

# 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。

本次 kaggle 项目 “Forecast Rossmann Store Sales” 中，使用的评价指标为均方误差，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。

预测值:  $\hat{y} = \{\hat{y}_1, \hat{y}_2, \dots, \hat{y}_n\}$

真实值:  $y = \{y_1, y_2, \dots, y_n\}$

图 1 真实值与预测值

$$RMSE = \sqrt{\frac{1}{n} \sum (y_i - \hat{y}_i)^2}$$

图 2 RMSE 评价指标

## 2 分析

### 2.1 数据可视化

竞赛数据集有两个子集，一个训练数据集，一个测试数据集

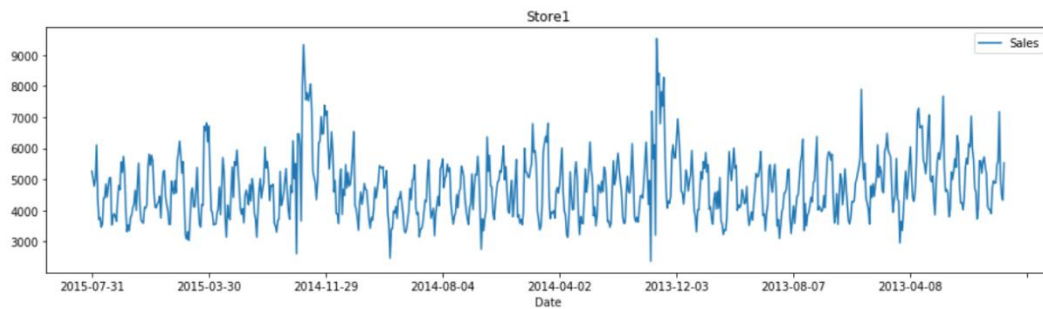


图 1 店铺销量随时间的变化

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 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。

## 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

Train	Store	Test
Store	Store	Id
DayOfWeek	StoreType	Store
Sales	Assortment	DayOfWeek
Customers	CompetitionDistance	Date
Open	CompetitionOpenSinceMonth	Open
Promo	Promo2	Promo
StateHoliday	Promo2SinceWeek	StateHoliday
SchoolHoliday	Promo2SinceYear	SchoolHoliday
	PromoInternal	

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。

Field name	Amount of unique values		Unique values		NaNs	
	Training set	Test set	Training set	Test set	Training	Test
Store	1115	856			0	0
DayOfWeek	7	7	5 4 3 2 1 7 6	4 3 2 1 7 6 5	0	0
Date	942	48			0	0
Sales	21734	-			0	0
Customers	4086	-			0	0
Open	2	2	1 0	1 nan 0	0	11
Promo	2	2	1 0	1 0	0	0
StateHoliday	5	2	'0' 'a' 'b' 'c' 0L	'0' 'a'	0	0
SchoolHoliday	2	2	1 0	1 0	0	0

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

<matplotlib.axes.\_subplots.AxesSubplot at 0x2bc00453c48>

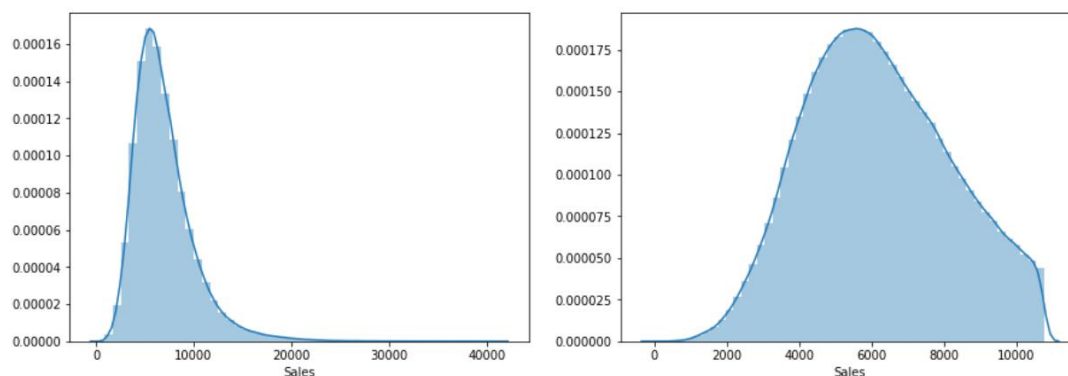


图 长尾分布

这里很明显可以看出数据的标签（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。

<matplotlib.axes.\_subplots.AxesSubplot at 0x2bc2b399108>

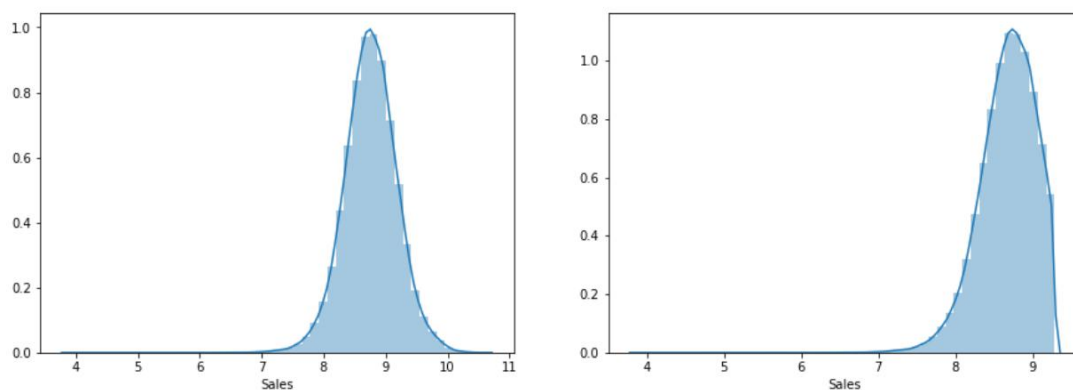


图 正态分布

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。

## 3.2 实现

### 3.2.1 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。

### 3.2.1 结果的生成

Validating RMSPE: 0.102849

## 3.3 改进

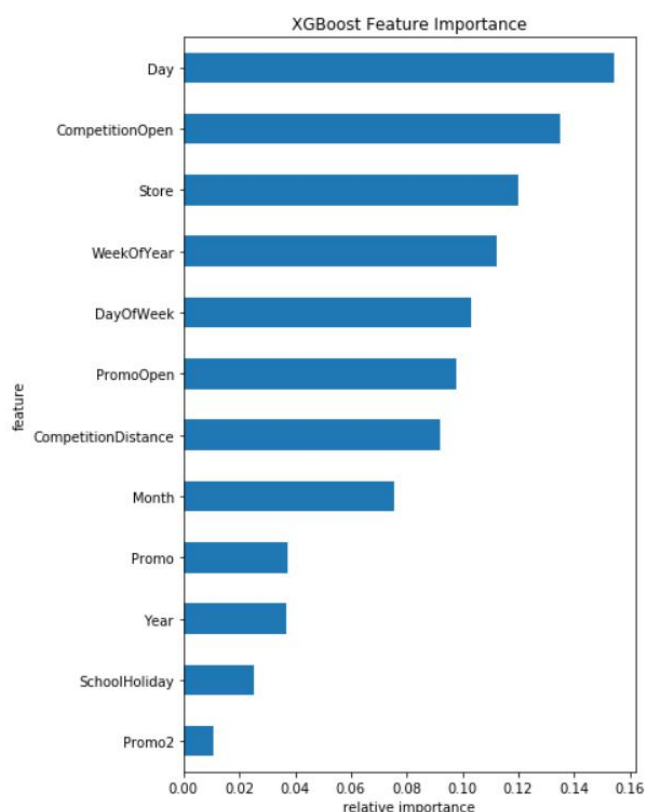
增加其他模型，如时间序列线性模型，随机森林。  
进行多模型融合，以达到比单个模型本身更好的效果。

## 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>