机器学习纳米学位

毕业项目 Joe 优达学城

2019 年 3 月 13 日

毕业论文 UDA 学员 李欣

2019.3.13 农历（己亥年）猪年 北京

I. 问题的定义

1.1 项目概述

这是一个来自于 Kaggle 的真实竞赛项目，项目要求预测 Rossmann 商店的日常销售。Rossmann 是一个在 7 个欧洲国家经营着 3,000 多家商店的连锁企业。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额，结果的准确性可能会有很大差异。

Kaggle 的比赛要求：预测遍布德国各地的 1,115 家商店未来 6 周的销售额。可靠的销售预测可以使商店经理能够创建有效的员工时间表，从而提高生产力和动力。通过帮助 Rossmann 创建一个强大的预测模型，您将帮助商店经理专注于对他们最重要的事情：他们的客户和他们的团队！

Kaggle 提供了该项目所需要的全部数据集，包括有 1,115 家 Rossmann 商店的历史销售数据(2013.01.01-2015.07.31)：其中，训练集（train.csv）1017209 条；测试集（test.csv）41088 条；商店的特征集（store.csv）1115 条。

1.2 问题陈述

该项目的具体任务是在 1,115 家 Rossmann 商店的历史销售数据中预测测试集的“Sales”列，这属于回归类的需求。可以通过构建一个有监督学习类的模型来预测 6 周的销售数据，在利用 XGBoost 学习建模后，通过减少预测值（$\hat{y}$）与实际值（y）之间的误差来解决项目需求。

首先，搭建一个实现项目基本需求的基础模型，为之后的模型优化提供基准。

其次，使用 XGBoost 算法训练数据，再依据 Kaggle 推荐的衡量方式：RMSPE 来评估模型，最后选出最佳模型再对测试集的商店销售额进行预测。

最后，将预测提交到数据 Kaggle 上参加竞赛，目标：进入 leaderboard private 的 top 10%。实际对于测试集的 Rmspe 评分 ：至少要达到 0.11773 分。

1.3 评价指标

使用 Kaggle 的 RMSPE 函数来验证真实的销售数据与预测数据的差异性，主要是先利用特征工程从杂乱的数据中遴选出有效特征，再使用回归模型训练数据集，最后使用均方根百分比误差函数 Rmspe 来验证训练结果。RMSPE：

$$RMSPE = \sqrt{\frac{1}{n}\Sigma_{i=1}^{n}\left(\frac{yi-\hat{yi}}{yi}\right)^2}$$

均方根百分比误差函数中：yi 表示真实的销售数据， ŷ 是预测数据，该函数计算预测值和实际值之间的差额平方，并将结果累加起来求平均，最后再开方，以此衡量预测值与实际值之间差异。

II. 分析

2.1 数据的探索

根据 Kaggle 提供的数据，现有 1,115 家 Rossmann 商店的历史(2013.01.01-2015.07.31)销售数据（如下表，训练集：train.csv；测试集:test.csv；商店的特征集:store.csv）。 测试集的“Sales”列就是实际值（y）。 数据集中的某些商店暂时关闭以进行翻新。由于指标中 yi 不能为零，因此在数据清理过程中将排除实际销售额为零的数据。

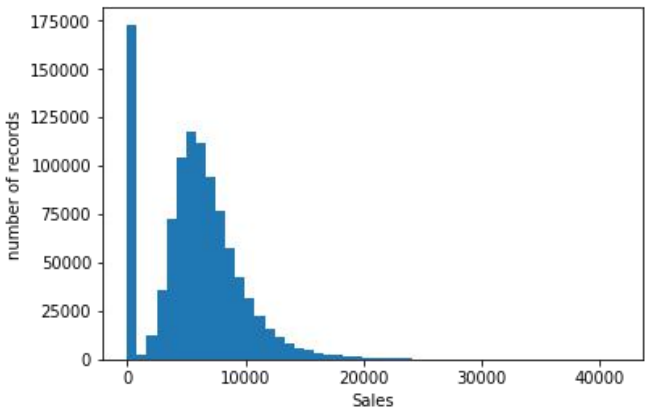
train.csv 包含 9 个特征,共 1017209 个样本;test.csv 有 8 个特征，41088 个样本，测试集比训练集少 Sales 字段（Sales 字段即预测目标）和 Customers 特征同时增加 Id 字段（表示记录编号）。Store.csv 有 10 个特征，1115 条记录。训练集没有缺失值， 测试集的部分字段有缺失值，需要填充。对于异常值需要转换数据类型。

数据集	内 容	数 量
train.csv	销售日、销售额、顾客数、促销、营业、国家假期、学校假期	1017209
test.csv	销售日、顾客数、促销、营业、国家假期、学校假期	41088
store.csv	商店的特征信息，包含：商店类型，等级，竞争者信息，促销活动	1115

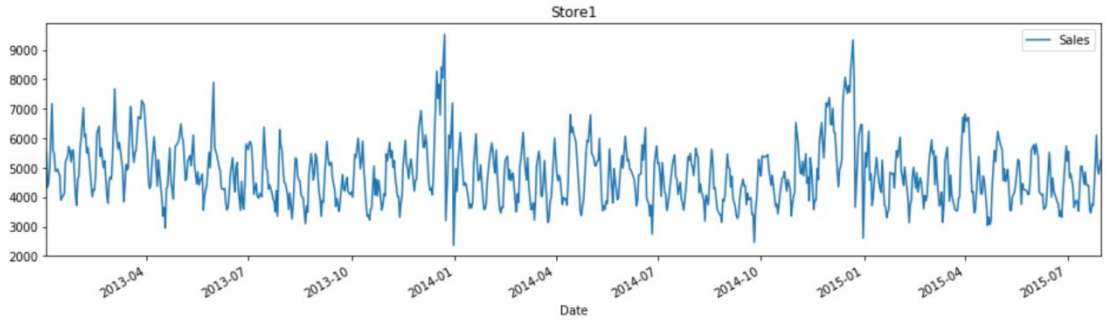
特征名	含义
Store	商店编号，1-1115
Sales	日销售额
Customers	日顾客数
Open	是否营业，0 关门/1 开门
StateHoliday	州假日，a=全部假日/b=复活节/c=圣诞节
SchoolHoliday	学校假日是否开门，0 关门/1 开门
StoreType	商店类型 ， a,b,c,d
Assortment	商店评级 ， a,b,c
CompetitionDistance	与最近的竞争者之间的距离
CompetitionOpenSinceYear	最近的竞争者开业的年份
CompetitionOpenSinceMonth	最近的竞争者开业的月份
Promo	当天是否促销，0 否/1 是
Promo2	商店是否参与长期促销，0 否/1 是
Promo2SinceYear	商店参与长期促销开始的年份
Promo2SinceWeek	商店参与长期促销开始的日历周
PromoInterval	长期促销的月份，Feb/May/Aug/Nov

2.2 探索性可视化

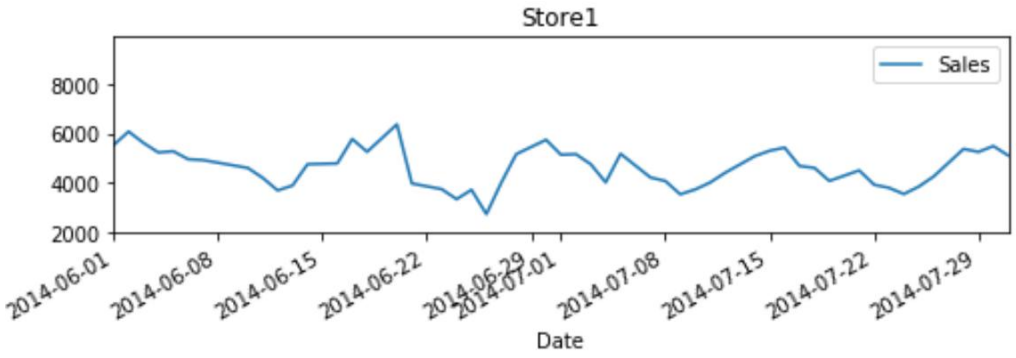
2.2.1 本项目模型的预测标的是商店的销售额，即训练集的 Sales 字段，下图是 Sales 的可视化数据分布：

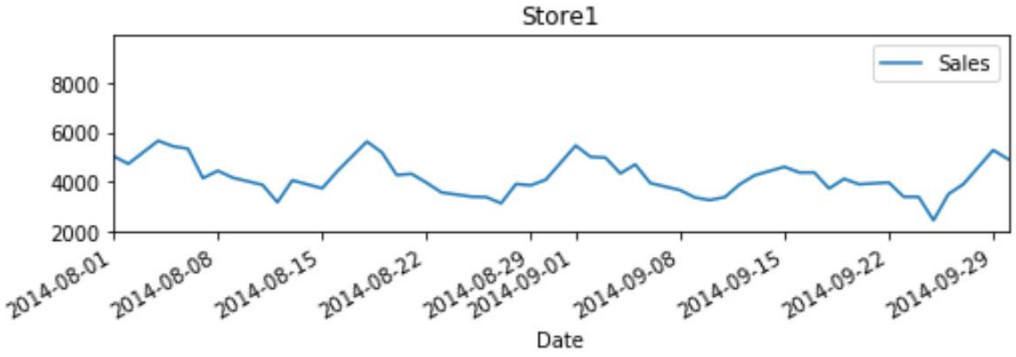


2.2.2 查看 2013-2015 年的销售数据，可以看到销量是有规律的、周期性变化的。一年中，因为季节性因素和促销日、节假日等原因，11，12 月份销量要高于其他月份：

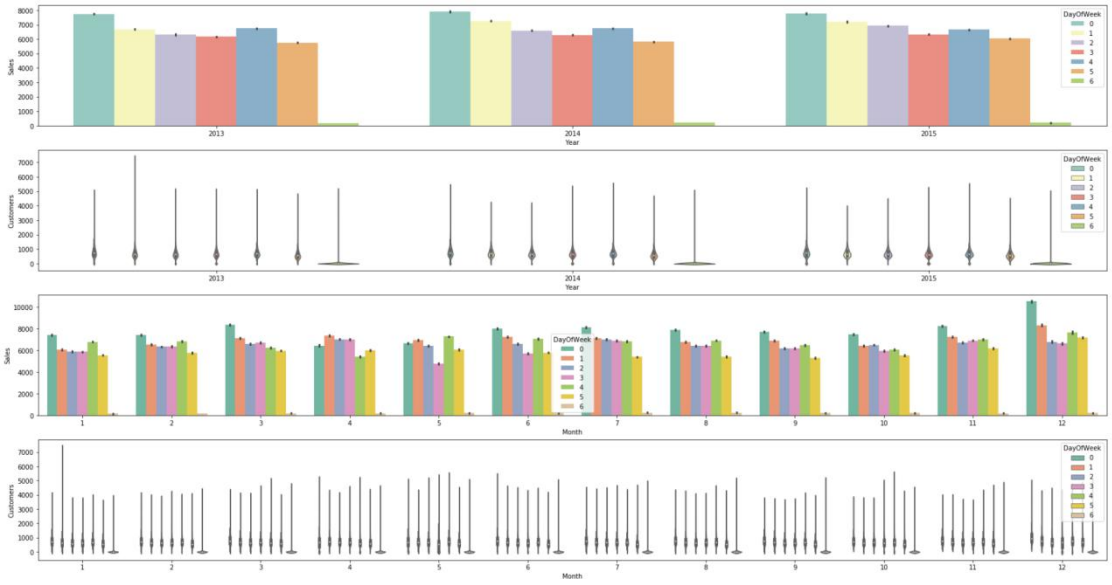


2.2.3 查看 2014.6.1-2014.9.30 之间的店铺销量变化（如下图），两个图的销量走势类似。本项目需要预测 6 周的销售数据，即 2015 年的 8-9 月份。因此，下面可以把 2015 年的 6-7 月份的销售数据作为 hold-out 划分，以便验证预测数据。

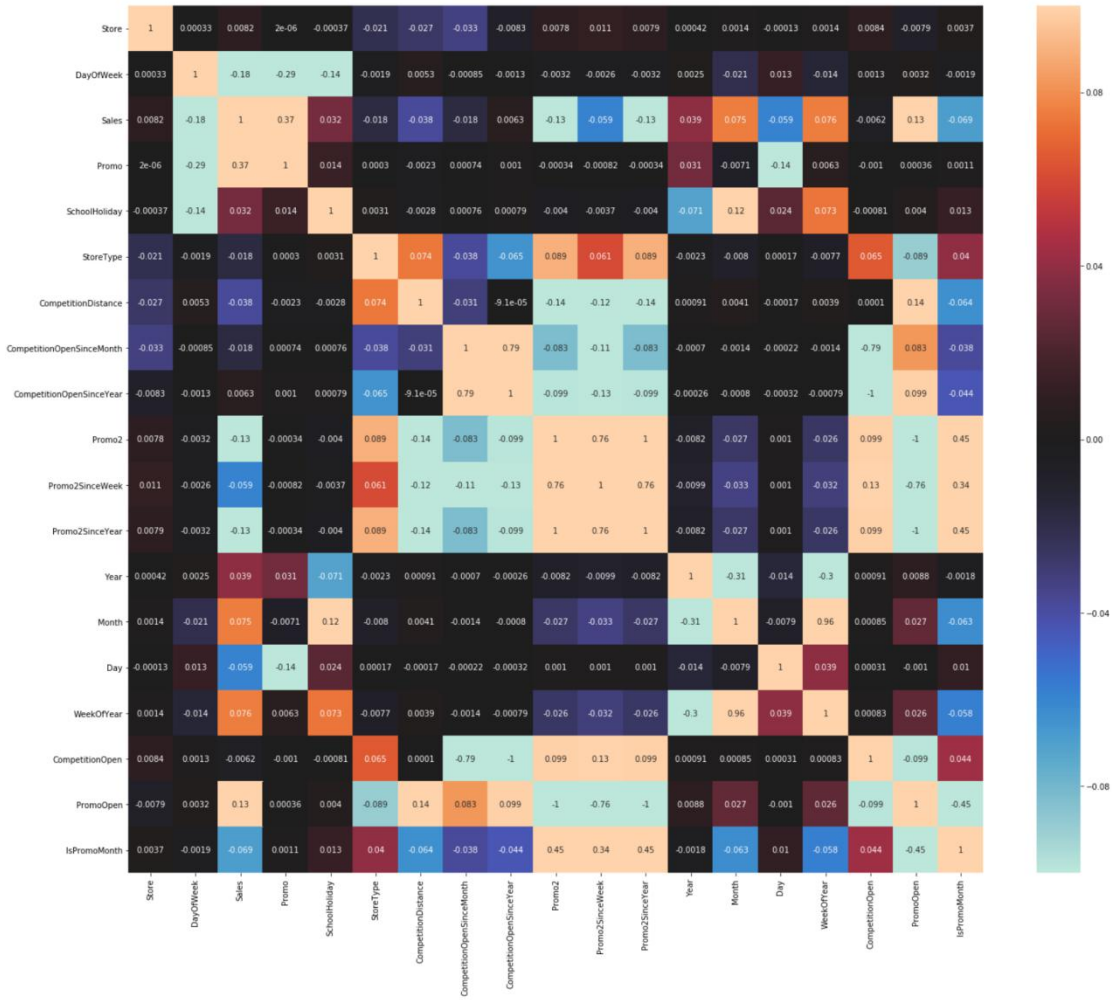




2.2.4 查看一年中每月对应的七天销售额的柱状图与其琴箱图。数据可视化后显示周日都是关店状态。



2.2.5 分析训练数据集中特征相关性以及特征与'Sales'标签相关性



2.3 算法和技术

2.3.1 解决方案

本项目属于回归类课题，且有大量离散型数据，因此使用 GBDT 搭建模型。目前，业界主流的 GDBT 框架一个是微软的 LightGBM，以及陈天奇 XGBoost。对比之下（对比图如下），虽然 LightGBN 效率更高，但 XGBoost 更加成熟且更容易达到高精确度，因此本项目决定使用 XGBoost 算法搭建回归模型。

	XGBoost	Light BGM	
Parameters Used	max_depth: 50 learning_rate: 0.16 min_child_weight: 1 n_estimators: 200	max_depth: 50 learning_rate: 0.1 num_leaves: 900 n_estimators: 300	
Training AUC Score	0.999	Without passing indices of categorical features	Passing indices of categorical features
		0.992	0.999
Test AUC Score	0.789	0.785	0.772
Training Time	970 secs	153 secs	326 secs
Prediction Time	184 secs	40 secs	156 secs
Parameter Tuning Time (for 81 fits, 200 iteration)	500 minutes	200 minutes	

2.3.2 XGBoost 算法

XGBoost 是 eXtreme Gradient Boosting 的缩写，即极端梯度提升树，是梯度提升机器算法（GBM）的扩展。XGBoost 在每轮迭代中生成一棵新的回归树，并综合所有回归树的结果，使预测值越来越逼近真实值。该算法有很强的泛化能力，其学习策略是每次学习当前的树，找到当前最佳的树模型加入到整体模型中；其核心是每棵树通过学习之前所有树的残差，并将所有结果累加。

在 XGBoost 中下一次预测的 yt 值可以表达成上一次预测值加上一个修正即 yt= yt-1 + f(x)，带入到损失函数中，再进行泰勒二次展开，然后进行参数的定义并将目标函数进行改写，最后得到目标函数：

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

XGBoost 有三种类型的参数: general parameters，booster parameters 和 task parameters。general parameters，常规参数用于配置提升器，通常是树模型或线性模型，提升器参数取决于选择的提升器；booster parameters，学习任

务的参数决定了学习场景，例如回归任务可以使用不同的参数进行排序相关的任务；命令行参数的行为与 XGBoost 的 CLI 版本相关。

本项目使用 XGBoost 自带的 API，其参数设置使用 key-value 字典的方式存储配置：params = { }

参数名称	默认值	参数意义	本例初始值
'booster'	gbtree	*弱学习器类型*	gbtree
'objective'	multi:softmax	*多分类的问题*	"reg:linear"
'eta'	0.007	*如同学习率*	0.03
'num_class'	10	*类别数，与 multisoftmax 并用*	10
'gamma'	0.1	*控制剪枝的参数,越大越保守，一般 0.1、0.2*	0.1
max_depth	12	*构建树的深度，越大越容易过拟合*	10
'lambda'	2	*控制模型复杂度的权重值的 L2 正则化项参数，参数越大，模型越不容易过拟合*	2
'subsample'	0.7	*随机采样训练样本*	0.7
'colsample_bytree'	0.7	*生成树时进行的列采样*	0.7
'silent'	1	*运行信息输出设置*	1
Seed	100	*随机数种子*	10
'nthread'	4	*CPU 线程数*	4

2.4 基准模型

本项目的基准模型是依据 Kaggle 的项目需求，至少达到 leaderboard private 的 top 10%以内。实际对应测试集的 Rmpse 评分 ，需要至少达到 0.11773 分。

III. 方法

3.1 数据预处理

3.1.1 查补缺失值

训练集没有缺失值， 测试集的部分字段有缺失值，需要填充。缺失数据都来自于第 622 号店铺，从周 1 到周 6 而且没有假期，因此将该店铺的状态更改为正常营业。

店铺竞争数据缺失的原因不明，且数量较多，以用中值或者 0 来填充，后续的实验发现以 0 填充的效果更好。店铺促销信息的缺失是因为没有参加促销活动，因此也以 0 填充。

3.1.2 数据整理

合并训练集和测试机的 store 信息；留出最近的 6 周数据作为 hold_out 数据集进行测试；因销售额为 0 的记录不计入评分，因此，只采用店铺状态为营业，且销售额大于 0 的店铺数据进行训练。

3.2 特征工程

3.2.1 特征处理与转化，定义特征处理函数

1.将存在其他字符表示分类的特征转化为数字；

2.将时间特征进行拆分和转化，并加入'WeekOfYear'特征；

3.新增'CompetitionOpen'和'PromoOpen'特征,计算某天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；

4.将'PromoInterval'特征转化为'IsPromoMonth'特征,表示某天某店铺是否处于促销月，1表示是，0表示否；

5.使用特征处理函数对训练，保留以及测试数据集进行特征转化；

6.删掉训练和保留数据集中不需要的特征；

7.分析训练数据集中特征相关性以及特征与'Sales'标签相关性；

8.拆分特征与标签，并将标签取对数处理；

9.删掉测试集中对应的特征与训练集保持一致；

3.3 模型构建

3.3.1 模型初始化构建 使用如下参数构建模型

```
params = {"objective": "reg:linear",
          "booster" : "gbtree",
          "eta": 0.03,          "max_depth": 10,
          "subsample": 0.9,
          "colsample_bytree": 0.7,
          "silent": 1,
          "seed": 10
          }
num_boost_round = 6000
dtrain = xgb.DMatrix(ho_xtrain, ho_ytrain)
dvalid = xgb.DMatrix(ho_xtest, ho_ytest)
watchlist = [(dtrain, 'train'), (dvalid, 'eval')]
```

3.3.2 初始模型训练

初始模型迭代了 3185 次，共用时约 1.2 小时。最佳得分如下：

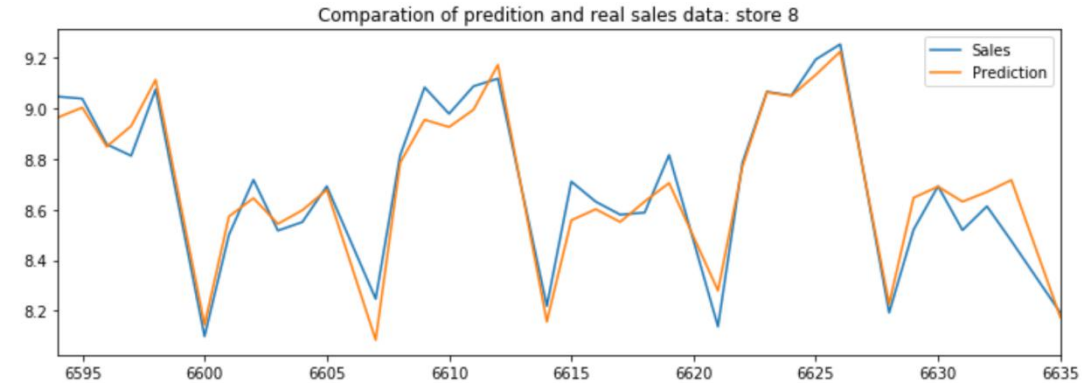
Stopping. Best iteration:

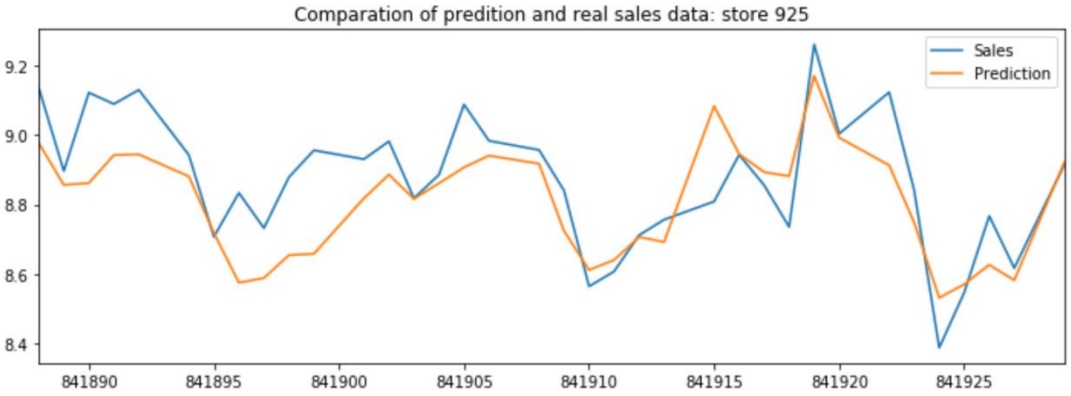
[3085] train-rmse:0.066367 eval-rmse:0.117232 train-rmspe:0.070345 eval-rmspe:0.127409

Training time is 4314.224169 s.

3.4 模型结果分析

3.4.1 分析初始模型保留数据集中任意三个店铺的预测结果如下：





3.4.2 分析偏差最大的 10 个预测结果：

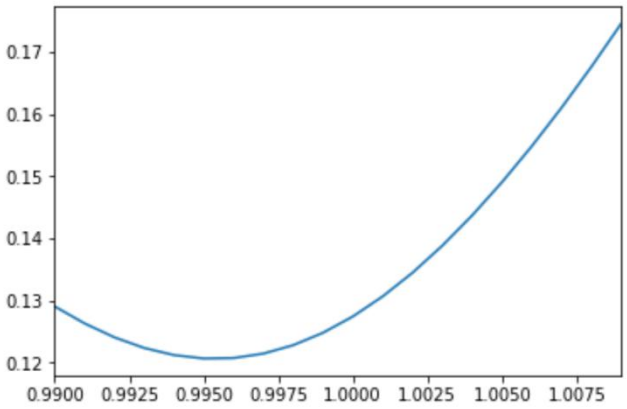
eMonth	CompetitionOpenSinceYear	...	Day	WeekOfYear	CompetitionOpen	PromoOpen	IsPromoMonth	Sales	Prediction	Ratio	Error	Weight
6.0	2009.0	...	10	28	73.0	24187.00	0	6.920672	8.553830	1.235983	0.235983	0.809073
8.0	2003.0	...	26	26	142.0	42.25	0	7.260523	8.596755	1.184041	0.184041	0.844565
0.0	0.0	...	1	27	24187.0	67.50	0	8.174139	9.543799	1.167560	0.167560	0.856487
0.0	0.0	...	22	26	24186.0	67.25	0	10.634701	9.167545	0.862041	0.137959	1.160038
4.0	2005.0	...	4	27	123.0	2.25	0	9.596215	8.325567	0.867589	0.132411	1.152620
6.0	2009.0	...	29	27	72.0	24186.75	0	10.280622	9.041227	0.879444	0.120556	1.137083
6.0	2009.0	...	4	27	73.0	24186.75	0	7.406711	8.299530	1.120542	0.120542	0.892425
4.0	2005.0	...	6	28	123.0	2.50	0	10.215777	8.998613	0.880855	0.119145	1.135261
8.0	2003.0	...	27	26	142.0	42.25	0	7.714677	8.586529	1.113012	0.113012	0.898463
9.0	2006.0	...	25	30	106.0	24187.50	0	7.360104	8.185041	1.112082	0.112082	0.899214

通过分析以上结果，可以看出初始模型已基本预测出 hold-out 数据集的销售趋势，但是相对真实值，模型的预测值有些偏高。从对偏差数据分析来看，偏差最大的 3 个数据也是明显偏高。因此，可以将 hold-out 数据集作为标准对模型进行偏差校正。

3.4 模型优化与完善

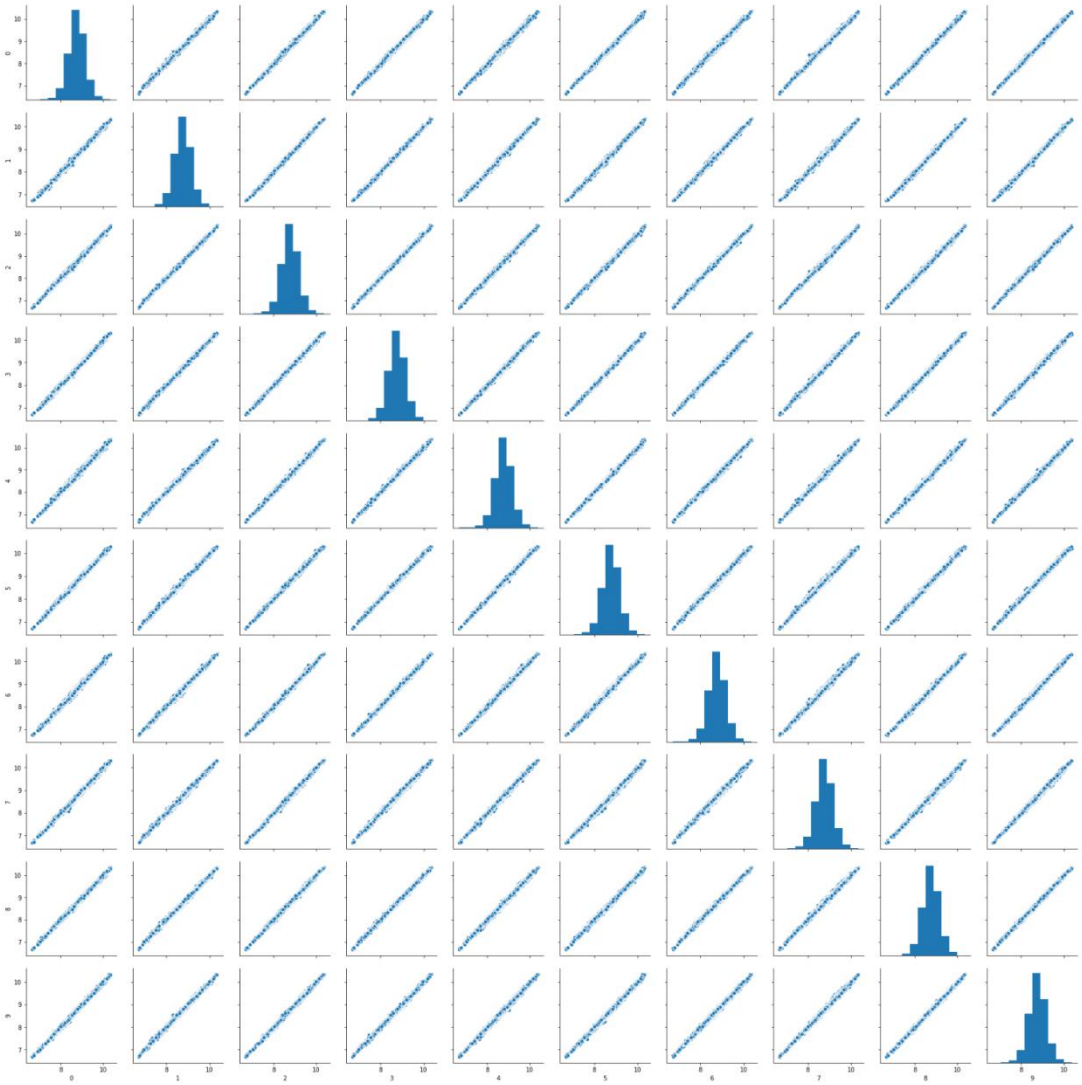
3.4.1 偏差校正分析

当校正系数为 0.995-0.9975 时，相对于初始模型的 RMSPE 得分最优。由于店铺特性各异，因此，针对不同的店铺模型采取不同的偏差校正策略。



3.4.2 偏差精准校正

使用不同的 seed 训练 10 个模型,每个模型单独进行精准偏差校正后，再进行融合，总共用时 12.3 小时。如图示，模型相关性很高，可使用简单平均融合法进行模型融合。**简单平均融合模型在 hold-out 数据集上的表现，RMSPE for mean: 0.113798，较之初始模型有了极大的提升。**



依据 hold-out 模型中的得分情况，为个别模型分配权重。**权重模型较均值模型又有了更佳的得分：RMSPE for weight: 0.112982。**

IV. 结果

4.1 模型的评价与验证

模型预测的精准度依赖于特征的重要性分析，强有效特征概况起来有四类，包括：

1. 周期性特征'Day'，'DayOfWeek'，'WeekOfYera'，'Month'等，可见店铺的销售额与时间是息息相关的，尤其是周期较短的时间特征；

2. 店铺差异'Store'和'StoreType'特征，不同店铺的销售额存在个性；

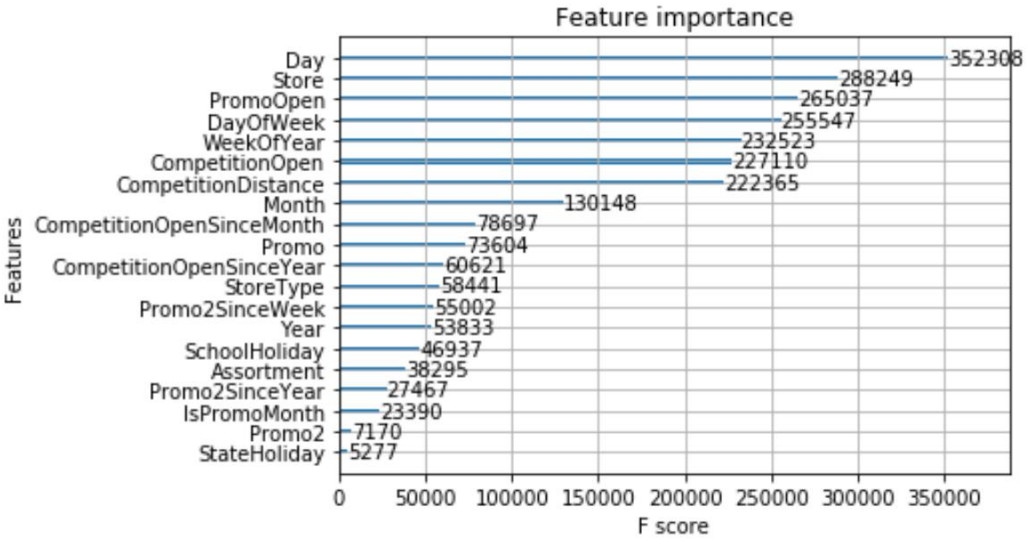
3. 短期促销（Promo)情况:'PromoOpen'和'Promo'特征，促销时间的长短与营业额相关性比较大；

4. 竞争对手相关特征包括：'CompetitionOpen'，‘CompetitionDistance'，'CompetitionOpenSinceMoth'以及'CompetitionOpenScinceyear'，竞争者的距离与营业年限对销售额有影响。

弱有效性特征主要包括两类：

5. 假期特征：'SchoolHoliday'和'StateHoliday'，假期对销售额影响不大，有可能是假期店铺大多不营业，对模型预测没有太大帮助。

6. 持续促销(Promo2)相关的特征：'Promo2'，'Prom2SinceYear'以及'Prom2SinceWeek'等特征，有可能持续的促销活动对短期的销售额影响有限。



4.2 合理性分析

从新旧模型预测结果最大的几个偏差对比的情况来看，最终的融合模型在这几个预测值上都有明显的提升，证明模型的校正和融合确实有效。

	Store	Ratio	Error	Store_new	Ratio_new	Error_new
264207	292	1.235983	0.235983	292	1.221274	0.221274
711449	782	1.184041	0.184041	782	1.172604	0.172604
827582	909	1.167560	0.167560	909	1.155518	0.155518
827591	909	0.862041	0.137959	909	0.852097	0.147903
797965	876	0.867589	0.132411	876	0.853161	0.146839
264218	292	0.879444	0.120556	292	0.871383	0.128617
264213	292	1.120542	0.120542	292	1.111659	0.111659
797963	876	0.880855	0.119145	876	0.873200	0.126800
711448	782	1.113012	0.113012	782	1.102314	0.102314
456286	501	1.112082	0.112082	501	1.096999	0.096999

V. 项目结论

5.1 结果可视化

最后使用优化后的 XGBoost 建模，并将结果上传到 kaggle 上参加竞赛，**得到评分：0.111114**

5.2 对项目的思考

本项目基于 XGBoost 算法对实体零售业销售额进行预测。论文以德国零售业 Rossmanns 公司 1115 家实体门店的商场信息和销售数据为数据源。通过在特征工程中对原始数据进行特征提取、选择和构建，筛选出用于训练的特征。为提高 XGBoost 预测模型的精度和泛化能力，通过特征工程尝试多种模型的集成学习方法和参数调优，利用偏差校正方法获得模型的最优配置方案。实验表明，模型优化后，在性能上优于 XGBoost 的初始模型。

这种基于 XGBoost 的组合模型不仅适用于对德国零售业销售额的预测，也适用于各种线性回归预测，可高效地分析客观事物的数量关系，可广泛的应用于社会经济现象变量之间的影响因素和关联的研究。

在实验过程中发现，基于树模型的 XGBoost 很适合处理表格数据，同时还拥有一些深度神经网络所没有的特性（如：模型的可解释性、输入数据的不变性、更易于调参等特点）。这些特性非常利于揭示影响市场现象的多种复杂因素之间的依存关系，准确测定现象之间的数量变动．以提高预测和控制的准确度。

5.3 需要作出的改进

按照项目的目标，本文已经实现，若需考虑在算法与技术层面的进一步完善，可在模型的加权融合方面再进行更加细致的微调，因为每个店铺都有自己的特点，要想有更精准的预测，必须针对不同的店铺再进行细致的偏差校正。而我看重的是更加贴近现实需求的改进设想：

时间序列的预测有着持续性和不间断的特点，而在本项目中的预测时长只有6周，且是指定的时段，对于预测精准度有着非常大的局限性。而在现实中的实际需求中，时间序列预测的都是当下或未知的时间线，且没有范围，例如资本市场的预测（股票/外汇/期货等）。

因此，本文中的特征工程就不适用于持续性和不间断的时间序列预测了。必须加入更多的特征信息。所以，搜寻、筛选影响市场走势的特征因子就成了该类项目重中之重的改进需求了。

参考文献

1.Rossmann Stroe Sales, https://www.kaggle.com/c/rossmann-stroe-sales

2.LightGBM, https://lightgbm.readthedocs.io/en/latest

3.XGBoost, https://XGBoost.readthedocs.io/en/latest

4. Rossmann 销售预测 Top1%, https://blog.csdn.net/aicanghai_smile/article/details/80987666

5. Model documentation 1st place https://www.kaggle.com/c/rossmann-store-sales/discussion/18024

6. entity-embedding-rossmann, https://github.com/entron/entity-embedding-rossmann

7.李航·统计学习方法[M].北京：清华大学出版社