机器学习工程师毕业项目

2020年 7 月 31 日

王泓霏

1 定义

1.1 项目概述

这是一个 Kaggle 竞赛项目，目标是训练一个模型，来预测Rossmann零售店的销售额情况，是一个典型的回归问题。

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。商店不定时会举办短期的促销活动以及连续的促销活动以此来提高销售额。除此之外，商店的销售还受到许多因素的影响，包括促销、竞争、学校和国家假日、季节性和周期性。

我选择这个问题，一是因为它具有挑战性，可以提升自己的专业能力，二是我本人对于数据挖掘部分很感兴趣，渴望在这部分得到更多的锻炼，为接下来的专业深造做铺垫。

其中，药妆店的信息，在kaggle项目“Forecast Rossmann Store Sales”中提供。

本项目使用的模型是xgboost。

项目选择的数据集是 Kaggle 竞赛提供的数据，训练集包括1017209个样本，共9个特征，测试集包括41088个样本。对于，模型需要预测出是未来一段时间内，零售店的销售额情况。

1.2 问题陈述

Kaggle 竞赛提供的数据是Rossmann零售店过去实际的销售额情况，因此数据有大量的缺失值，这些都增加了问题难度。

这是数据挖掘中的回归问题，首先对于数据进行大致探索，关于此药妆店的开店情况，竞争对手的情况，进行可视化的展示。

目标是根据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。理想情况是，能够通过现有的数据集中药妆店的特征，较准确的地分析出未来的销售额的情况。

1.3 评价指标

y\_i是一个店在一天内的销售额，而y均是预测的平均值。

我选择与本次kaggle项目“Forecast Rossmann Store Sales”相同的的评价指标，即均方误差，即Root Mean Square Percentage Error (RMSPE). 由于RMSPE 对一组预测中的特大或特小误差反映非常敏感，因而它能够很好地反映出模型预测的精密度。对于需要预测的销售量，用 RMSPE 能够比较不错地表示模型的效果。

得出来的值越小，越接近真实值，模型的效果也就越好。最小值是0，也就是预测结果完全与真实值重合。

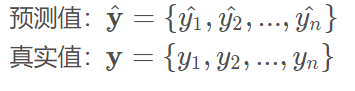


图1 真实值与预测值

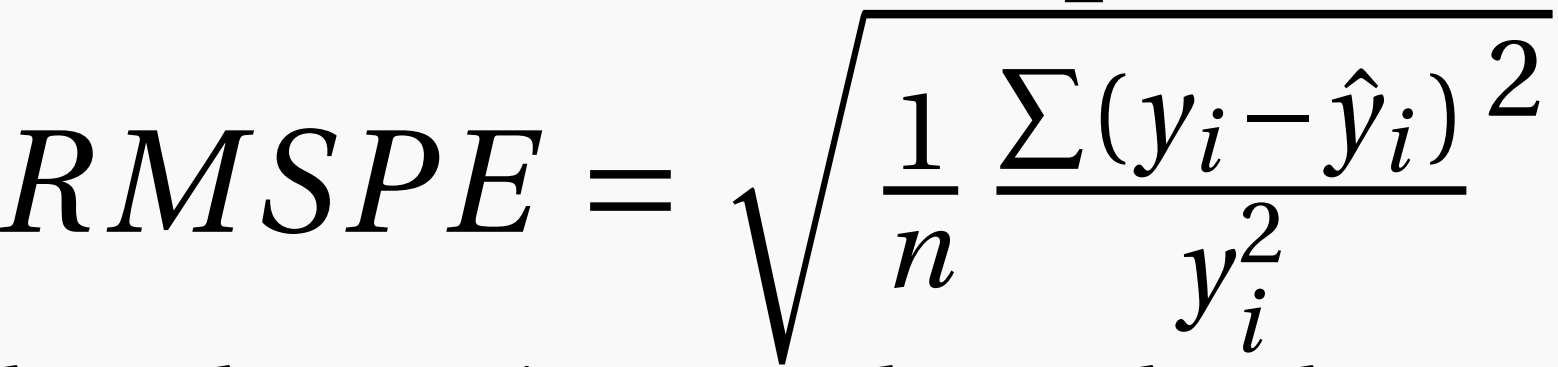


图2 RMSPE评价指标

2 分析

2.1 数据可视化

2.1.1 店铺销量

竞赛数据集有两个子集，一个训练数据集，一个测试数据集

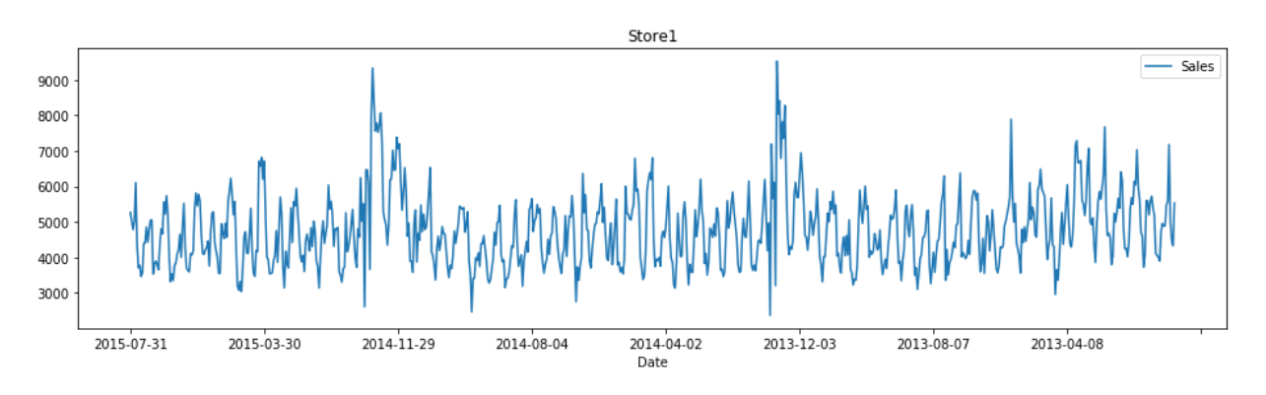
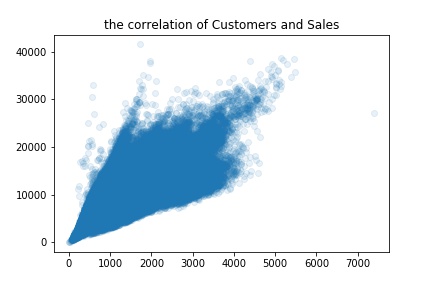
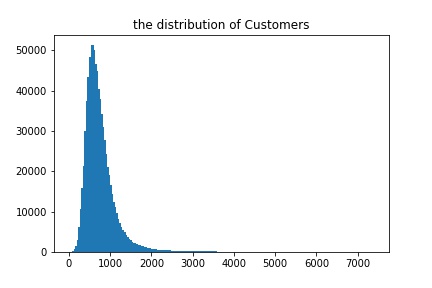


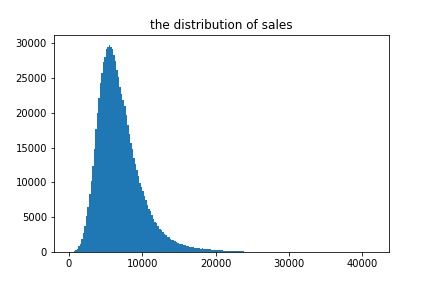
图 1 店铺销量随时间的变化

店铺的销售额是有周期性变化的，一年中11,12月份销量相对较高，可能是季节因素或者促销等原因 此外从2014年6-9月份的销量来看，6,7月份的销售趋势与8,9月份类似，而我们需要预测的6周在2015年8,9月份，因此我们可以把2015年6,7月份最近6周的1115家店的数据留出作为测试数据，用于模型的优化和验证

2.2.2 顾客和销量之间的相关性







由图可知，顾客数量和销量的之间有很强的相关性，顾客数量越多，销量越高，尤其在顾客数量更多的时候，相关性更加明显。

2.2 算法和技术

2.2.1 xgboost

XGB也是一种gradient boosting的方法，基于决策树的集成机器学习算法，XGBoost是由GBDT发展而来。XGBoost可以使用Regression Tree（CART）作为基学习器，也可以使用线性分类器作为基学习器。以CART作为基学习器时，其决策规则和决策树是一样的，但CART的每一个叶节点具有一个权重，也就是叶节点的得分或者说是叶节点的预测值。

和其他算法相比，XGBoost 算法的不同之处有以下几点：

应用范围广泛：该算法可以解决回归、分类、排序以及用户自定义的预测问题；

可移植性：该算法可以在 Windows、Linux 和 OS X 上流畅地运行；

语言：支持包括 C++、Python、R、Java、Scala 和 Julia 在内的几乎所有主流编程语言；

云集成：支持 AWS、Azure 和 Yarn 集群，也可以很好地配合 Flink、 Spark 等其他生态系统。

在此项目中将应用xgboost作为主要模型,来进行数据集的预测。

参数：

params 这是一个字典，里面包含着训练中的参数关键字和对应的值，形式是params = {‘booster’:’gbtree’,’eta’:0.1}

silent [default=0]:设置成1则没有运行信息输出，最好是设置为0.

nthread [default to maximum number of threads available if not set]：线程数

dtrain 训练的数据

evals 这是一个列表，用于对训练过程中进行评估列表中的元素。形式是evals = [(dtrain,’train’),(dval,’val’)]或者是evals = [(dtrain,’train’)],对于第一种情况，它使得我们可以在训练过程中观察验证集的效果。

obj,自定义目的函数

feval,自定义评估函数

early\_stopping\_rounds,早期停止次数 ，假设为100，验证集的误差迭代到一定程度在100次内不能再继续降低，就停止迭代。这要求evals 里至少有 一个元素，如果有多个，按最后一个去执行。返回的是最后的迭代次数（不是最好的）。如果early\_stopping\_rounds 存在，则模型会生成三个属性，bst.best\_score,bst.best\_iteration,和bst.best\_ntree\_limit

evals\_result 字典，存储在watchlist 中的元素的评估结果。

verbose\_eval (可以输入布尔型或数值型)，也要求evals 里至少有 一个元素。如果为True ,则对evals中元素的评估结果会输出在结果中；如果输入数字，假设为5，则每隔5个迭代输出一次。

xgb\_model ,在训练之前用于加载的xgb model。

scale\_pos\_weight [默认 1]

在各类别样本十分不平衡时，把这个参数设定为一个正值，可以使算法更快收敛。

max\_delta\_step[默认0]

这参数限制每棵树权重改变的最大步长。如果这个参数的值为0，那就意味着没有约束。如果它被赋予了某个正值，那么它会让这个算法更加保守。

通常，这个参数不需要设置。但是当各类别的样本十分不平衡时，它对逻辑回归是很有帮助的。

2.3 基准指标

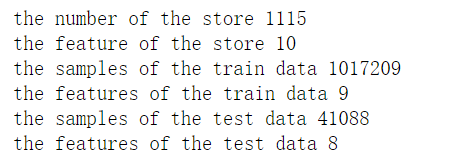
本次竞赛以RMSPE作为误差函数来进行评价。

考虑到训练所需要的大量硬件资源，以及需要的大量时间，受限于客观条件，项目设定的目标为 RMSPE小于0.11

3 具体方法

3.1 数据预处理

3.1.1 数据分析



从对数据集的分析可知，一共有1115个商店。训练集的样本数量超过100万个，是一个比较庞大的数据集。

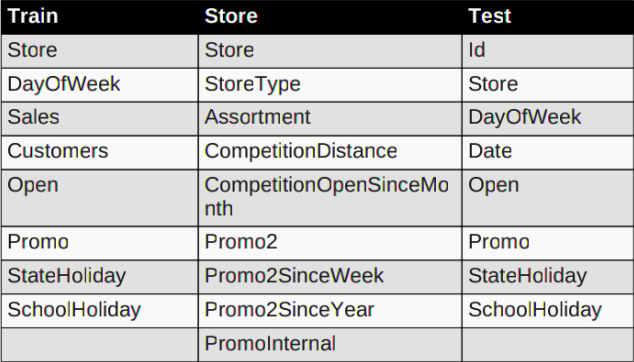


图 特征分布

由图可知有三个数据及其中的store是关于商店类型的，因此在后续数据处理时考虑将store分别和test和train进行合并。

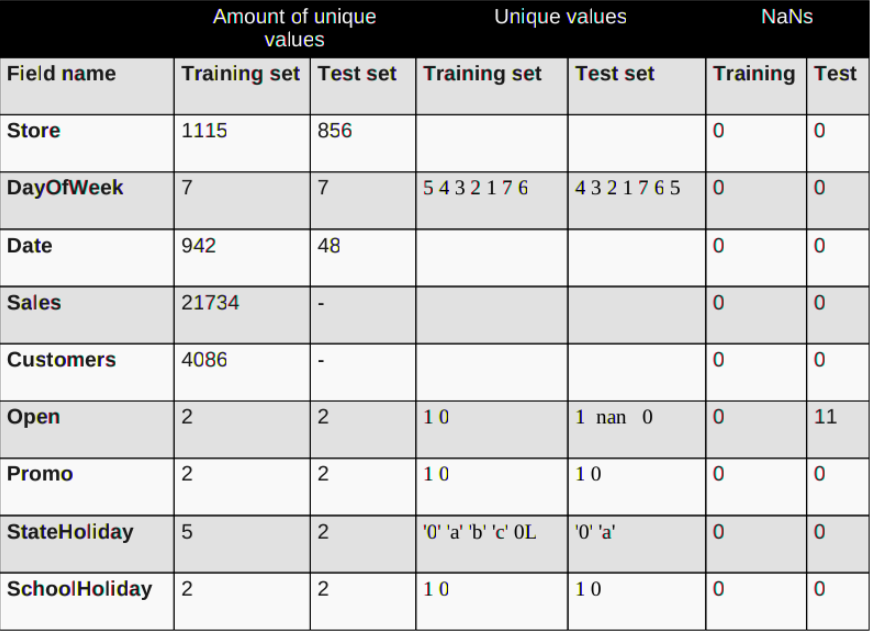


图 唯一值 缺失值

从表格中我们可以看到，在test集open中有11个缺失值，同时在train和test中，stateholiday并不一致，test中只有0和a。

3.1.2 数据集合并

本项目给了三个数据集，有store，test和train。其中store是关于每一个商店的信息的，因此将store分别和test和train进行合并。

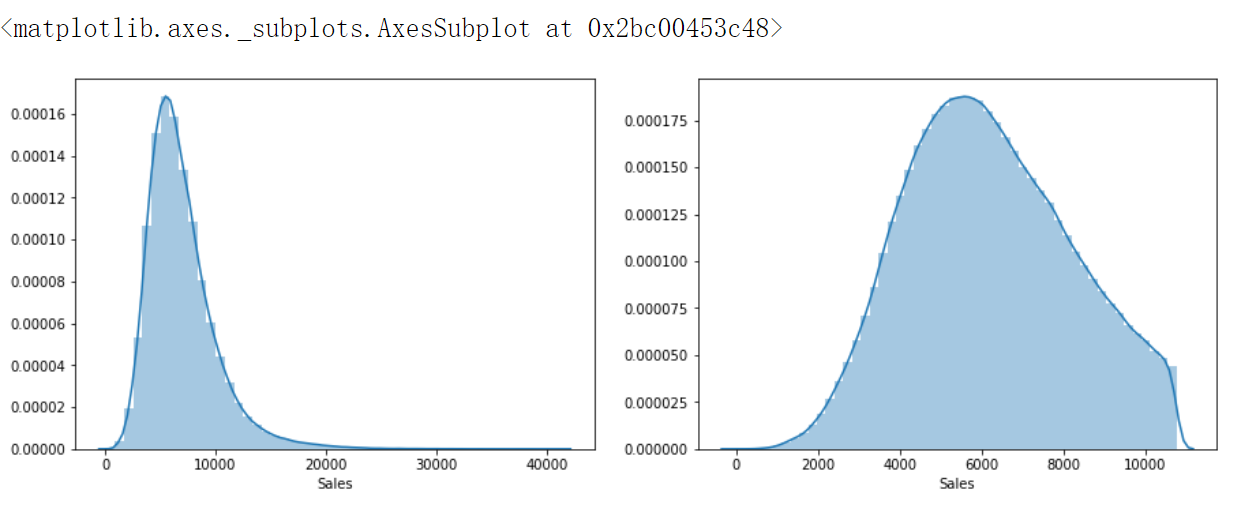
3.1.3 数据变换

图 长尾分布

这里很明显可以看出来数据的标签（Sales）是长尾分布，并不利于数据的预测，因为很多的模型都假设数据符合正态分布，所以我们需要对长尾分布的数据进行对数变换。

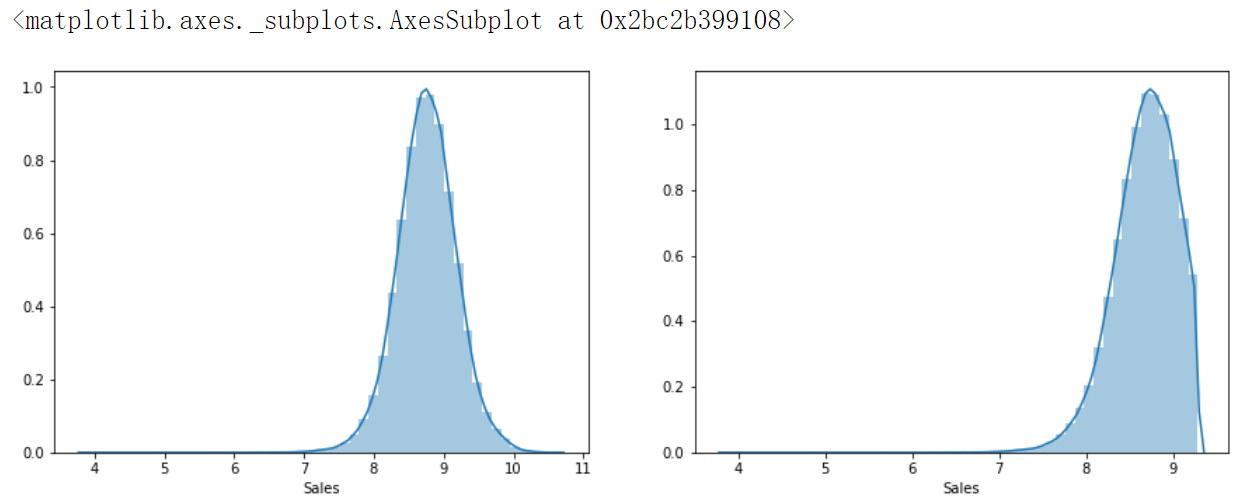


图 正态分布

对数变换后的数据更加符合正态分布，有利于之后训练模型。

3.1.3 缺失值处理

店铺竞争数据缺失，而且缺失的都是对应的。原因不明，而且数量也比较多，如果用中值或均值来填充，有失偏颇。暂且填0，解释意义就是刚开业

店铺促销信息的缺失是因为没有参加促销活动，所以我们以0填充

3.2 实现

#3.2章节可以总结一下你的算法实施过程。

3.2.1 构建特征

首先将数据集中不需要处理的特征，加入到选择的特征集中，有'Store', 'CompetitionDistance','Promo', 'Promo2', 'SchoolHoliday'。

然后对'StoreType', 'Assortment', 'StateHoliday'这几个分类变量进行0-4编码。

在处理时间序列，因为所给的时间序列是带年月日的完整日期，所以需要将年月日分开来形成三个单独的特征，并获取每天的星期以及在当年的周数，从而形成5个特征。'DayOfWeek', 'Month', 'Day', 'Year', 'WeekOfYear'

最后再对'CompetitionOpen', 'PromoOpen'这两个变量进行处理，将所给的日期转化成所持续的时间数据。

最终选择的特征是'Store', 'CompetitionDistance', 'Promo', 'Promo2', 'SchoolHoliday', 'StoreType', 'Assortment', 'StateHoliday', 'DayOfWeek', 'Month', 'Day', 'Year', 'WeekOfYear', 'CompetitionOpen', 'PromoOpen'

3.2.2 模型的训练

使用xgb模型进行训练，程序设置了 Early Stop，监视 val\_loss。当 val\_loss 连续 100 个 epoch 没有改进时停 止训练。

3.2.3 结果的生成

Validating RMSPE: 0.102849

3.3 改进

增加其他模型，如时间序列线性模型，随机森林。

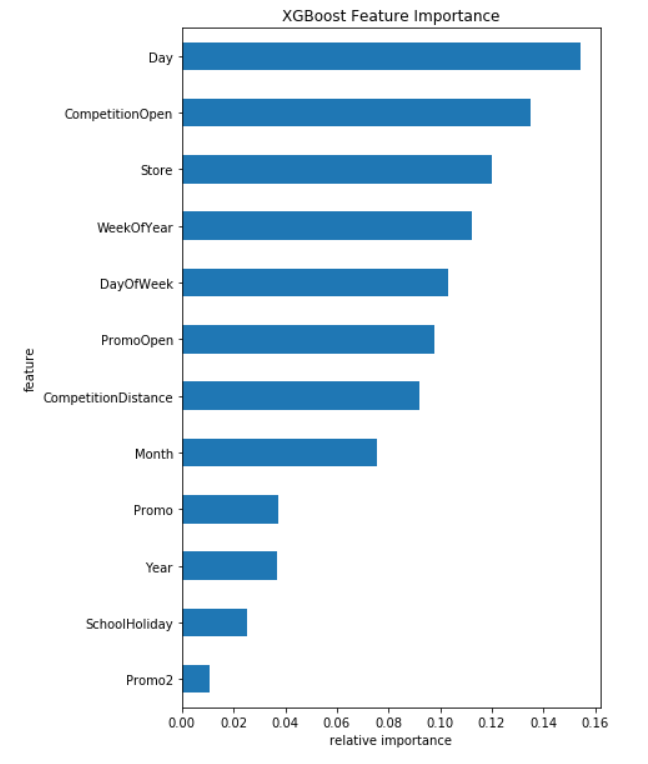
进行多模型融合，以达到比单个模型本身更好的效果。

4 结果

4.1 模型评价与验证

4.1.1 特征分析

图 模型特征重要性排序

从图上可以知道，在模型中，最重要的特征有天数，竞争对手开店时间，商店种类。由fscore可知分别占了0.154189，0.134849，0.120229 

4.2 结果分析

Validating RMSPE: 0.102849，达到预期。

5 结论

挑战：处理海量数据（10,17,210个样本，13个特征）

180个商店关闭了6个月，无法填补这些商店销量的空白。

对1115左右个商店进行预测，且商店门类还有很大不同。

The Rossmann Store Sales problem 此次的预测销售额竞赛事项，关于数据挖掘的任务，其中，我认为特征选择比模型选择更为重要，我花了大部分的时间在于挖掘数据之间的联系，以构建更好的特征。

参考文献

[1] Kaggle 官网

[2]https://xgboost.readthedocs.org/en/latest/

[3] https://www.kaggle.com/wiki/RandomForests