Forecast Rossmann Store Sales

贺新宇
2018/1/8

1 问题的定义

1.1 项目概述

Rossmann 是欧洲药店品牌，在欧洲的 7 个国家里运营着超过 3000 家药店。项目的任务是帮助 Rossmann 预测其药店未来一段时期的销量。对于零售行业而言，销售预测具有十分重要的意义。根据对未来销量的估计，可以提前指导、调整供应链的安排，例如提前进行库存调整、包装等耗材的采购与备货、甚至物流中心的建设等，通过这种提前调整与未来销售状况进行匹配，从而能够更有效地利用资源。

项目使用的算法是 XGBoost。XGBoost 是基于梯度提升框架进行优化的算法，能够实现并行计算、近似建树、有效处理稀疏数据以及优化内存使用等优点，使其成为处理机器学习问题最有力的算法之一。

项目使用的数据集来自 kaggle，train.csv 和 test.csv 分别是训练集和数据集，store.csv 是关于每个药店的其他详细信息。

1.2 问题陈述

项目要求预测位于德国境内 1115 家 Rossmann 药店未来 6 周的销量。我们知道药店的销量受多种因素影响，包括促销情况、竞争情况、学校和国家的假期、季节效应和位置等。如果能够根据这些现有数据估计其未来的销售规模，那么就能据此提前调整药品供应和员工安排，更有效率地利用现有资源应对未来的销售情况。

具体做法就是建立一个处理回归问题的预测模型，通过带有各种经过处理后的特征的数据来训练模型，用这个模型去预测未来销量这个未知量。

1.3 评价指标

根据评估标准，可以将模型预测的结果与实际结果进行比较、计算，以评估模型的性能。对于回归问题，评估指标包括 MSE、RMSE、R2 等，为便于与 kaggle 上的方案比较，项目使用 RMSPE 作为评估标准。

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

其中y_i和$\hat{y}_i$分别表示销量的真实值和模型预测值。

2 分析

2.1 数据的探索

数据集包括训练集、测试集以及药店补充信息，将补充信息合并到数据集中，然后对数据进行宏观的探索：

训练集共有 1017209 条记录，18 个属性，测试集共有 41088 条记录，17 个属性。包括待预测的目标属性'Sales'，与之有关的其他特征、药店的信息等。

依次对各个属性进行观察，可以发现数据集的一些特点：

一是数据类型的问题：需要对一些属性进行格式转换、对数转换和独热编码。

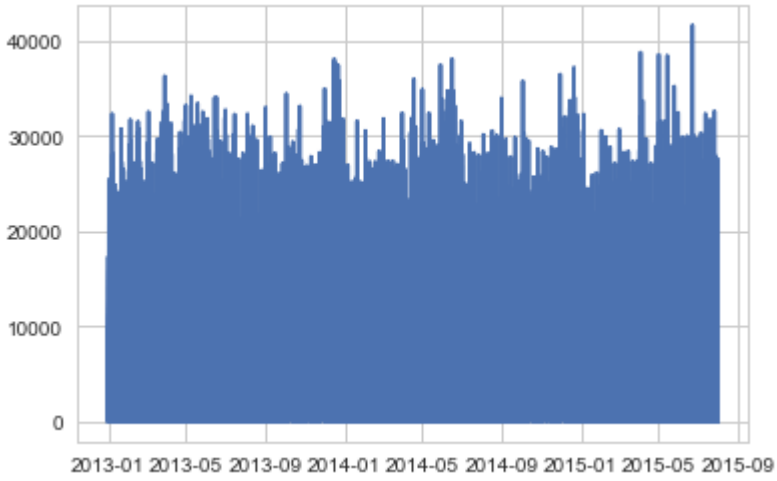
二是缺失值的问题：store 数据集和 test 数据集均有缺失值的情况。

三是训练集和测试集特征不一致，测试集相比训练集少了一个'Customers'特征。

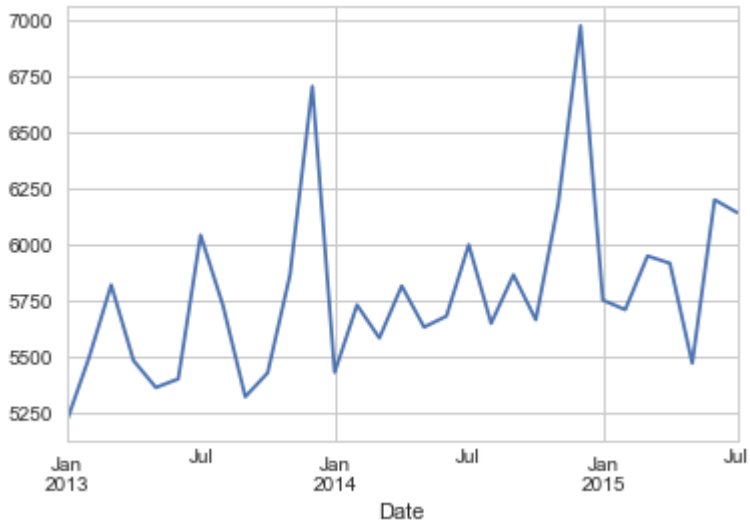
四是特征选择的问题，需要对特征进行增删、处理，筛选出对预测最有用的特征。

2.2 数据可视化和预处理

（1）首先观察'Date'变量对于'Sales'的影响：

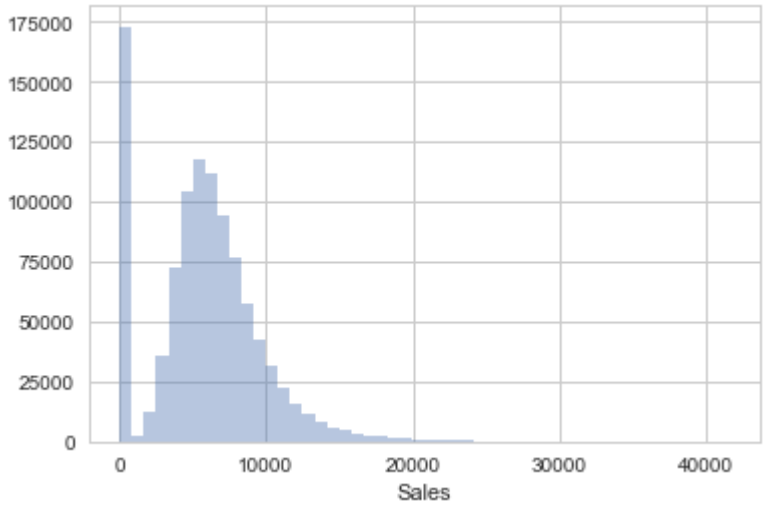


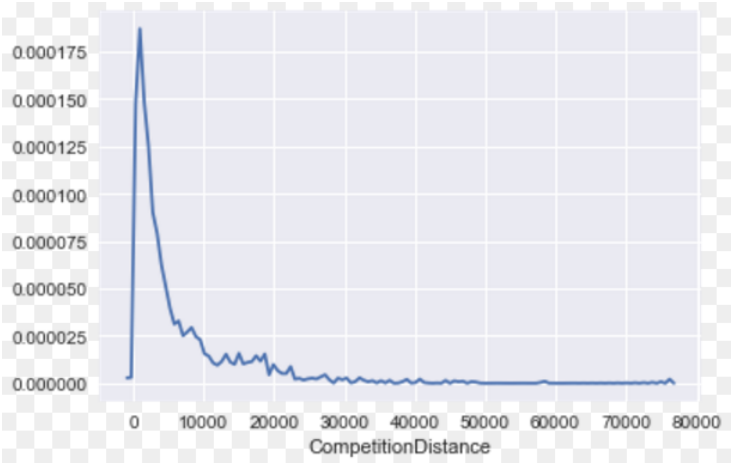
'Date'变量的原始格式不易观察，将其按月平滑后得到如下结果，可以看到销量随时间的变化有比较明显的规律性，因此时间变量对预测而言将是十分重要的变量。



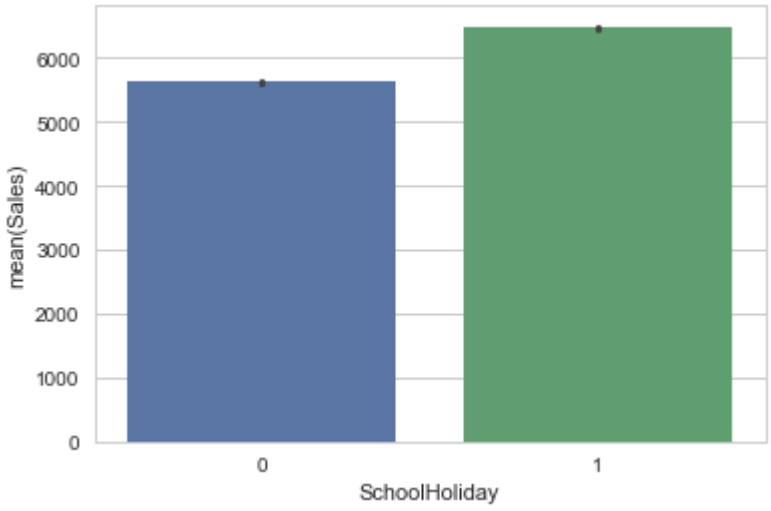
（2）观察'Sales'和'CompetitionDistance'的分布，可知需要对其进行对

数转换。





（6）观察'SchoolHoliday'变量与销量的关系，'SchoolHoliday'对销量具有一定影响，可以作为初始特征。对其他类似变量'Promo'、'Promo2'、'StoreType'、'Assortment'等变量的观察与此类似，不再重复。



2.3 算法和技术

销售预测是一个传统机器学习的回归问题，理论上可以用任何监督学习的经典算法。但这里根据数据的规模和特点、运行速度的要求、精度的要求，选择XGBoost算法进行建模，并使用GridSearchCV方法进行调参。

原因在于:XGBoost基于BoostedTree优化而来,BoostedTree精确度较高、对输入数据的要求不敏感，是数据挖掘和机器学习领域的经典算法，而XGBoost是大规模并行BoostedTree的工具,它进一步提升了BoostedTree的性能和运行速度，因此十分适合此项目任务。

XGBoost 较多的参数使得参数设置比较繁杂，这里用网格搜索法重点调整'learning_rate','min_child_weight','max_depth'三个参数。

因为数据集包含1000家以上的药店，因此采取的思路是每个店铺的数据分别训练、调参，不同的药店对应不同的参数和模型，最终通过保存的id再将预测数据整合在一起。

2.4 基准模型

Kaggle 竞赛中，Gert 基于 XGBoost 模型，遵照 recent data、temporal information、current trends 等三个原则生成特征，以0.1的误差结果获得第一名。OmarElGabry 则分享了其构建模型的过程和思路,项目将其作为基准模型,并将 kaggle 排名 50%以内作为基准阈值。

3 方法

3.1 预处理

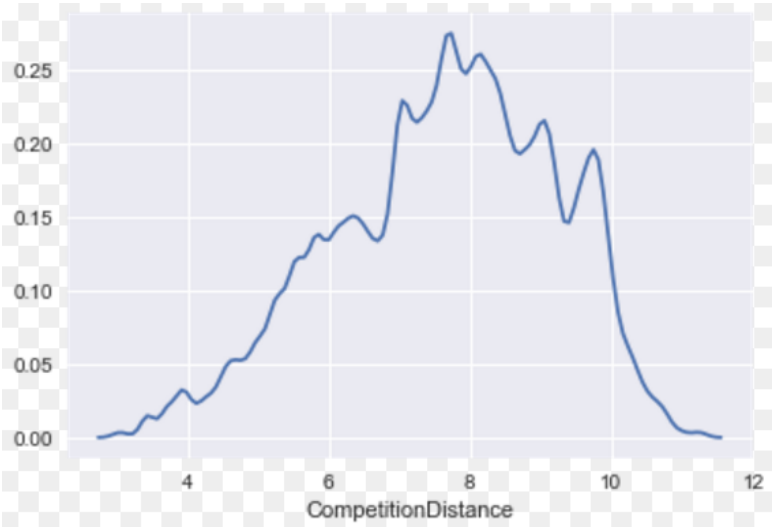
（1）时间序列'Date'是无法作为特征的，而从前面数据可视化的过程中得知，时间相关的变量将会极大影响销量的变化，因此从'Date'变量中提取出'Year'、'MonthofYear'、'WeekofYear'、'DayofMonth'等变量作为新增独立特征加入数据集。

（2）由于'Open'变量的缺失值对预测影响较大，涉及单独取出关门店门的销量需预测为0,因此需对其缺失值进行填充。而当药店关门时其销量应该为0，因此可以考虑将关门的记录暂时删除，等到提交时直接填充为0，与预测好的未关门店铺再一起提交。

（3）数据探索时观察到'StateHoliday'变量的取值:array(['0', 'a', 'b', 'c', 0L]，需对'0'和0L的形式进行统一。

（4）'StateHoliday'、'StoreType'、'Assortment'等变量的取值均为字符串格式，需要进行数值替换或独热编码。

（5）由前面的数据可视化已知，需对'CompetitionDistance'和'Sales'进行对数转换。转换后的分布如图:



（6）将'CompetitionOpenSinceMonth'和'CompetitionOpenSinceYear'变量结合形成新的变量'CompetitionOpenMonth'，'Promo2SinceWeek'和'Promo2SinceYear'变量结合形成新变量'Promo2sinceMonth'。

3.2 执行过程

基于 XGBoost 构建一个预测模型。项目的一个不同之处在于，采用的思路为一个药店对应一次调参。

具体流程为：将模型构建、网格搜索调参等过程装入一个函数，每次提取一个药店的数据进行训练，用网格搜索法进行调参，得到与这部分数据集匹配的一个模型。一个药店对应一组不同的参数，最后通过一个 for 循环将最终预测结果整合到一起。

问题在于如何先将数据打散，然后再按原来的顺序还原？这里使用数据集中的'Id'列，先保存关门的 records 的 Id，然后将未关门的 records 的 Id，以对应的 Store 为关键字，Id 为值的方式以字典的形式储存，最后合并时，将关门的 records 对照 Id 填入预测值 0（关门的药店无销量），将未关门的 records 按照 Store 的不同分别通过模型得到预测结果，然后提取出对应 Store 的 Id 依次填入。

3.3 完善过程

最开始考虑使用留出法分离验证集，在药店 1 的样本上手动调参'max_depth'找到最佳模型，然后应用于所有数据集，在 kaggle 上得到结果：0.13678、0.12112。

第二次考虑使用交叉验证和网格搜索法自动调参并返回最佳参数，但只调了参数'max_depth'，然后利用保存 id 的方式，在每个药店的数据上分别训练、调参，每个药店对应一个不同的模型，在 kaggle 上得到结果：0.13651、0.11872。

第三次考虑加入 CompetitionOpenMonth 和 Promo2sinceMonth 辅助特征，得到结果：0.13568、0.11882。

第四次增加了更多调参的参数，包括：

'learning_rate':(0.1,0.3),'min_child_weight':(1,3,5),'max_depth':(4,5,6)。结果是：0.13398、0.11771

最后通过通过获取各个特征重要性的评分，删除 StateHoliday、

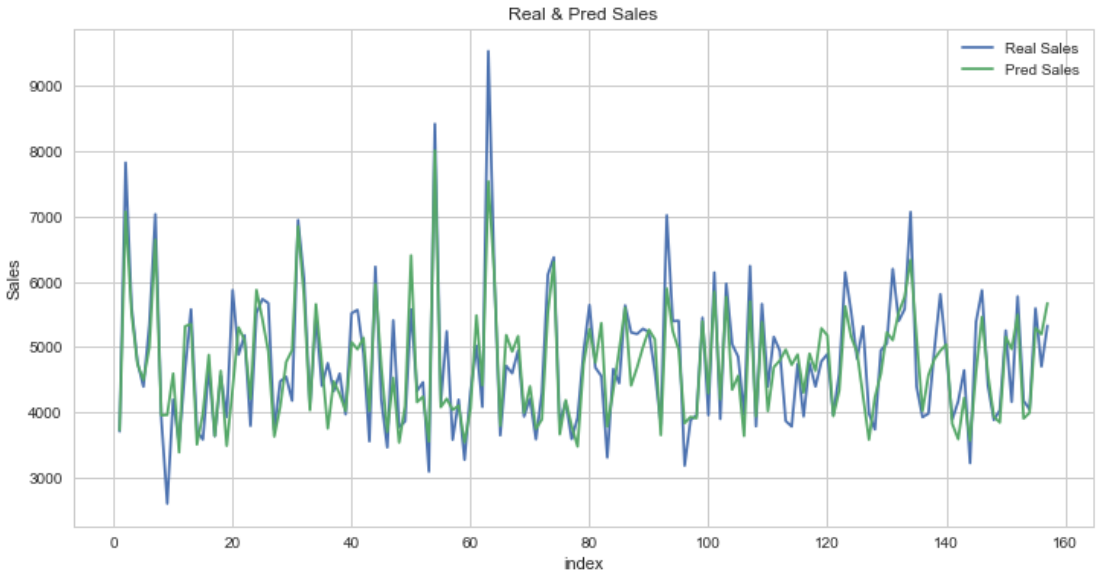
SchoolHoliday、StoreType、Assortment 、Promo2、Promo2sinceMonth 和 CompetitionDistance 等不重要的特征，得到结果：0.12790、0.11662。

4 结果

4.1 模型评价与验证

模型最终获得 0.12790（publicleaderboard）、0.11662（privateleaderboard）的结果，kaggle 排名在 50%以内，达到预期水平的基准线。在后期的历次试验中，privateleaderboard 的分数基本上变化不大，略有浮动，publicleaderboard 上的分数则明显的持续降低，因此后期的主要任务是提高模型的泛化能力。

真实值和预测值的对比（取部分样本）



可以看到模型基本捕捉了销量的真实值，但相对而言，真实值的波动幅度更大，模型预测值的波动较为缓和。

稳定性方面，将模型的输入数据进行一些微小的变换，在十几次试验过程中，结果大致稳定在可控范围内，保持了一定的稳健性，结果较为可信。

4.2 合理性分析

与基准模型相比，本项目在其基础上做了较多的完善，比如提取、新增了更多的特征，应用网格搜索法进行调参，依据特征重要性进行特征的筛选等。

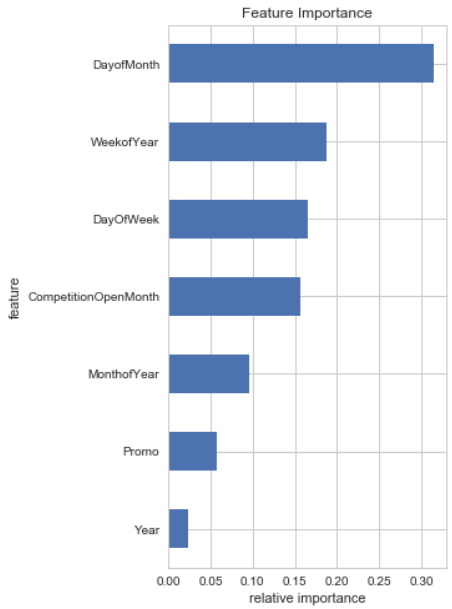
而相比第一名的模型，可以看到第一名使用更为巧妙的方法，首先使用多个模型集成取结果的调和平均值的方式，其次在特征选择过程中使用代表时间趋势的特征，并且还获取了这一段时间的天气特征数据，使得模型更加复杂。

本项目最终的结果远远好于基准模型的 0.16961，但相比第一名的 0.1 还有一定差距，结果是比较合理的。

5 结论

5.1 结果可视化

对各特征重要性的可视化:



由图展示了模型最重要的七个特征,可以发现时间趋势信息对预测销量而言十分重要,店铺类型等特征在这里反而没那么重要。依据此排序将其他相对不重要的特征删除,可以有效提升模型的泛化能力。

5.2 对项目的思考

项目的整体流程:

(1) 探索数据(包括可视化)、分析问题
(2) 数据预处理
(3) 建模(XGBoost)
(4) 调参
(5) 预测
(6) 分析结果、改善模型

在完成项目的过程中发现,并不是特征越多越好,太多无用的特征反而影响模型的准确度,在通过特征重要性删减模型之后,发现模型的泛化能力提升了。另外,对于机器学习的模型而言,特征与标签之间的关系是相关性而非因果性,所以不能简单的以因果关系的逻辑来筛选、增加特征,而应该结合领域知识和模型自身产生的特征重要性来选择特征。

最后,虽然模型取得了不错的结果,但可以看到本项目的模型在处理销售预测问题时不具备很高的通用性,因为本项目主要涉及的对特征的处理以及调参过程,不同问题都需要具体分析。

5.3 需要作出的改进

在算法方面,本项目未来还可尝试微软的 LightGBM 算法,以加快模型的运行速度。而在结果提升方面,通过对特征处理、网格搜索、特征重要性、一店一

模型几种方法或思路对模型不断地调试，已经使得模型的性能达到一个瓶颈。如果要进一步提升结果，则需思考其他特征处理的方法，比如第一名使用的多个模型结合取结果平均值、新增时间趋势特征、天气特征等。

另外，因为这次 kaggle 项目已关闭，所以可以无限次查看 privateleader 的分数，否则可能还需要人为切出一个测试集以测试泛化能力。

6 参考文献与资料

1. XGBoost https://github.com/dmlc/xgboost
2. Tianqi Chen,Carlos Guestrin.XGBoost: A Scalable Tree Boosting System,2016
3. 探索性数据分析：https://www.kaggle.com/thie1e/exploratory-analysis-rossmann
4. https://www.kaggle.com/omarelgabry/a-journey-through-rossmann-stores
5. 特征重要性：https://www.kaggle.com/cast42/xgboost-in-python-with-rmspe-v2/code
6. xgboost 调参：http://www.analyticsvidhya.com/blog/2016/03/complete-guide-parameter-tuning-xgboost-with-codes-python/
7. 结果可视化：https://www.kaggle.com/shearerp/interactive-sales-visualization