2274（6 周）

根据基础测试，确定模型效果的基准阈值为 0.2274。

3. 方法

3.1. 数据预处理

1、载入数据源，包含 train.csv， test.csv， store.csv，将 store 信息按店铺 ID 关联到 train 和 test；

2、数据变换与基本特征构造：

2.1、将 StoreType，Assortment 的原始值映射为[0,1,2,3,4]；

2.2、由 Date 新增字段：Year（年）、Month（月）、Day（日）、DayOfWeek（星期）、WeekOfYear（年中周）、DayOfYear（年中日）、Tenday（旬），WeekOfYear 和 DayOfYear 做对数变换，用连续属性体现长期趋势效应；

2.3、由 CompetitionOpenSinceYear、Year、Month、Day 新增 CompetitionOpen（竞争对手是否开业天数，1：是，0：否）；竞争对手若未开业，CompetitionDistance（竞争对手距离）置为 0；

2.4、根据 Promo2SinceYear、Promo2SinceWeek、Year、WeekOfYear 新建 Promo2Open（长期促销是否开始，1：是，0：否），再根据 Month、PromoInterval 和 Promo2Open 创建新字段 InPromo2（是否处在长期促销中，1：是，0：否）；

2.5、根据 DayOfWeek 和 Open 为每家店铺新增字段：WillClosedTomorrow_TodayIsSat（明日闭店且今日是周六）、WillClosedTomorrow_TodayIsNotSat（明日闭店且今日非周六）、WasClosedYesterday_TodayIsMon（昨日闭店且今日是周一）、WasClosedYesterday_TodayIsNotMon（昨日闭店且今日非周一）；

3、缺失值处理。CompetitionDistance，CompetitionOpenSinceMonth，CompetitionOpenSinceYear，Promo2SinceWeek，Promo2SinceYear，PromoInterval，Open 等字段含有缺失值，Open 缺失用 1 填补默认视为营业，PromoInterval 缺失用空格填补便于构造 InPromo2，其余存在缺失可忽略，因为不影响特征的计算，如 Promo2SinceYear 等缺失表示不存在长期促销，不会妨碍 Promo2Open 的计算；

4、深度特征构造，在探索性分析中根据假设检验的效果构造，参照上文（2.1 数据研究与探索性可视化）：

4.1、StoreSales（店铺销售规模分类），取值[0,1,2,3,4,5]；

4.2、CompetitionState（竞争格局），自始至终都没有竞争对手的门店（0）、自始至终都有竞争对手的门店（1）、此前没有后来进入竞争对手的门店（2）；

4.3、CompetitionDistanceStr（竞争对手距离离散化），无竞争对手（0）、短距离（(0,350 米]）、中距离（(350 米,4000 米]）、远距离（4000 米以上）；

4.4、DayStr（销量极值日特殊标记），[1,2,3,16,17,18,30]日记为“异常高”，[11,12,24,25,26]日记为“异常低”；

4.5、WeekOfYearStr（年中周离散化），18 周以前且偶数周（0）、18 周以前且奇数周（1）、19 周以后且偶数周（2）、19 周以后且奇数周（3）

4.6、DayOfYearOutlier（日期值离群值），日期对应的销量高于 0.5 倍四分位级差记为 1，低于记为 2，正常为 0；

4.7、DayOfYearSlopeStr（日期斜率离群值），日期对应的销量斜率高于 1 倍四分位级差记为 1 低于记为 2，正常为 0。

3.2．实施与改进

使用正式算法 XGBoost 进行模型训练、验证和测试，实施过程中只针对开业的数据剔除闭店的数据，数据集划分如下：

训练集：2013-01-01 ~ 2015-06-14

验证集：2015-06-15 ~ 2015-07-31（6 周）

来源：train.csv

测试集：2015-08-01 ~ 2015-09-17（6 周）

来源：test.csv

最初的解决方案是结合所有店铺数据训练一个整体模型，主要考虑到这样模型结构简洁，处理速度较快，但是经过验证发现模型效果相对于基准模型提升幅度有限。因此，最终方案是分单体和整体两类模型，单体模型针对每个店铺单独训练一个模型，整体模型针对所有店铺训练一个整体模型，最终的预测结果单体模型占 80%权重，整体模型占 20%权重。两类模型所用字段特征如下所示：

字段 英文名	字段 中文名	字段 类型	字段 处理	适用 模型
Sales	销量	float	log	

WeekOfYear	年中周	float	log	
DayOfYear	年中日	float	log	
CompetitionDistance	竞争对手距离	float	log	仅整体
Promo	是否短期促销	int	LabelBinarizer	
InPromo2	是否长期促销	int	LabelBinarizer	
CompetitionOpen	竞争对手是否营业	int	LabelBinarizer	
WillClosedTomorrow_TodayIsSat	明日闭店且今日是周六	int	LabelBinarizer	
WillClosedTomorrow_TodayIsNotSat	明日闭店且今日非周六	int	LabelBinarizer	
WasClosedYesterday_TodayIsMon	昨日闭店且今日是周一	int	LabelBinarizer	
WasClosedYesterday_TodayIsNotMon	昨日闭店且今日非周一	int	LabelBinarizer	
SchoolHoliday	是否学校放假	int	LabelBinarizer	
StateHoliday	是否州假	int	OneHotEncoder	
StoreType	店铺类型	int	OneHotEncoder	仅整体
Assortment	店铺分类	int	OneHotEncoder	仅整体
StoreSales	销售分类	int	OneHotEncoder	仅整体
Year	年	int	OneHotEncoder	
Month	月	int	OneHotEncoder	
Tenday	旬	int	OneHotEncoder	
Day	日	int	OneHotEncoder	
DayStr	日离散	int	OneHotEncoder	
DayOfWeek	星期	int	OneHotEncoder	
WeekOfYearStr	年中周离散	int	OneHotEncoder	
DayOfYearOutlier	日期值离群值	int	OneHotEncoder	
DayOfYearSlopeStr	日期斜率离群值	int	OneHotEncoder	
CompetitionState	竞争格局	int	OneHotEncoder	仅整体
CompetitionDistanceStr	竞争距离离散	int	OneHotEncoder	仅整体

单体模型的训练与调参步骤如下：

1、 提取每个店铺的数据划分训练集和验证集，初始值：分类器=2000，学习率=0.1，gamma=0，

max_depth=7，min_child_weight=0.001，subsample=0.9，colsample_bytree=0.9，reg_alpha=0，reg_lambda=1，max_delta_step=0，scale_pos_weight=1，objective="reg:linear"，eval_metric="rmspe"，交叉验证=5 折；

2、为每个店铺训练更新最优分类器个数；

3、交叉验证选择最优学习率：[0.1, 0.2, 0.3]；

4、交叉验证学习最优 max_depth 和 min_child_weight，"max_depth": [3,5,7,9,11]，"min_child_weight": [0.001, 0.01, 0.1, 1]；

5、交叉验证学习最优 gamma，"gamma": [0, 0.5, 1, 1.5, 2, 2.5]；

6、为每个店铺训练调整最优分类器个数；

7、交叉验证学习最优 subsample 和 colsample_bytree，"subsample": [0.6,0.7,0.8,0.9]，"colsample_bytree": [0.6,0.7,0.8,0.9]；

8、交叉验证学习最优 reg_alpha，"reg_alpha": [0, 1, 2, 3]；

9、交叉验证学习最优 reg_lambda，"reg_lambda": [1,3,5,7]；

10、交叉验证学习最优 max_delta_step 和 scale_pos_weight，"max_delta_step": [0, 1, 3, 5]，"scale_pos_weight": [1, 3, 5, 7]；

11、为每个店铺训练调整最优分类器个数；

12、为每个店铺训练调整学习率，在原有基础上微调；

13、用优化好的参数训练模型；

14、用训练好的模型在测试集上实施预测，输出单体模型预测结果。

整体模型的训练与调参步骤如下：

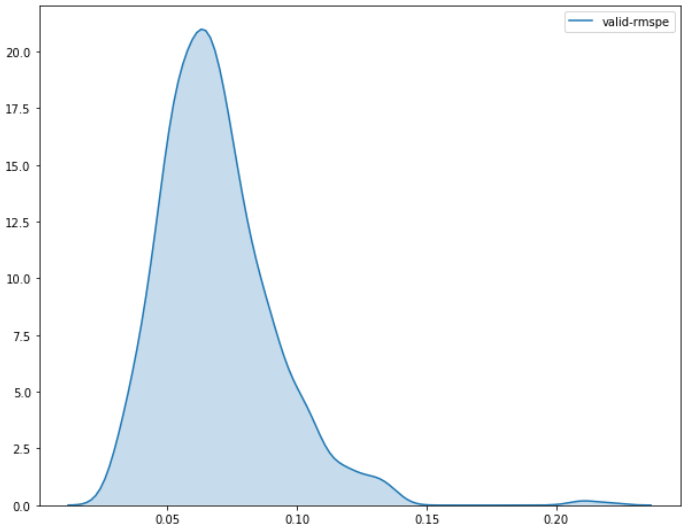
1、基于整体数据划分训练集和验证集，分类器个数=20000，学习率=0.01，max_depth = 9，subsample = 0.9， colsample_bytree = 0.9，early_stopping_rounds=1000，其余使用默认参数；

2、用训练好的模型在测试集上实施预测，输出整体模型预测结果。

4. 结果

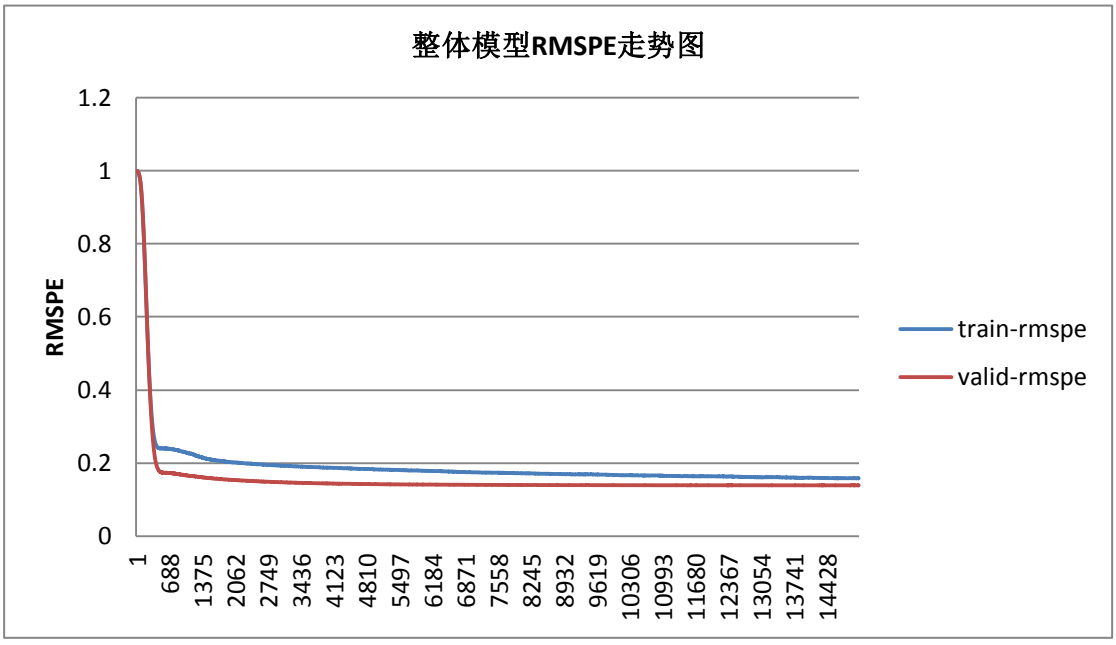
4.1. 模型评估与验证

针对单体模型，输出每个子模型在验证集上的 rmspe 值，valid-rmspe 均值 0.06925，标准差 0.0222，通过可视化可见绝大多数子模型的误差集中在 0.06 附近，少数店铺预测误差较大，超过 0.2，分布图如下：



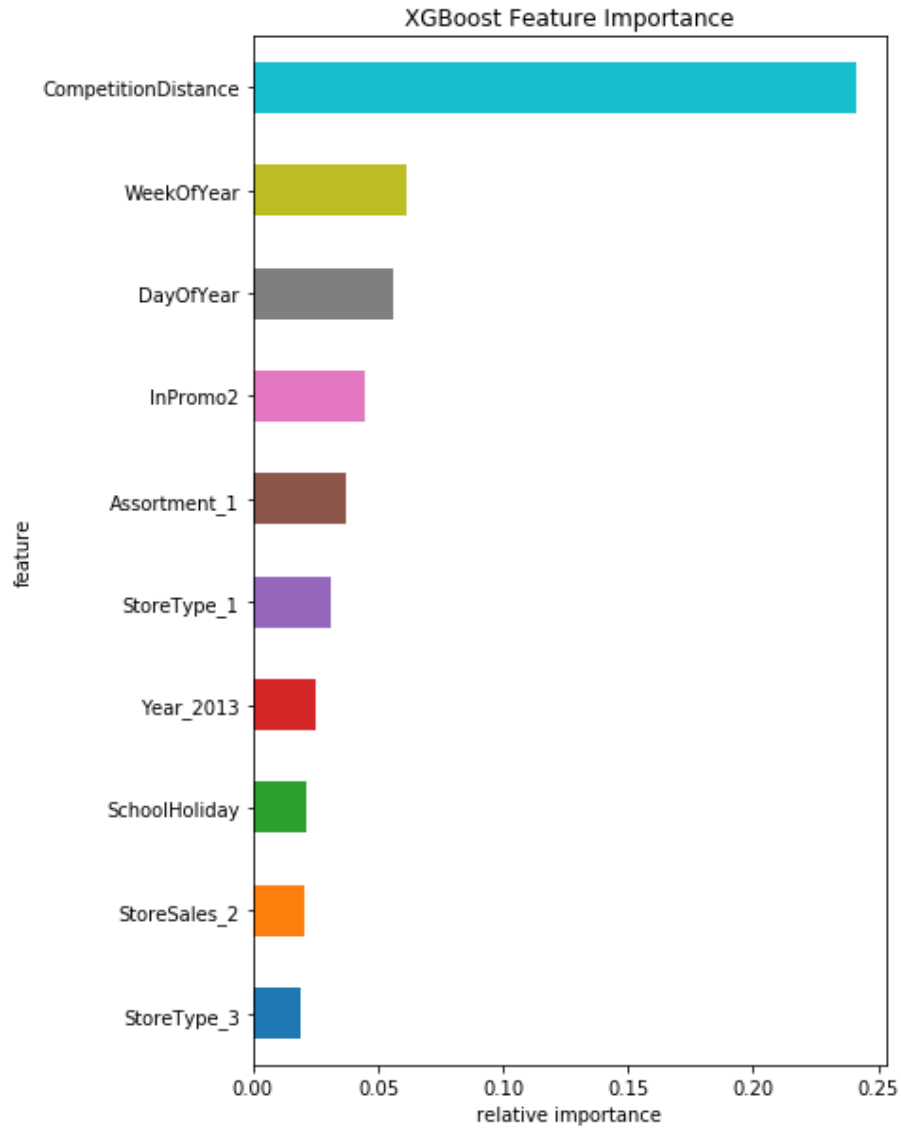
单体模型验证集 RMSPE 分布图

针对整体模型，输出整体的训练与验证结果，模型于第 14094 步达到最优迭代次数， train-rmspe = 0.160247，valid-rmspe = 0.138307，模型迭代走势图如下：

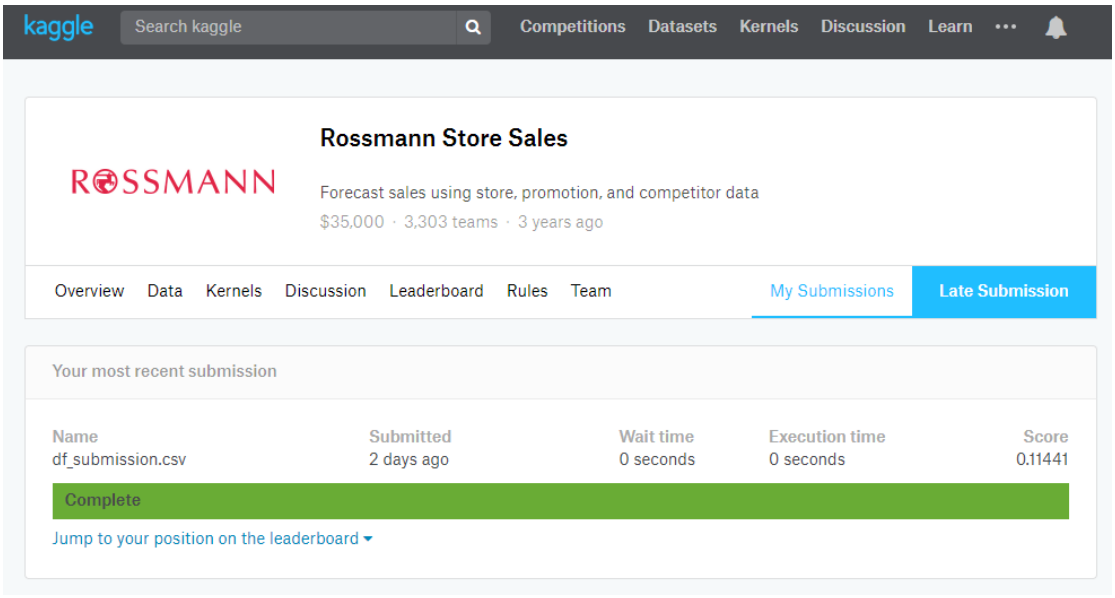


两类模型验证效果均优于基准模型，单体模型误差比基准模型降低了 0.15815，整体模型的误差比模型降低了 0.089093，说明该解决方案能够解决本案预测问题。

从特征重要性程度来看，竞争对手距离、年中周、年中日、是否处在长期促销中等 10 个特征是影响销量最重要的特征，Top10 特征重要性排序如下：



最后，分别用单体模型和整体模型对测试集开业数据实施预测，闭店数据销量统一置为 0，输出加权预测结果，其中单体模型权重占 80%，整体模型权重占 20%，提交至 Kaggle 官方网站，输出测试结果为 0.11441，等同于实际比赛排名的第 113 位（3303 位），位于 3.4%，截图如下：



113	▾60	tks		0.11441	2	3y

5. 思考与改进

针对本项目，笔者认为影响销量的因素除了给定的促销、竞争对手、节假日因素以外，作为线下实体店，天气与气候因素对销量的影响也很重要，例如：平均气温、气压、最高温度、最低温度、平均湿度、降雨量等因素，如果能够获取每个店铺所属城市的历史天气数据和未来天气预测，对提升预测准确率会有进一步的帮助。

[i] https://www.kaggle.com/c/rossmann-store-sales
[ii] https://xgboost.readthedocs.io/en/latest/
[iii] Chen T, Guestrin C. Xgboost: A scalable tree boosting system[C]//Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining. ACM, 2016: 785-794
[iv] https://github.com/szilard/benchm-ml/issues/1