机器学习工程师纳米学位

报告人/时间

何龙

2018-09-28

开题报告

领域背景

Rossmann 在全欧洲有超过 6000 家药店，预测销售额一直是他们商店经理的工作，他们根据直觉来预测，准确率有很大变化，现在我们要帮助构建一个销售额预测模型，针对位于德国的 1115 家店进行 6 周的销售额预测，对于销售额的预测可以帮助经理们更合理的安排员工上班时间表、销售活动等。

使用机器学习去预测能够达到人不能做到的准确率，因为模型考虑了整个分布在德国的商店的数据，并结合这些数据得到结论，相比较，商店经理更多的只是依赖自己管理的商店的数据，我们直到在数据量少的时候结果是无法泛化的，准确率也就无法保证，这是一个典型的机器学习有监督的回归问题，相关算法模型已经非常成熟，同时拥有 Rossmann 提供的大量数据保证了这一问题是可以解决的。

这是一个非常普遍的机器学习应用场景，相信在现实中也有无数公司在利用机器学习进行自己的销售额预测，解决一个现实的问题会让我更加对机器学习有自信，相信它是未来社会不可或缺的工具，并且跟人们的生活息息相关，解决起来体会会更深。

问题描述

针对位于德国的 1115 家店进行 6 周的销售额预测，销售额是一个数值型数据，所需数据 kaggle 已经提供，其中包含训练数据、商店数据、测试数据，根据评价指标，也就是预测值与实际值的差值越小越好。

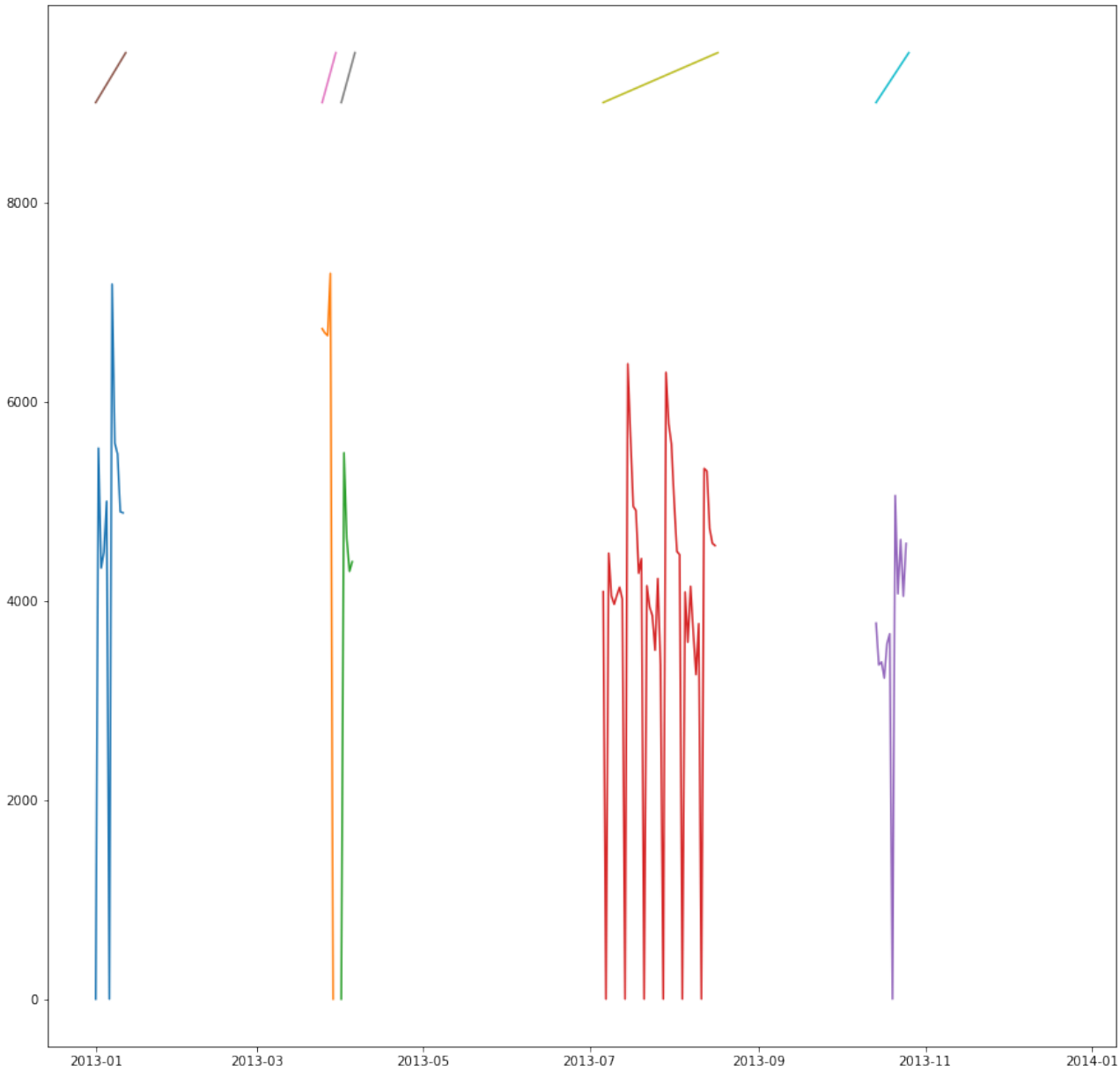
数据集和输入

本项目数据集来源于 kaggle，数据本身是针对该项目收集的数据，所以可以使用他们来进行模型的训练，对于额外的 store.csv 数据，它是针对每个商店的具体信息，既然是分析商店的销售额，它肯定是非常有用的，因此要想办法结合进 train.csv 中一起处理，同时对于类别数据要多关注，时间序列相关的字段也要特别处理。

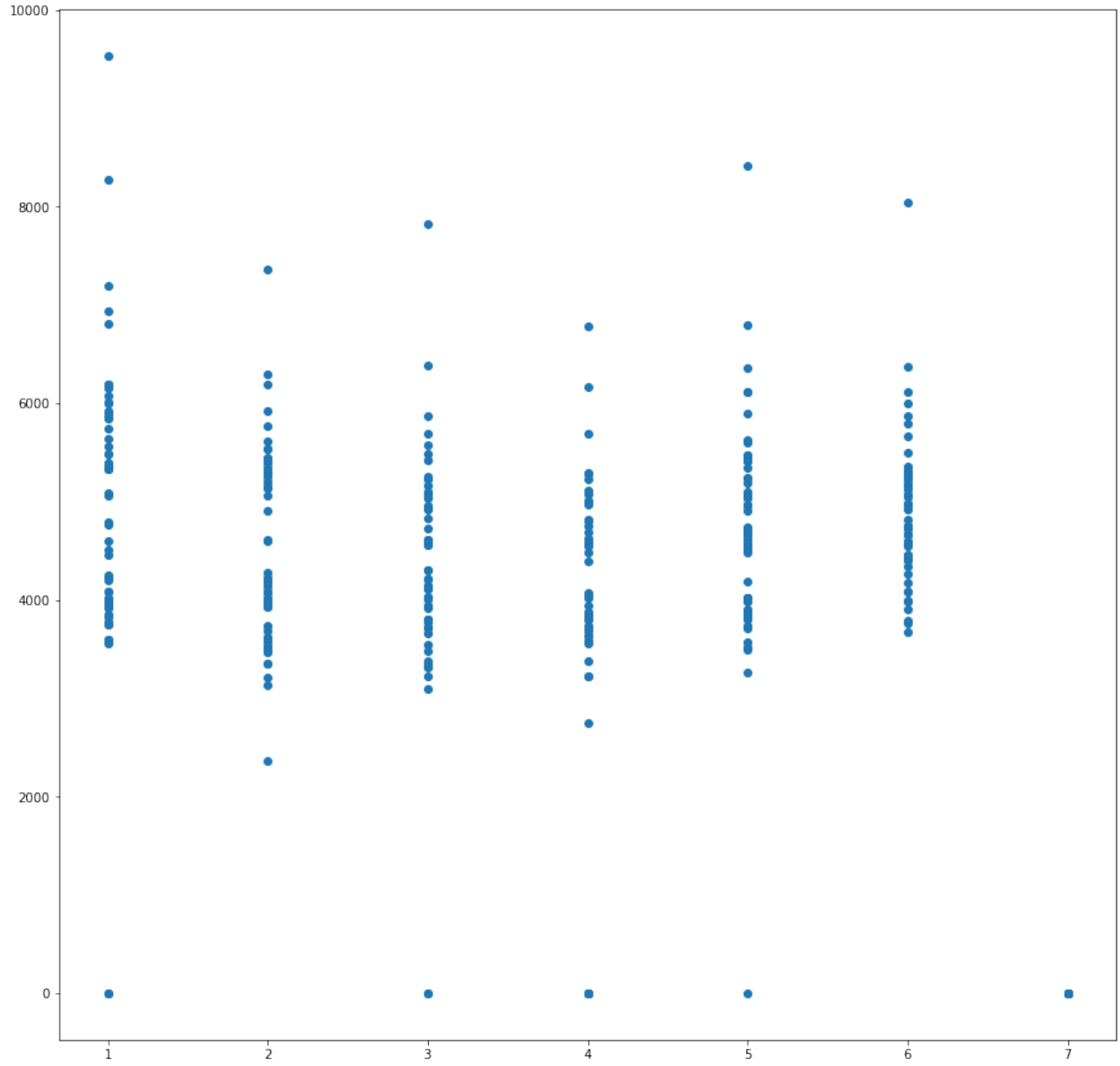
使用数据文件总共有三个：store.csv,train.csv,test.csv，其中 train.csv 和 test.csv 分别用于训练以及最终的测试，而 store.csv 表示每家店的信息，需要我们跟 train.csv、test.csv 结合使用，下面主要分析 train.csv 和 store.csv，test.csv 跟 train.csv 结构是基本一致的。

Train.csv 共有 1017209 条数据，时间跨度从 13 年 1 月 1 号到 15 年 7 月 31 号大概两年半的时间，好消息是 train.csv 中没有 null 数据。而 store.csv 共有 1115 条数据，代表 1115 家店，其中跟竞争对手、促销活动相关的字段存在 null 值，与竞争对手相关的字段可以使用平均值填充，而跟促销相关的字段存在 null 值是因为对应店铺没有参与促销活动，因此不能也不需要填充。

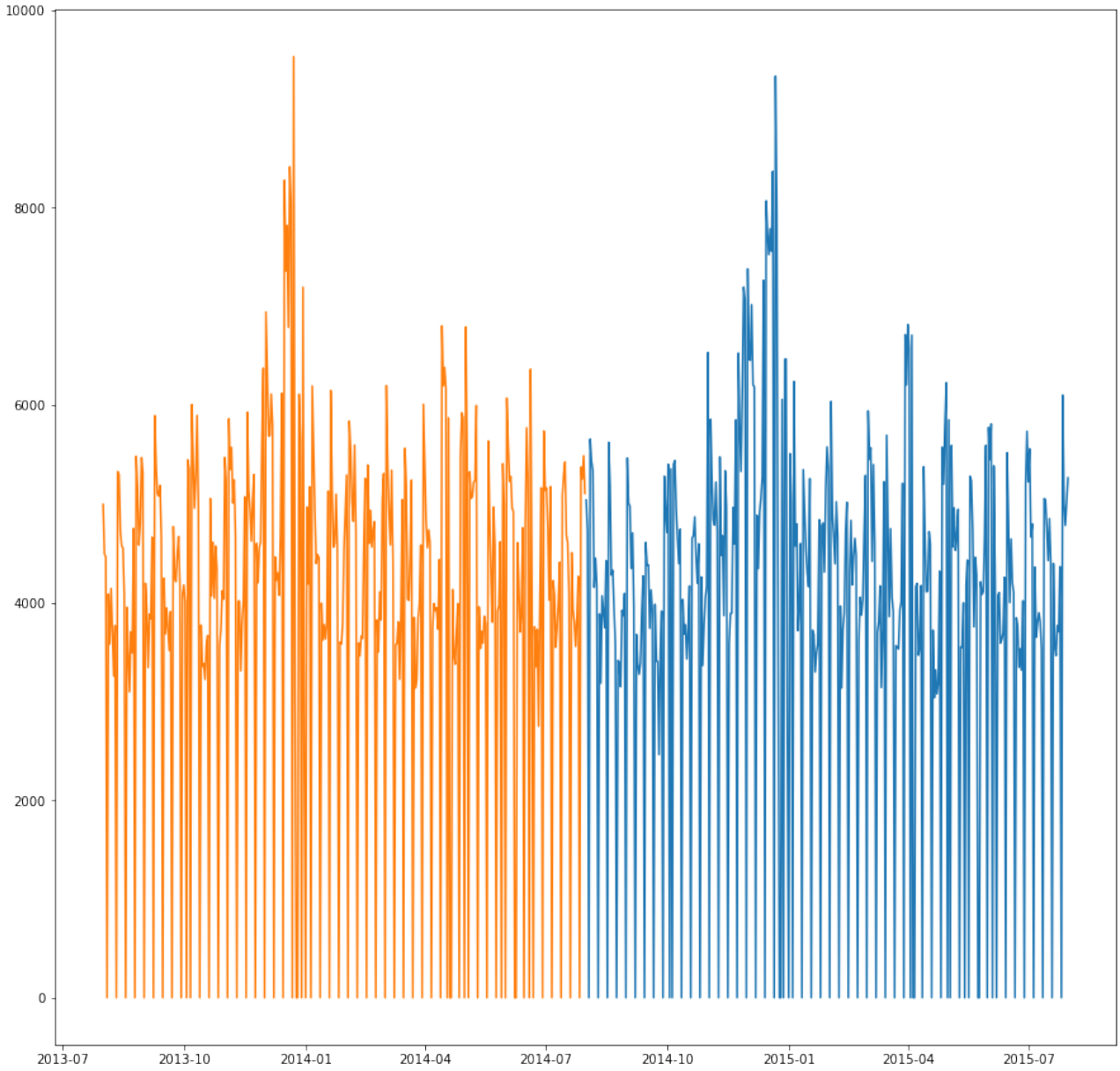
下面我们可视化一些数据的特点，对数据进行分析、挖掘。首先，我们根据是否处于假期来查看一下假期期间的销售额的变化吧：



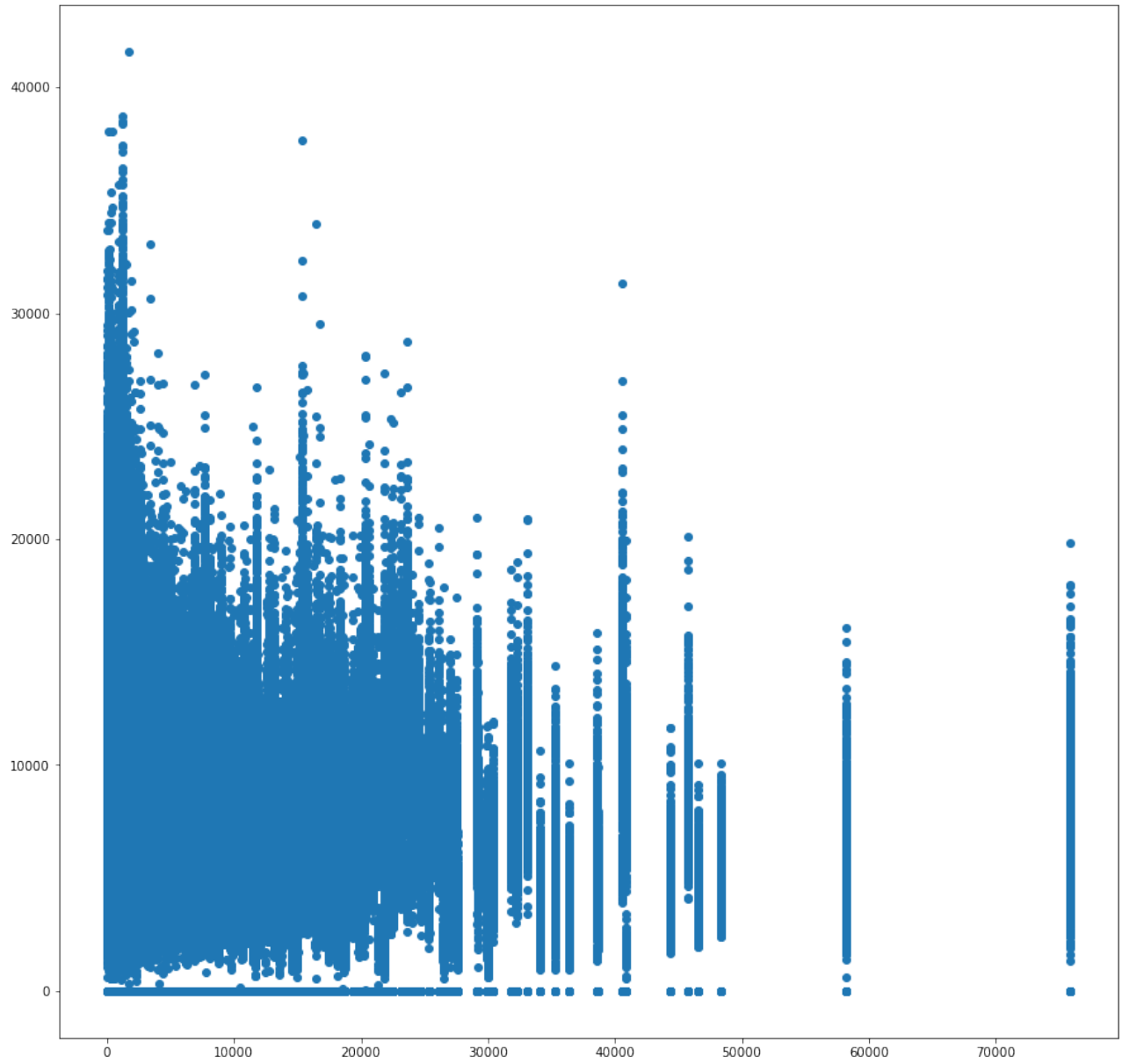
上方的每一条短线，表示一个假期的开始到结束，对应它下方就是假期期间的销售额，并不能看到明显的趋势变化，因此认为这一特点对于我们的预测帮助不大，接下来看看根据是星期几绘制的销售额图：



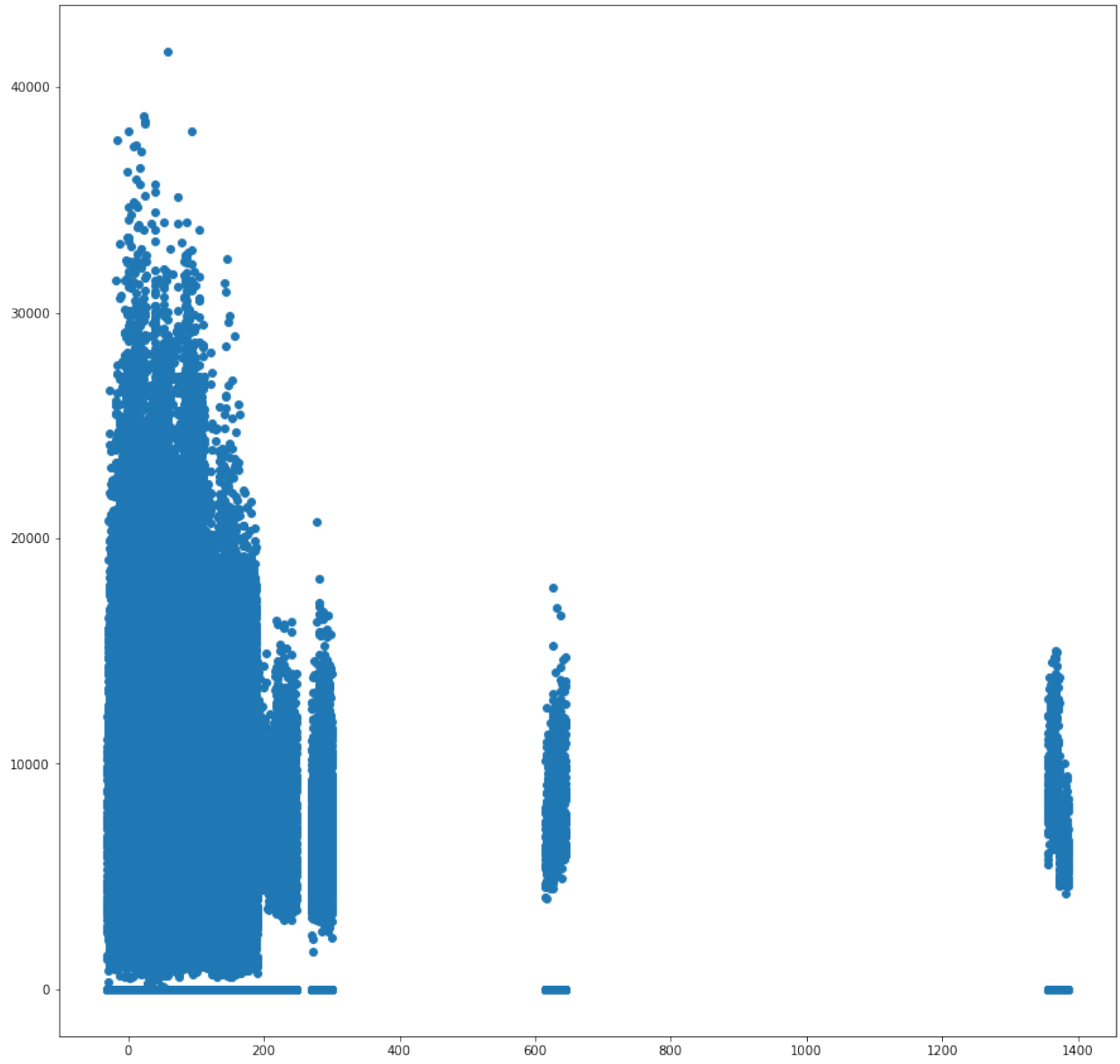
可以看到首先，星期天都是 0 表示不开门，而其他时间貌似看不出具体差别，但是我们可以利用是否是星期天来帮助我们预测，因为星期天都是 0 而跟其他时间有很明显的差别。下面我们看看两年之间同期销售额趋势对比，看一下销售额是否具有某种周期性：



这个图就很明显的能看到，不能的两年之间，销售额的整体趋势非常相似，这就说明销售额跟具体日期有很大关系，也就是具有一定的季节性等，这一点体现在现实中，很可能是因为发病的情况跟季节相关性比较强，比如流感、过敏等等都是季节强相关的，因此对于 Date 字段，我们要多多挖掘出更深层次的时间信息。再来看看竞争对手对我们销售额的影响，首先是竞争对手的距离：



很特别的一点是，数据显示更多的情况是对手距离我们越近，我们的销售额更高，这个可能是一种类似某种商业圈的影响，假如这一地区药店比较多、比较密集的话，那么也许人们也就更倾向于来这里买药，那么也变向提高了我们的销售额，再来看看竞争对手的开张时间：



这个就很好理解了，开张时间更短的竞争对手，我们的销售额更高，也说明了客户更支持老字号。到此，我们可以看到可以挖掘的信息包括具体的时间信息（年、月、日等）、是否是周末、以及竞争对手的开张时长等，这些信息都 预测都会有很大的帮助。

解决方案

首先对于数据的预处理中要关注 `store.csv` 以及类别字段、时间序列字段等，由于处理后维度很高，此处需要进行主成分分析，模型上使用集成学习 `XGBoost`，根据 `kaggle` 指定的评价指标 `RMSPE` 进行模型评估以及调参。`XGBoost` 相对于 `Adaboost` 来说有以下优点：`1.`通过计算梯度来定位模型的不足，因此支持更多目标函数，比如我们的性能指标 `RMSPE`，`2.`速度更快，`3.`支持线性分类器，`4.`有更多的超参数可以设置等。

基准模型

基准模型选择恒预测为 `mean` 值，通过计算，这一阈值为 `5774.91505842`，而它对应的 `RMSPE` 值为 `0.457834385457`，后续模型预测结果的 `RMSPE` 值与该值进行比较即可评估我们的模型对于预测的帮助。

评价指标

`RMSPE:sqrt((sum(X_obs_i-X_model_i/X_obs_i)^2)/n)`

`X_obs_i`：表示实际的销售额数值。

`X_model_i`：表示模型预测的销售额数值。

n 表示预测的数据个数，公式本意就是计算所有预测的数值与实际值之间误差的百分比形式的平均值来衡量模型性能，该值越小越好，如果是 `0`，那么就表示预测值与实际值一致。

均方根百分误差用于描述预测值与实际值之间的偏差的百分比形式，应用于本项目即为预测销售额与实际销售额之前的偏差，可以很好的评估模型的性能。

项目设计

1. 工作流程

    0. train.csv、store.csv 数据读取。

    1. 根据 store Id 链接成一个大表。

    2. One-Hot 处理类别字段。

    3. 时间序列信息挖掘。

    4. PCA [1]。

    5. 划分为训练数据和验证数据。

    6. 应用模型训练。

    7. 对比模型与基准模型在评价指标上的得分。

    8. 持续优化模型。

    9. 上传到 kaggle。

2. 采用什么策略

    XGBoost [2]：相比较 Adaboost，继承自 GBDT 的 XGBoost 采用梯度的方式来评估模型的不足，支持更多的的目标函数，同时相比较 GBDT，支持并行，速度更快，可调参数更多。

3. 需要对数据进行哪些前期分析

    0. train、stroe、test 中的缺失值、异常值处理。

    1. 类别字段编码处理。

    2. 时间序列字段信息提取：

        1. 根据 Date 字段挖掘年、月、日、季度、一年的第几周、是否是周末信息。

2. 根据 CompetitionOpenSinceYear、CompetionOpenSinceMonth 挖掘竞争对手的总营业时间。

3. 根据 Promo2SinceYear、Promo2SinceWeek、PromoInterval、Promo2 挖掘当前商店当前时间是否处于促销活动中以及已经促销了多久的信息。

3. PCA 处理：由于对字段进行合理的挖掘以及舍弃，对枚举字段没有进行过度的 OneHot，因此可以不用进行 PCA 处理，毕竟 PCA 对于数据包含的信息量是有一定程度的损失的。

参考文献

【1】PCA：https://github.com/NemoHoHaloAi/machine-learning-plus/blob/master/notes/非监督学习/5.PCA-主成分分析.md

【2】XGBoost 官方文档：https://xgboost.readthedocs.io/en/latest/

【3】Kaggle 数据说明：https://www.kaggle.com/c/rossmann-store-sales/data

【4】Pandas 官方文档：http://pandas.pydata.org/pandas-docs/stable/index.html

【5】Udacity 监督学习笔记：https://github.com/NemoHoHaloAi/machine-learning-plus/tree/master/notes/监督学习