|  |
| --- |
| **机器学习** |
| **罗斯曼商店销售** |
| 主 研 人：刘一阳  参 研 人： |
| 审 核 人：  方 向：工业算法案例研究  版 本 号：A |
|  |

**版本更新**

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **更新描述** | **作者** |
| 2019/03/25 | A | 初稿 | 刘一阳 |

**选题表格**

|  |  |  |  |
| --- | --- | --- | --- |
| 时间 | 竞赛名 | 竞赛背景描述（50字以内） | 类型（分类/回归） |
| 2019/03/25 | 罗斯曼商店销售（Rossman Store Sales） | Rossmann 在7个欧洲国家经营着3000多家药店，目前，Rossmann 商店经理的任务是提前六周预测他们的日常销售，商店销售主要收到包括促销，竞争，学校等多个因素的影响 | 回归 |

目录

[1. 背景描述 4](#_Toc25575_WPSOffice_Level1)

[1.1 竞赛赛题描述 4](#_Toc5940_WPSOffice_Level2)

[1.2 评估指标描述 5](#_Toc12357_WPSOffice_Level2)

[2. 数据来源及描述性统计分析 5](#_Toc5940_WPSOffice_Level1)

[2.1 大赛数据来源 5](#_Toc22683_WPSOffice_Level2)

[2.2 数据的描述性统计 5](#_Toc16174_WPSOffice_Level2)

[2.2.1 数据基本情况描述： 5](#_Toc5940_WPSOffice_Level3)

[2.2.2 数据字段介绍： 5](#_Toc12357_WPSOffice_Level3)

[2.2.3 数据描述性统计 6](#_Toc22683_WPSOffice_Level3)

[3. 优秀算法思路 6](#_Toc12357_WPSOffice_Level1)

[3.1 方案一 6](#_Toc32293_WPSOffice_Level2)

[3.1.1 方案一数据预处理及特征工程部分方案 6](#_Toc16174_WPSOffice_Level3)

[3.1.2 方案一模型设计、建立部分方案 6](#_Toc32293_WPSOffice_Level3)

[3.1.3 方案一结果、排名等 6](#_Toc29922_WPSOffice_Level3)

[3.1.4 方案一算法流程图 6](#_Toc4787_WPSOffice_Level3)

[3.2 方案二 7](#_Toc29922_WPSOffice_Level2)

[… 8](#_Toc4787_WPSOffice_Level2)

[4. 算法比较 8](#_Toc22683_WPSOffice_Level1)

[表4-1 算法比较 8](#_Toc7253_WPSOffice_Level2)

[5. 总结与展望 8](#_Toc16174_WPSOffice_Level1)

[5.1 总结 8](#_Toc23198_WPSOffice_Level2)

[5.2 建模思路 8](#_Toc16619_WPSOffice_Level2)

1. 背景描述

Dirk Rossmann GmbH是德国第二大药店连锁店（仅次于dm-drogerie markt），在欧洲拥有超过3,790家商店。Rossmann由Dirk Rossmann于1972年创立。Rossmann家族拥有60％的股份，而香港的AS Watson集团拥有该公司40％的股份。公司总部位于汉诺威附近的德国城镇Burgwedel。

**1.1 竞赛赛题描述**

Rossmann在7个欧洲国家经营着3,000多家药店。目前，Rossmann商店管理者们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情况，结果的准确性可能会有很大差异。

<https://www.kaggle.com/c/rossmann-store-sales>

**1.2 评估指标描述**

竞赛中评估模型优劣的指标。

选手的提交的评估指标是均方根百分比误差（RMSPE），其计算公式为：

其中 代表一个商店的日销售额， 为预测值。如果某天某店的销售为0则可以在计算时被忽略。

1. 数据来源及描述性统计分析

**2.1 大赛数据来源**

描述大赛数据的来源

大赛数据提供方罗斯曼集团，为大赛提供1115家Rossmann商店的历史销售数据，从软硬件环境诸多方面提供大赛支撑。

这里是数据的超链接。

链接：[https://pan.baidu.com/s/1bDSB0Ht-I8xluAQFOBFpwA](https://pan.baidu.com/s/1bDSB0Ht-I8xluAQFOBFpwA%20)

提取码：p37w

**2.2 数据的描述性统计**

* + 1. **数据基本情况描述：**

您将获得1,115家Rossmann商店的历史销售数据。任务是预测测试集的“Sales”列。请注意，数据集中的某些商店暂时关闭以进行翻新。

* + 1. **数据字段介绍：**

train.csv - 训练集

test.csv – 测试集

sample\_submission.csv – 提交文件的样式

store.csv - 关于商店的补充信息

**表2-1 train.csv数据表字段介绍**

|  |  |  |  |
| --- | --- | --- | --- |
|  | 变量含义 | 变量类型 | 缺失率 |
| Store | 每家商店的id | 连续 | 0% |
| DayOfWeek | **日期** | **连续** | **0%** |
| Date | 日期 | 连续 | 0% |
| Sales | **每天的营业额** | **离散** | **0%** |
| Customers | 改天的顾客数量 | 离散 | 0% |
| Open | **是否开门** | **是否** | **0%** |
| Promo | 当天是否有促销 | 是否 | 0% |
| StateHoliday | **当天是否为特定的节假日通常情况下，除了少数例外，所有商店都会在州假期关闭。请注意，所有学校在公众假期和周末都关闭。a =公共假日，b =复活节假期，c =圣诞节，0 =无** |  | **0%** |
| SchoolHoliday | 显示商店、时间是否被学校的关门影响 | 是否 | 0% |

**表2-2 test.csv数据表字段介绍**

|  |  |  |  |
| --- | --- | --- | --- |
|  | 变量含义 | 变量类型 | 缺失率 |
| Id | 表示测试集的(Store, Date)的ID | 名称 | 0% |
| Store | **每家商店的id** | **连续** | **0%** |
| DayOfWeek | 日期 | 连续 | 0% |
| Date | **日期** | **连续** | **0%** |
| Open | 是否开门 | 是否 | 0% |
| Promo | **当天是否有促销** | **是否** | **0%** |
| StateHoliday | 当天是否为特定的节假日通常情况下，除了少数例外，所有商店都会在州假期关闭。请注意，所有学校在公众假期和周末都关闭。a =公共假日，b =复活节假期，c =圣诞节，0 =无 |  | 0% |
| SchoolHoliday | **显示商店、时间是否被学校的关门影响** | **是否** | **0%** |

**表2-3 sample\_submission.csv数据表字段介绍**

|  |  |  |  |
| --- | --- | --- | --- |
|  | 变量含义 | 变量类型 | 缺失率 |
| Id | 表示测试集的(Store, Date)的ID | 名称 | 0% |
| Sales | **每天的营业额** | **离散** | **0%** |

**表2-4 store.csv数据表字段介绍**

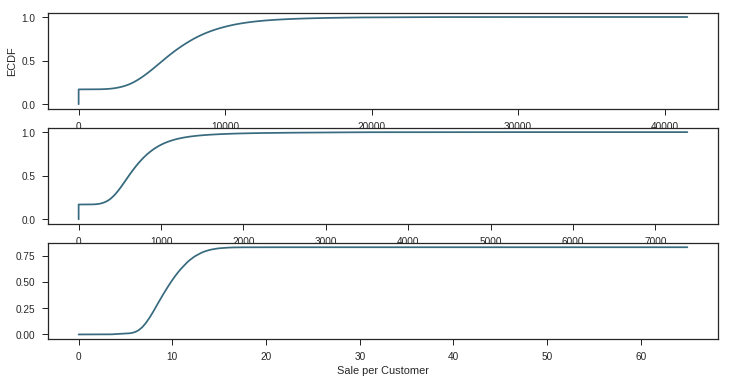
|  |  |  |  |
| --- | --- | --- | --- |
|  | 变量含义 | 变量类型 | 缺失率 |
| Store | 每家商店的id | 连续 | 0% |
| StoreType | **四种商店模型的（a, b, c, d）** | **离散** | **0%** |
| Assortment | 分类、混合物等级（a, b, c） | 离散 | 0% |
| CompetitionDistance | **离最近竞争者距离（m）** | **离散** | **0%** |
| CompetitionOpenSinceMonth | 最近的竞争者开店的月份 | 离散 | 0% |
| CompetitionOpenSinceYear | **最近的竞争者开店的年份** | **离散** | **0%** |
| Promo2 | 是否参加持续性的促销活动（0未参加，1参加） | 是否 | 0% |
| Promo2SinceWeek | **商店参加促销活动的周** | **离散** | **0%** |
| Promo2SinceYear | 商店参加促销活动的年 | 离散 | 0% |
| PromoInterval | **描述Promo2启动的连续间隔，命名重新开始促销的月份。例如“2月，5月，8月，11月”意味着每一轮在该商店的任何给定年份的2月，5月，8月和11月开始** | **离散** | **0%** |

* + 1. **数据描述性统计**

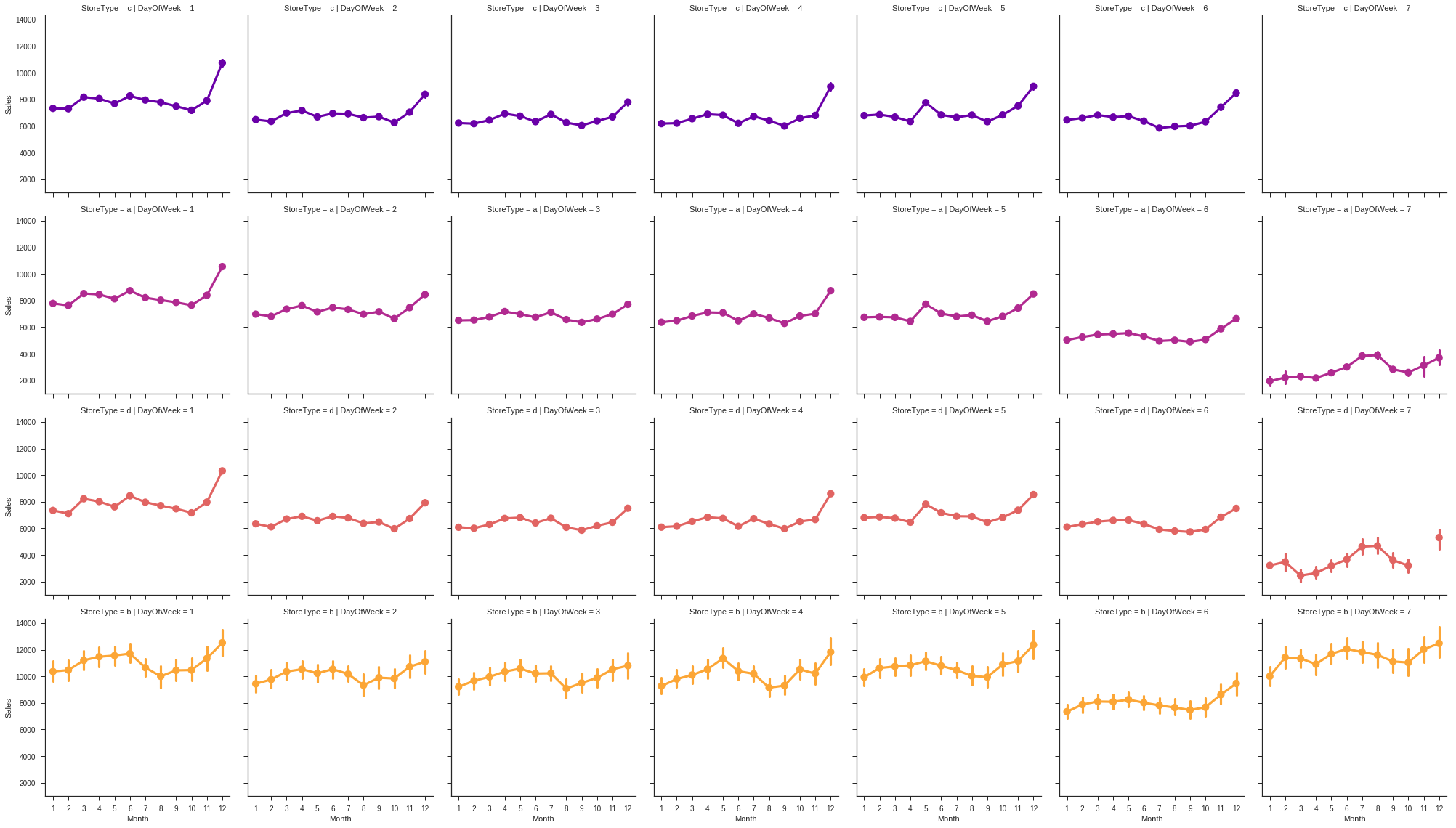
能够找到的比较好的思路主要是集中于：Time Series Analysis。

本处放的是来自Elena Petrova 的分享，使用了facebook 推出的*new methodology Prophet*。

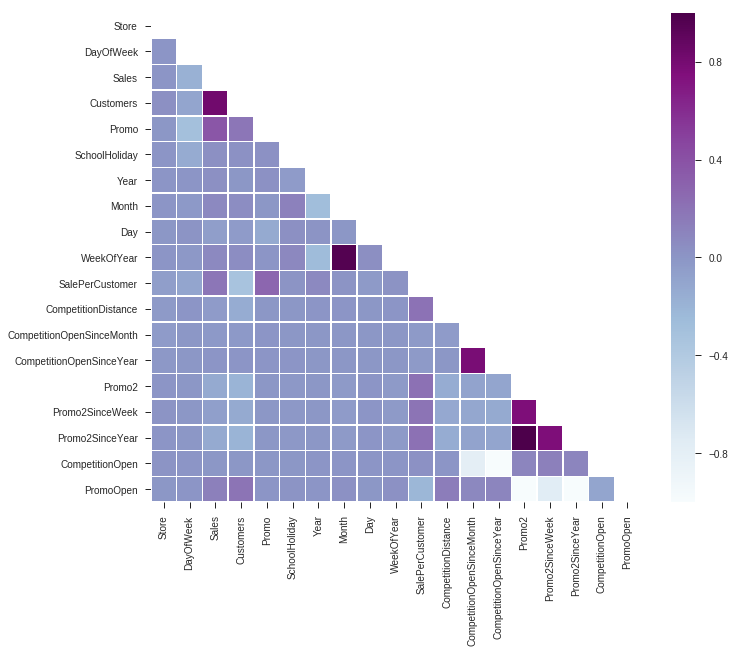
<https://www.kaggle.com/elenapetrova/time-series-analysis-and-forecasts-with-prophet/data>



**图2-2-1 经验累积分布函数（ECDF）**

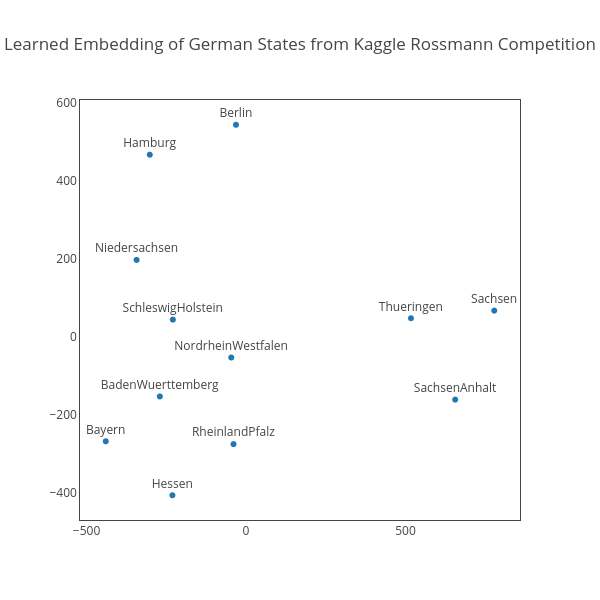


**图2-2-2 商店类型（store types）**

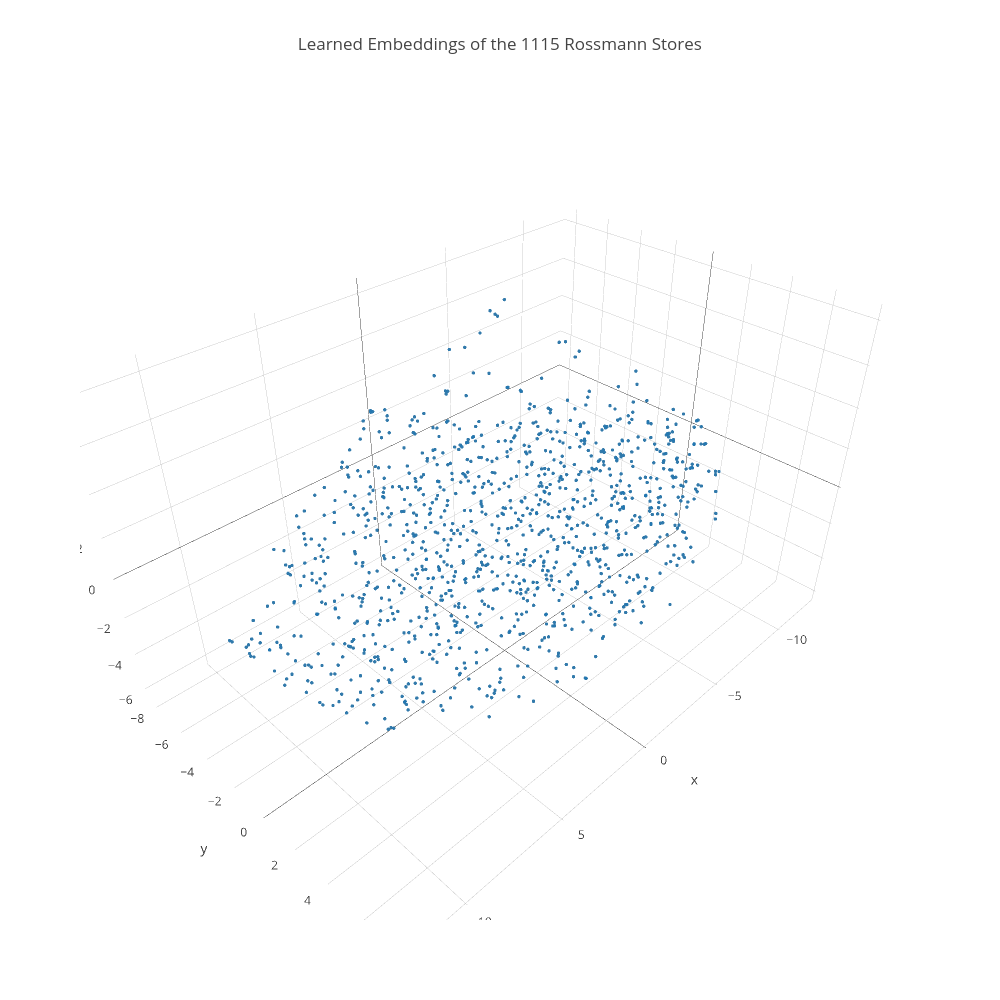


**图2-2-3 相关性分析（Correlational Analysis）**

第三名的数据：



**图2-2-4 从表中读出的德国各个州的信息**



**图2-2-5 1115个Rossmann Stores 的3D图象**

<https://github.com/entron/entity-embedding-rossmann>

3. 优秀算法思路

**3.1 方案一**

**3.1.1** 方案一数据预处理及特征工程部分方案

在本次比赛中，我不得不基于近三年商店的销售历史预测超过1000名罗斯曼店面的6周的销售额。我使用的模型是极端的梯度增强（XGBoost）一种基于决策的通用工具树木。我的特征生成由三个主要原则指导：每列火车和测试记录，该模型应具有以下特征：1）最近的数据2）时间信息3）目前的趋势。对于特征选择和模型集成，我充分利用了过去六周的销售历史。

*功能选择/提取*

对于特征提取，我区分了三种类型的特征：1）最近的数据2）时间信息3）当前趋势。

1） 最近的数据

为了在最近的数据上创建特征，我选择了列车组中每个月的商店特定销售数据集（即销售历史的三年）。然后，对于每条记录，我记录了该记录的日期，并使用了上个月的数据，并将其作为该记录的最近历史记录。我在上个季度，上半年，去年和过去2年中提取了功能。我上个月也只进行过实验，但是认为预测销售额提前六周是没有用的。最近关于商店的数据仍然变化很大，我发现了三个对这种差异贡献最大的特征：星期几，促销和假期。因此，我通过这三个变量进一步拆分了我的商店特定集合，并计算了它们的不同组合的近期平均值。为了总结最近的数据，我使用了中心度量：中位数，平均值和调和平均值，以及传播度量：标准偏差，偏度，峰度和10％/ 90％百分位数。我还尝试在汇总之前记录转换销售，但只有一个转换变量在特征选择中幸存。在主模型的一个变体中，我计算了最近的客户数量的数据特征，而不是销售额。

2） 时间信息

对于时间信息，我创建了“日期计数器”来表示每个记录与某些事件或周期的关系。日期计数器指示事件之前，之后或之内的天数。作为活动，我有促销周期（每14天），次要促销周期（每三个月），暑假（重要的是因为它们部分发生在6周测试集中），商店翻新，比赛开始和开始 二次晋升周期。还有一周中的一天，一天中的某一天以及一年中的一天/一周/一个月。除了白天的柜台，我还增加了本周，上周和下周假期数量的功能。

3） 当前趋势。

为了估计商店销售趋势，我使用了有关上一季度和去年的数据集，类似于最近数据功能的数据集。在每个数据集中，我基于1）日期编号- 将趋势推断为六周时段2）星期几3）促销 上拟合商店特定的线性模型。作为线性模型，我使用了来自scikits-learn的Ridge回归和默认正则化参数，我没有尝试任何替代方案。对于每家商店，我还计算了上个月的年度趋势，但该功能似乎不太重要。

4） 其他信息

其他重要信息是关于商店的：数据集变量存储id，分类和商店类型，以及商店的一些聚合：每个客户的平均销售额，促销/假期/星期六的销售比例，学校假期的比例和商店开放的日子。最后，它有助于包括有关州特定天气的数据：从论坛下载的最高温度和mm降水量。

5） 功能和模型集成的选择

我创建了比模型可以处理的更多功能 - 特别是时间信息。将所有特征组合在一起，模型很容易过度匹配列车组，导致测试装置的性能欠佳。因此，我需要一种方法来减少功能集并选择最有助于预测到测试集中的功能。作为测试集的代理，我使用了我的验证集，包括列车集的最后6周。我开始采用一些似乎对我有意义的组合。这样我很快就注意到传播特征使得过度拟合变得容易。精挑细算花了我很多时间，我意识到它受到了我的想法的偏见。因此，我决定在随机选择的特征上创建一些模型。一些表现最好的随机模型在我使用精心挑选的模型进行整合时给出了很好的改进。注意到这一点后，我运行了500多个随机模型并系统地计算了每个模型的模型上的验证错误。从最好的模型对中我构建了一个更大的整体，由10多个不同的模型组成（实际上是具有不同特征的相同模型）。然后我想到将所有选定模型的特征组合在一起并将它们组合成一个模型：结果非常好-最后我只将这个组合模型与两个精心挑选的模型结合在一起。在整个过程中，模型集成和特征选择齐头并进。

**3.1.2** 方案一模型设计、建立部分方案

对于建模我完全依赖于XGBoost，我的重点完全在于特征提取和选择。 我从（in）着名的公共剧本中的参数开始，这些参数在公共排行榜上表现得比在我的坚持集中重现的要好得多。 在一些实验中，我只改变了轮数（从3000到5000）和de column sample by tree（从0.7到0.3，因为我有很多功能）。

为了进一步丰富和稳定我的模型组合，我添加了单独的模型，这些模型仅在五年到九月期间接受过三年的培训。我之所以选择这几个月，是因为它们涵盖了测试集，并且在测试集之前有一些2015年的历史。另一个合奏的伎俩是增加“月前模型”，跳过最近一个月来计算近期/趋势特征。这些“月前模型”几乎与使用所有最新数据的模型一样好。为了预测测试集第二个月的销售额（最近一个月的销售额根本没有），我使用的是仅有“月前模型”的集合。

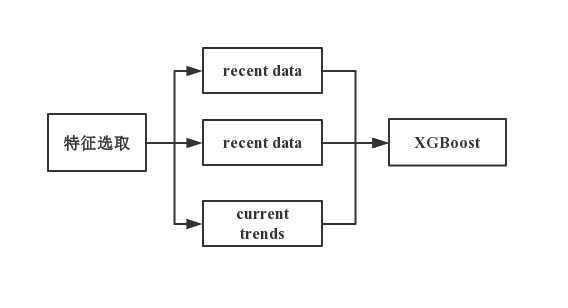
像大多数其他团队一样，我记录转换因变量（销售）并且不包括用于训练模型的零销售。对于最终的合奏，我应用了调和均值。我还将所有预测乘以常数因子来改进它们。对于我的验证集和公共测试集，因子0.995是最佳的。但我应该在论坛上应用0.985的数学估计，以获得远低于0.10的私人排行榜得分。

**3.1.3** 方案一结果、排名等

最后结果：0.10021

排名：1/3500

**3.1.4** 方案一算法流程图



**表3-1 流程图**

链接：[https://pan.baidu.com/s/1kBfCmoGptdcarVzDmFUp-Q](https://pan.baidu.com/s/1kBfCmoGptdcarVzDmFUp-Q%20)

提取码：56z9

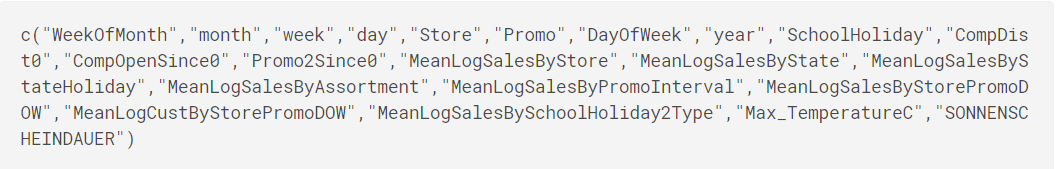
**3.2 方案二**

**3.2.1** 方案二数据预处理及特征工程部分方案

对于CV，我开始使用过去六周作为坚持，但是卡住了一段时间并关注2015年7月第一周的高分（特别是7月4日，一个炎热的星期六，销量很低）。 所以我最终改为最后三周。 它的局部分数通常接近LB分数，但我确实遇到了一些不一致的模型。

对于异常值，我最终筛选出销售量<900的培训观察结果。销售额<500的七个点很麻烦，但似乎对我早期的一些模型有所帮助。

对于功能，我花了一些时间进行基本的功能工程，但更多的是功能选择，使用Occam’sRazor，希望它不会变成Sweeney Todd’s的。 我最好的single model（0.09727公开，0.10713私人，只是瞥见Gertland）只有22个功能：



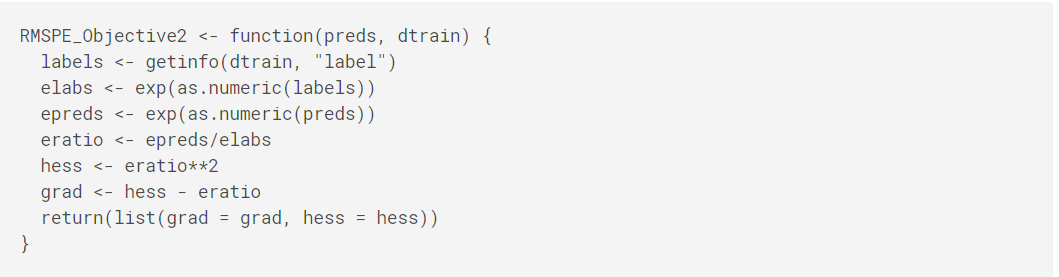
最后一个来自DWD并且给了我一个LOL：如果你的模型感觉阴沉，只需添加一点阳光！ 为了安全起见，我在进行简历时计算平均特征而不使用保持数据，然后在构建提交时切换到完整的训练数据。 最后一个特征是使用Tobias Wolfanger友好提供的学校假期类型构建的（他和我必须是远房表兄弟;）。 模型的简单性可能有助于他们进行推广，特别是在公共 - 私人分割点附近发生销售的最低点。

**3.2.2** 方案二模型设计、建立部分方案

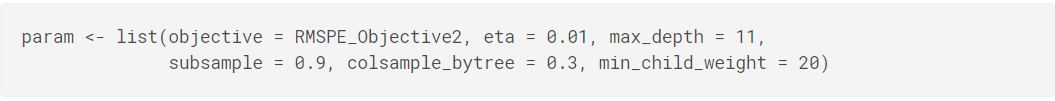
像大多数人一样，我使用了xgboost模型的集合，大约六个模型的几何加权平均值，每个模型不超过25个特征。 这些数据的时间序列横截面（又称面板）结构带来了潘多拉盒子的优良特性，我现在看到我只是部分打开了它。

**XGBoost Details：**

我在对数刻度上建模销售并使用自定义目标函数：

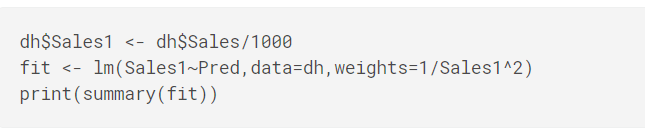


注意这里的Hessian是不正确的，因为它忽略了链规则; 它比正确的更平坦，似乎表现更好。我做了一些调整并确定了以下参数：



在20K轮次和早期停止900.每次运行在一些旧的Windows服务器上花了6-7个小时，并且使事情变得更加困难，结果有一些不稳定因此我最终得到了来自3个不同种子的平均结果。 有一次，我有36个并发R会话运行:)。

估计我使用的线性调整因子：



在保持结果上，采用截距和斜率估计，并将它们应用于相应的LB预测。 典型值分别为-0.05和0.995。

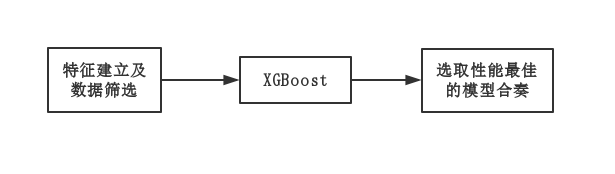
随着时间的推移，我使用性能最佳的模型制作了简单的合奏。

**3.2.3** 方案二结果、排名等

最后结果：0.10621

排名：4/3500

**3.2.4** 方案二算法流程图



**表3-2 流程图**

4. 算法比较

**表4-1 算法比较**

|  | **评估指标** | **特征工程** | **基础算法** | **基本库** |
| --- | --- | --- | --- | --- |
| **算法1** | **0.10021** | **XGboost** | **Xgboost,** | **Sklearn, xsboost** |
| **算法2** | **0.10621** | **XGboost** | **XGboost** | **JMP: data exploration, graphics, quick interactive analyses,**  **submission checking and ensembling**  **SAS: data preparation, linear and**  **time-series modeling**  **R: xgboost** |

制胜之处还是在参赛者的特征工程，排名第一的参赛者显然做了更加独特的特征工程，并且第一名利用权重进行计算，把XGBOOST,NN 和 ET三个模型的交叉验证预测集作为结果。

5. 总结与展望

**5.1 总结**

在查询过程中，我了解到这是kaggle最早也是最少欢迎的几个比赛之一，因而上面提供的两个方案能够在三千多支队伍中脱颖而出就显得格外不容易。对数据细致的处理是第一名的制胜法宝，而第四名的奥妙则是更好的使用了kaggle。

在收集整理的过程中也学习到了这些优秀参赛者的一些方法和经验，为今后的学习和工作提供了宝贵的经验。

**5.2 建模思路**

在学习了两位选手的思路后，我无疑会采用堆叠的方式获取足够多的特征组合，并且使用XGBoost进行计算从而获得更好的效果。