预测 Rossmann 未来的销售额报告

姚偲潼

2018 年 10 月 11 日

一　定义

1.1 项目概述

本项目是 Kaggle 中的一个比赛项目，Kaggle 是一个为开发商和数据科学家提供举办机器学习竞赛、托管数据库、编写和分享代码的平台。

Rossmann 是欧洲的一家连锁药店。本项目需要根据 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。对未来销售额的准确预测可以使店长的工作更高效 ,也可为企业发展战略提供指引。所以预测模型在商业的应用非常广泛。

现今社会生活中方方面面都会涉及预测的问题，机器学习中的监督学习是分析预测中常用到技术。如天气预报、环境监测、金融、医学等都有成熟的应用。所以本项目也可以应用机器学习技术得以解决。

1.2 问题陈述

对于销售额的预测，可以作为监督学习的一个回归问题进行处理。监督学习主要目的是使用有类标 (lable) 的训练数据构建模型，使用经训练得到的模型对未来数据进行预测，模型可看作是因变量与自变量之间构成的联系和规律。监督学习主要两个子类，一个是分类 ,另一个是回归 ,也是本项目中应用的技术。回归主要针对预测连续值输出，项目中的销售额就可以看作是连续值输出。

本项目可将药妆店销售额看作因变量，药妆店的其他信息看作自变量，利用监督学

习技术研究因变量与自变量之间的关系模型。利用药妆店过去的销售数据及相关信息进行训练，通过测试数据不断验证改进，得出一个相对精准的销售额预测模型，便可以使用该模型应用到实际的预测中。

1.3 评价指标

通过模拟参加 Kaggle 比赛，将测试结果提交到 Kaggle 来评估模型的表现，本比赛项目中使用 RMSPE 来评价提交者的预测数据的精准度。

RMSPE 计算公式：

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

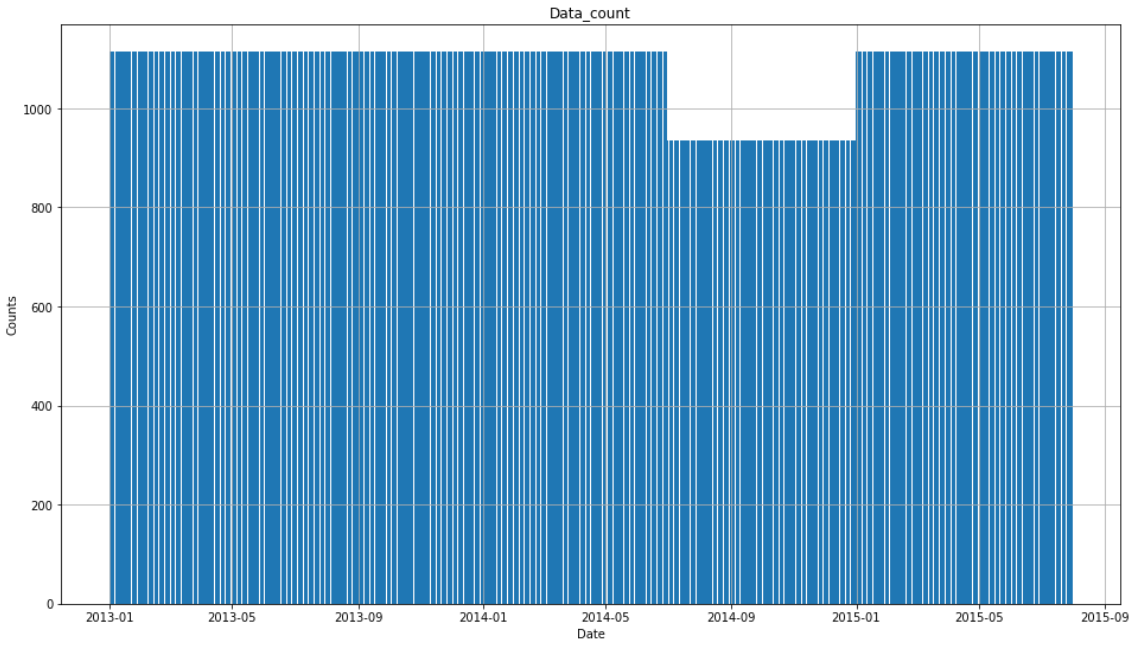
y_i 是指 i 店铺 i 日的实际销售额，yhat_i 是指通过模型预测 i 店铺 i 日的销售额，公式表示所有店铺某段时间段内的实际销售额与预测销售额之差所占实现销售额百分比的均方根，RMSPE 值越接近 0，说明整体预测的值与实际值越接近，精准度越高，误差越少，则比赛得分排名越靠前。RMSPE 值也将作为本项目基准模型训练的评估标准。

二　分析

2.1 数据的探索

本项目使用 Kaggle 上的提供数据集，包括训练数据集 train.csv、验证数据集 test.csv 和额外信息数据集 store.csv。

train.csv 数据集：



train.csv 数据信息图-1

通过查看该数据集信息得知，train.csv 数据集一共有 1017209 条数据，数据日期自 2013-01-01 至 2015-07-31。所有字段没有数据缺失，Date 日期字段为字符串。数据集中包含 Store 编号为 1-1115 的店铺信息，根据 train.csv 数据信息图-1 所示，其中 2014 年下半年的时间段里有部分店铺没有相关信息，猜想可能是部分店铺的装修导致的。白色竖杠部分表示当天没有数据，表明店铺的休息日应该是一致的。

test.csv 数据集：

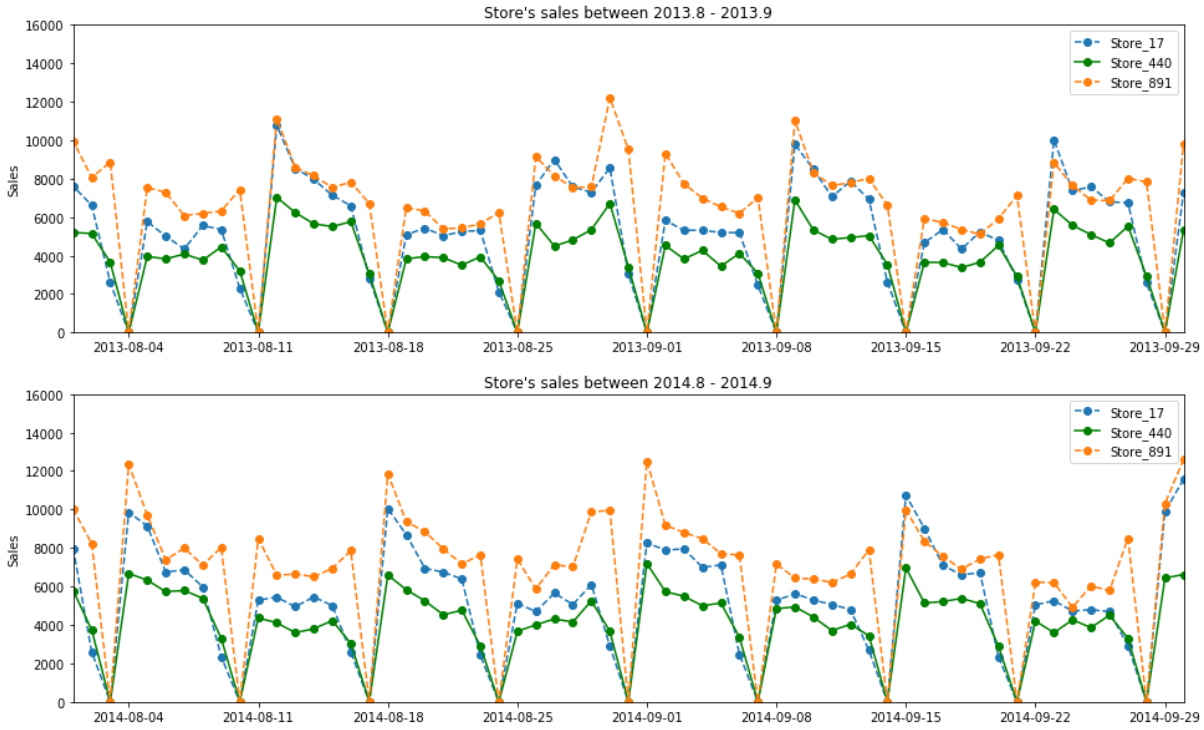
通过查看数据集信息得知，test.csv 数据集一共有 41088 条数据，Open 字段有 11 条数据缺失，没有 train.csv 数据集中的 Customers 字段。数据日期自 2015-08-01 至 2015-09-17，日期与 train 数据集的日期连续，这也是本次项目需要预测数据的时间段（48 天）。

store.csv 数据集：

通过查看数据集信息得知，store.csv 数据集一共有 1115 条数据，1115 条数据与 Store 店铺编号 1-1115 相符。StoreType、Assortment、PromInterval 为字符串字段，有多个字段数据缺失或为空值，后续需要进行填充和分隔。考虑将 store.csv 数据集的

信息拼接到 train 及 test 数据集中，为模型训练提供更多数据信息。

2.2 探索性可视化



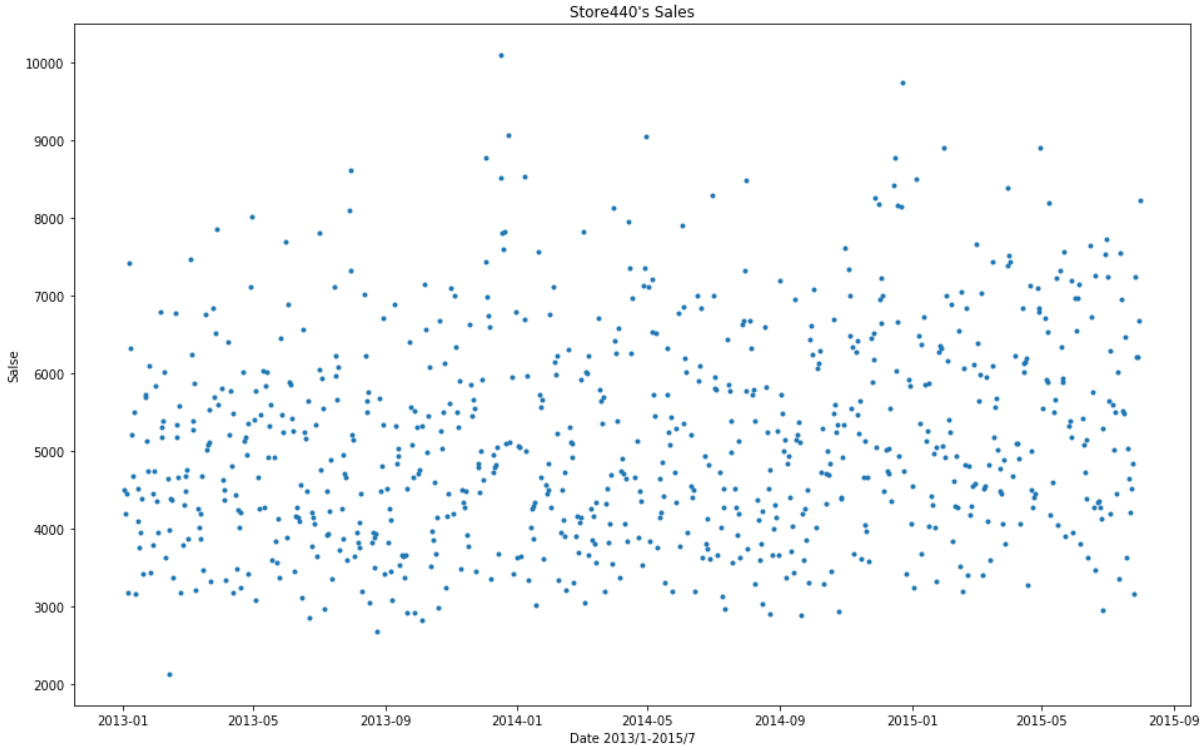
抽取店铺销售信息图

抽取 Store17、440、891 在需要预测的月份的往年销售数据绘制图表，Store440 的销售额相对于其余两家要低。Store17、891 两家的销售额较为接近，通过 Store.csv 信息查询这三家店铺的信息得知，Store17、891 的 StoreType 都属于相同类别 a 类，而 Store440 则属于 d 类，与图形体现的信息相符。

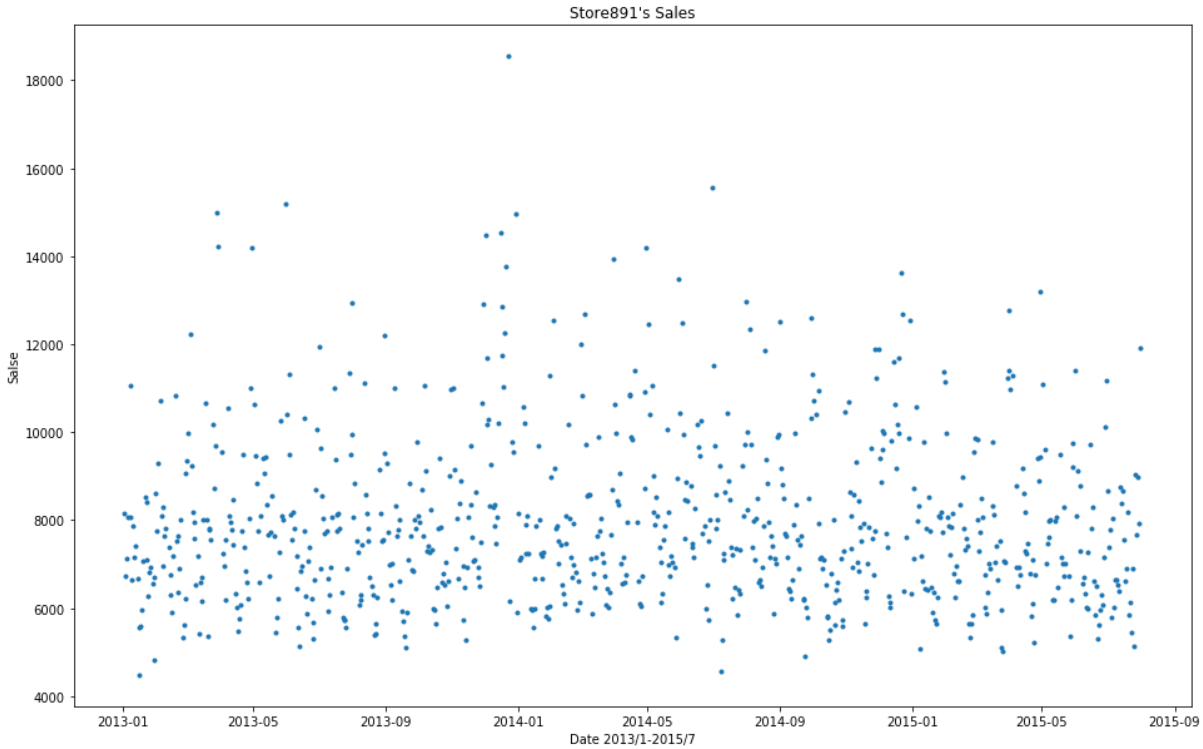
Store	StoreType	Assortment	CompetitionDistance	CompetitionOpenSinceMonth	CompetitionOpenSinceYear	Promo2	Promo2SinceWeek	Promo2SinceYear
17	a	a	50.0	12.0	2005.0	1	26.0	2010.0
440	d	a	3900.0	4.0	2005.0	1	45.0	2009.0
891	a	c	350.0	NaN	NaN	1	31.0	2013.0

抽取店铺的 store 信息表

根据抽取 Store440 的销量绘制散点图所示，该店铺的日销售额大部分集中于 3000-6000 的区间。



店铺 440 销售信息图



店铺 891 销售信息图

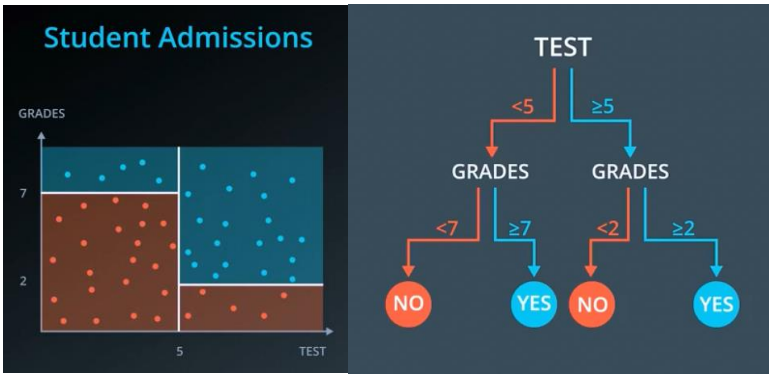
而 Store891 图形展现的销售额区间集中在 6000-8000，整体上高于 Store440。两家店铺在年中和年末都会出现销售的高峰，最高峰都出现在 2014 年年末，推断应该与年末的店铺促销有关。项目最后会将 Store891 作为检验对象进行预测分析。

2.3 算法和技术

2.3.1 决策树

决策树是一类常见的机器学习方法，基于训练数据集的特征，通过一系列的问题来推断样本的类标。它代表对象属性与对象值之间的一种映射关系。树中每个节点表示某个对象，而每个分叉路径则代表的某个可能的属性值，而每个叶结点则对应从根节点到该叶节点所经历的路径所表示的对象的值。在实际应用中，可能会生成一颗深度很大且拥有众多节点的树，容易产生过拟合问题，一般会通过对树进行“剪枝”来限定树的最大深度。

决策树分类的纯度越高效果越好，纯度量化的方式主要有信息增益，基尼系数等。二叉决策树（每个父节点被划分为两个子节点）常用的三个不纯度衡量标准或划分指标分别是：熵、基尼系数以及误分类率。如果某一节点中所有的样本都属于统一类别，则其熵为 0，当样本以相同的比例分属于不同的类时，熵的值最大。基尼系数与熵类似，当所有类别是等比例分布时，基尼系数的值最大，CART 主要使用基尼系数作分属性的选择。

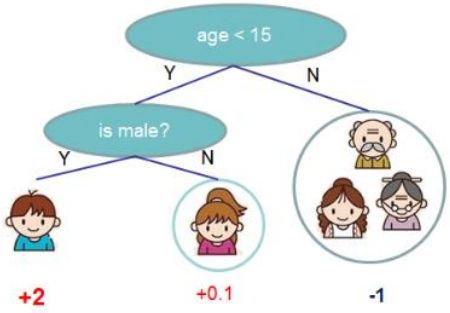


决策树示意图

2.3.2 分类与回归树-CART（Classification And Regression Tree）

分类与回归树(CART——Classification And Regression Tree))

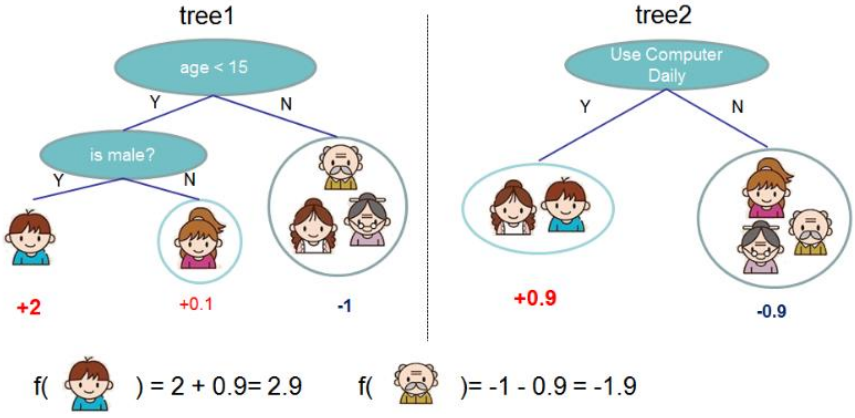
CART 分类回归树是一种典型的二叉决策树，可以做分类或者回归。数据对象的属性特征为离散型或连续型，并是不区别分类树与回归树的标准。作为分类决策树时，待预测样本落至某一叶子节点，则输出该叶子节点中所有样本所属类别最多的那一类（即叶子节点中的样本可能不是属于同一个类别，则多数为主）；作为回归决策树时，待预测样本落至某一叶子节点，则输出该叶子节点中所有样本的均值。



CART 回归示意图

2.3.3 树集成 - Tree Ensembles

通常情况下，单棵树由于过于简单而不够强大到可以支持在实践中使用的。实际使用的是所谓的 tree ensemble model（树集成模型），它将多棵树的预测加到一起。



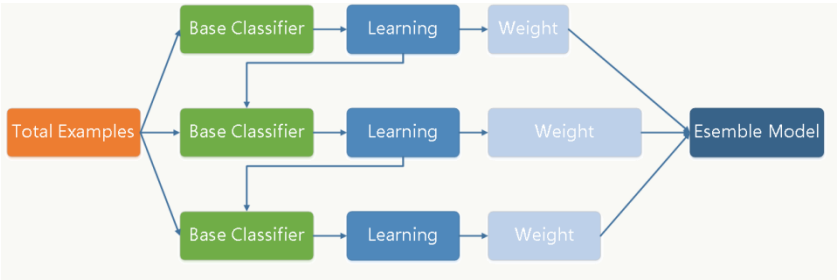
树集成示意图

2.3.4 技术

Gradient boosting(GB)机器学习中的学习算法的目标是为了优化或者说最小化 lo

ss Function， Gradient boosting 的思想是迭代生多个弱的模型，然后将每个弱模型的预测结果相加。

Gradient boosting Decision Tree(GBDT)，GB 算法中最典型的基学习器是决策树，尤其是 CART，正如名字的含义，GBDT 是 GB 和 DT 的结合。GBDT 是通过采用加法模型（即基函数的线性组合），以及不断减小训练过程产生的残差来达到将数据分类或者回归的算法。



GBDT 训练过程示意图

2.4 基准模型

本项目选择 XGBoost 作为基准模型，XGBoost 是 GBDT 算法的高效实现及优化。XGBoost 在竞赛和工业界都得到非常广泛的应用[3]。项目将以进入本竞赛排行榜（Private Leaderboard）的前 10%作为基准阈值。本竞赛一共有 3303 名参赛选手，前 10%即最终的模型评价标准 RMSPE 分值需低于第 330 名提交的分值 0.11773。进入前 10%基本上能够体现出模型的性能处于中上水平。

三　方法

3.1 数据预处理

3.1.1　统一字段类型及填充

首先将加载数据集 train.csv、test.csv、store.csv，由于 train 和 test 中的 StateHoliday 字段包含字符串和数字（0、a、b、c），所以先统一对该字段作字符串处理。

将 Date 字段的字符串类型改为日期类型，方便手续操作。

考虑到 store 表的空缺值比较多，所以先对 store 表中的空值进行填充，填充方式可以选择平均值、中位数或 0 等。在这里首先选择中位数对 CompetitionDistance 、CompetitionOpenSinceMonth、CompetitionOpenSinceYear 这个三个竞争对手的信息字段进行填充 。Promo2SinceWeek、Promo2SinceYear、PromoInterva 这三个字段的空缺值是 Promo2 字段为 0 没有做促销所致，所以可以直接填充 0 处理。将 CompetitionOpenSinceMonth、CompetitionOpenSinceYear、Promo2SinceWeek、Promo2SinceYear 四个字段原来的浮点型统一改为整型。

test 表的 Open 字段有 11 条为空值，通过分析空值所对应的日期，统计该日期值其他店铺的开业情况，99%的店铺在该日期段中都是开业的，所以将空值都填充为开业值 1。

将 stoer 表分别合并到 train 和 test 表上，生成新的 train 和 test 表。

3.1.2 特征编码

首先将生成新的 train 和 test 表对日期 Date 字段拆分为年 Year、月 Month、日 Day 为三个单独的字段。添加 Day_of_year 字段代表是一年当中的第 n 天。

将 StateHoliday、StoreType、Assortment、PromInterval 四个字符串字段进行 one-hot 热编码处理。

3.1.3 训练集、验证集划分

筛选 Open、Sales 字段大于 0 的数据，由于本项目数据集属于日期时间序列数据集，所以按日期时间段划分训练集和验证集，更有利于提高模型的性能。以 2015-6-1 为节点划分训练集和验证集 ,并移除所有字符串字段特征及 Date 和 Customers 特征。

3.2 实现

3.2.1 训练模型

编写评估指标 RMSPE 函数，作为模型的评估标准。加载训练集和验证集数据，初始化模型参数，训练模型。

3.2.2 预测

预测验证集、测试集数据。检验预测数据的 PMSPE 值，初始模型的 PMSPE 分值如下：

Val	Test_Private	Test_Public
0.15407	0.15490	0.15545

3.3 改进

由于开始只是对数据特征进行简单的处理，模型对数据的特性理解不够充分。需要尝试对数据特征进行更多优化及处理，提高模型性能。

3.3.1 优化特征

由于本项目数据为日期序列，对日期的敏感度较高，数据的变化规律会在时间段上有所体现。所以，在原有的特征基础上，增加更多能够体现时间特性的字段，包括：

Week_of_year（一年当中的第 n 周)、Quarter（季度)、Month_start（是否属于自然月的第 1 天)、Month_end(是否属于自然月的最后 1 天)、Quarter_start(是否属于某季度的第 1 天)、Quarter_end（是否属于某季度的最后 1 天)。

取消原有对 StateHoliday、StoreType、Assortment、PromInterval 四个字符串字段进行 one-hot 热编码处理，将 StateHoliday、StoreType、Assortmen 的字符串序列替换为数字序列，替换规则'0'/'a'/'b'/'c'/'d'->0/1/2/3/4。将 PromInterval 字段的内容通过计算匹配转换为 Is_promo2(当天是否进行 promo2 活动)。将对 CompetitionDistance 、CompetitionOpenSinceMonth、CompetitionOpenSinceYear 的中位数填充改为使用 0 填充。经过处理后，再次对模型进行训练，验证集的预测分值降低

至 0.14548。

3.3.2 新特征及目标变量缩放

根据 Promo2SinceYear、Promo2SinceWeek 字段以月为单位计算出 promo2 活动由开始到当日已经持续的时间，并记录在新的 Promo2_open 字段。

以月为单位计算出竞争对手开业时长，记录在新的 Competition_open 字段。结合 CompetitionDistance 和 Competition_open 的数据换算结果，建立 Competition_feature 字段，换算公式为: Sqrt(Max (ComDis) - ComDis) * ComOpen。

由于项目的预测目标变量是销售额，其数值区间都比较大，与数据的特征变量数值区间差异较大。所以在训练模型前，对销售额进行对数缩小处理，再次对模型进行训练，验证集的预测分值降低至 0.13115。

3.3.3 重新划分训练集

根据项目 test 数据集的日期区间为 48 天，将验证集的划分节点从 2015-6-1 改为 2016-6-13，使验证集与测试集的预测区间相对一致。同时使训练集数据从 785727 条增加至 797340 条，最终模型 val 分值降低至 0.11989，与基准阈值 0.11773 比较接近。

3.3.4 偏差校正

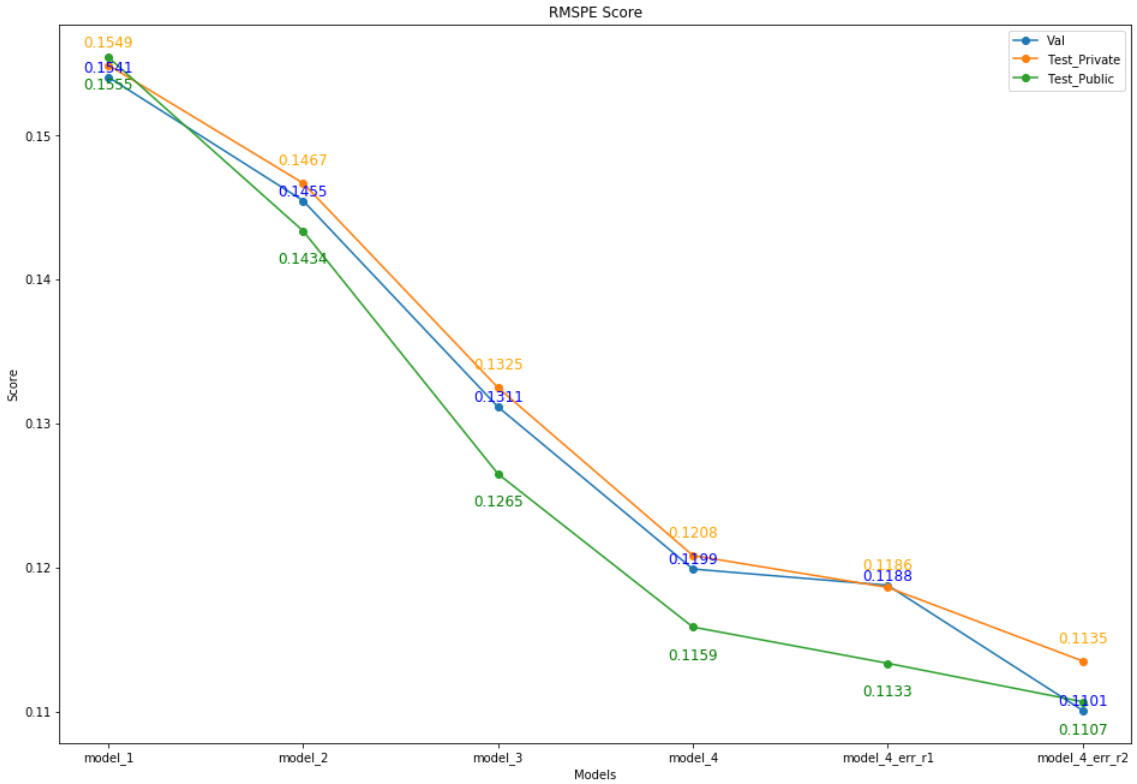
由于经过特征处理后的模型已经非常接近基准模型的检验标准，可以将提升模型的方式从特征转移到预测结果。对验证集的预测结果进行偏差校正，对预测结果进行±0.001 乘积校正，对比校正数据中各自的 PMSPE 值，得出最小的 PMSPE 值对应的校正系数。偏差校正方式分为两种，一种是整体校正、也一种是个体校正，一般个体校正比整体校正的结果更加细致、提升更多。分别对验证集进行整体校正以及根据每个 Store 编号对其预测结果进行个体校正，验证集预测值经过整体校正和个体校正后的 PMSPE 值分别是 0.11876 和 0.11007，其中通过个体校正后的 PMSPE 值已低于基准阈值 0.11773，性能提升显著。

四　结果

4.1 模型的评价与验证

将各阶段测试集的预测结果提交到 Kaggle 得出成绩测，对比初始模型和调整优化的各个阶段的验证集 RMSPE 值，模型经过不断特征优化性能提升处理，最终能够对达到预期的效果 ,超越基准模型阈值 ,成功进入竞赛的前 10% ,并以 Test_Private 0.11350 分值进入前 3%。

Model	Val	Test_Private	Test_Public
1	0.15407	0.15490	0.15545
2	0.14548	0.14670	0.14338
3	0.13115	0.13249	0.12650
4	0.11989	0.12083	0.11586
4_err_r1	0.11876	0.11861	0.11334
4_err_r2	0.11007	0.11350	0.11067



PEMSP 分值走势示意图

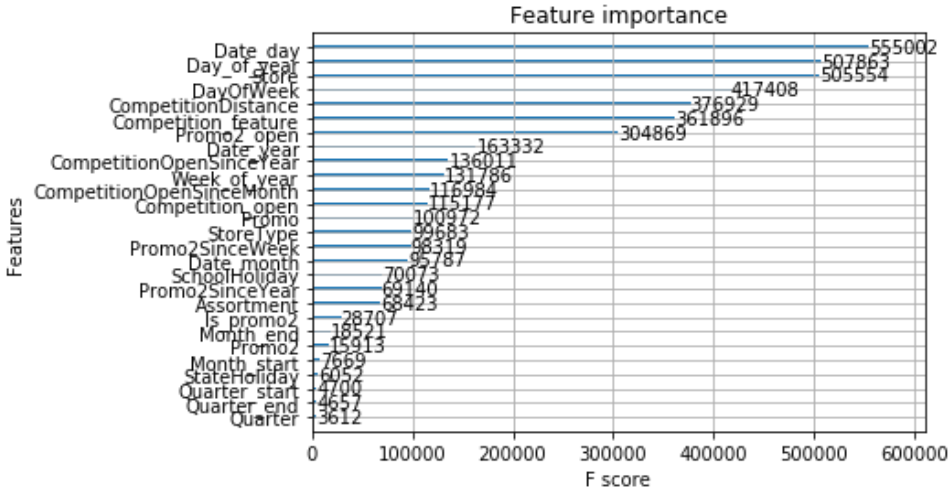
从图表上看模型对 Test_Public（test 数据的 31%）的销售数据预测最为精准。

五　结论

5.1 总结与思考

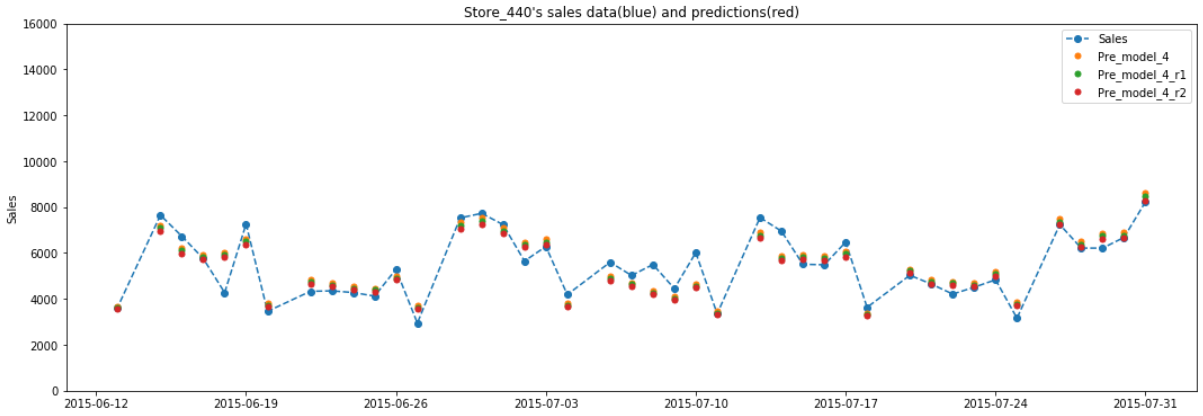
特征处理对模型的基础构建是十分重要，对特征的每项处理都会有可能影响到模型的性能。需要理解特征的隐藏含义，从项目给定的特征中挖掘构建更多有利于模型提升的新特征，使模型能够更容易更充分地理解和分析数据，才能构建健壮模型基础，有助于模型后续的优化与提升。

由最终模型列出的重要特征示意图得知，前六个重要分别是 Date_day、Day_of_year、Store、DayOfWeek、CompetionDistance 和 Competion_feature，证明对日期特征的分解及优化操作，对模型非常重要，至于 CompetionDistance 和 Competion_feature 特征对模型影响的重要性，在分析建模时是没有预想到的，特别是 Competion_feature 这个自建的特征，有如此大的影响有点出乎意料。

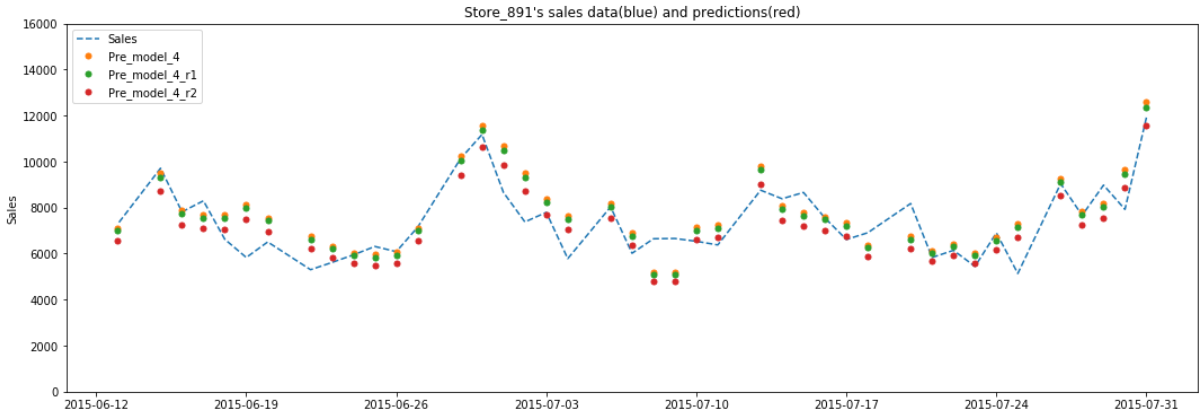


重要特征示意图

根据得出的最终模型，再次跟踪前面抽取的店铺 Store440、891，从验证集数据中观察模型预测的实际销售额的差异，以及最终模型与对结果的整体校正、个体校正之间对比变化。



Store440 销售预测示意图



Store891 销售预测示意图

从图中的预测点校正的方向显示，经过整体偏差校正后的预测值要比没有经过调整要低一些，抽样个体经过个体细致偏差校正后的预测值又比整体偏差校正的值要低，但是根据不同个体调整的幅度及方向是不同，Store891 的个体偏差校正的幅度要大于 Store440，个体校正更有利于对模型预测值整体偏高的有效校正，这也是个体偏差校正的预测效果优于整体校正的原因。

在项目中，需要对数据的序列类型要有清晰的认识，这对训练集的划分有非常关键的方向指引。例如本项目是日期序列，按日期时间段划分训练集，对模型的性能有着非常重要的作用，所以在划分训练集时需要根据数据的序列类型作为划分的依据。

5.2 改进

项目中对于空缺值的处理可以尝试让 XGBoost 模型自动处理，对比观察空值填充

的实际效果。另外，对模型特征的训练参数进行可以尝试更多参数组合，增大模型训练步值。

尝试构建多个基础模型，并对这些模型进行融合提升，对比差异。另外，可以尝试 XGBoost 模型以外的模型 ,如 Lightgbm 也是 boosting 类模型中非常常用及优秀的模型。在特征处理方面，可以尝试利用神经网络实现的 Entity Embedding，对结构化特征作实体嵌入，也有利于对模型的提升。相信在更多尝试的过程中，会对预测效果及理解数据特征方面有帮助或不一样的启发。

参考文献

[1] 周志华.机器学习. 清华大学出版社，2016.

[2] Sebastian Raschka.Python Machine Learning, 2017.

[3] hczheng. CSND 博客：xgboost 入门与实战（原理篇）. https://blog.csdn.net/sb19931201/article/details/52557382. Published:2016-09-16.

[4] zhihua_oba. CSDN 博客：决策树之 CART（分类回归树）详解. https://blog.csdn.net/zhihua_oba/article/details/72230427?utm_source=copy Published:2017-05-15.

[5] xgboost 文当. https://xgboost.readthedocs.io/en/latest/index.html

[6] pandas 文当. http://pandas.pydata.org/pandas-docs/stable/

[7] matplotlib 文当. https://matplotlib.org/contents.html

[8] Wes McKinney.Python for Data Analysis, 2013

[9] 陈天奇.XGBoost 与 Boosted Tree 我爱计算机 http://www.52cs.org/?p=429

[10] Modify's Programming. cnblogs 博客：https://www.cnblogs.com/ModifyRong/p/7744987.html