机器学习纳米学位

毕业项目

姜明

2018年5月2日

I. 问题的定义

项目概述

这是Kaggle的一个竞赛项目，目标是帮助Rossmann连锁药妆店的经理预测店铺未来6周的销售额，准确的预测店铺未来一段时间的销售额，能够帮助经理更加合理的安排员工的工作时间，同时也有利于提升员工的工作效率，可以看出这是一个典型的回归问题。

本项目使用的是XGBoost模型，XGBoost方法是由传统的GBDT方法发展而来，而GBDT的基学习器是CART，也就是基本的决策树模型。XGBoost方法最近几年屡次在各大数据比赛中获得第一[1]，该项目来自于kaggle，这个比赛的第一名使用的算法也是xgboost[2]。

该项目使用的数据集是由Kaggle提供，数据来源于坐落在德国的1115家店铺的销售数据（train.csv）和店铺信息（store.csv）数据。train.csv是店铺的销售数据，包含了每个店铺每天的**销售额**以及其他特征信息，比如说是否节假日，当天是否有促销活动等，该数据集中一共有1017209条数据，时间跨度为2013-01-01到2015-07-31。store.csv是1115家店铺的信息数据，共有1115条数据，主要特征有竞争对手开业时间，竞争对手距离，是否有长期促销活动等。模型需要预测的是未来6周每个店铺每天的销售额。

问题陈述

本项目的数据集来自真实世界中Rossmann连锁药妆店的销售数据和店铺数据，目标是要预测1115家店铺未来6周的销售额，这是一个标准的监督学习问题，同时也是一个时间序列的回归问题。

本项目一个很大的挑战在于，我们需要从现有的数据集中探索出店铺销售额的影响因素，从而才能预测店铺的销售额。另一个挑战在于，其预测周期为6周，较长的预测周期很有可能会给预测带来较大的误差，怎么才能找到影响店铺长期销售额的因素也很重要。

总之，大量的特征工程和数据分析需要花费较多的时间。寻找影响店铺销售额的影响因素需要从现有数据集的字段出发，比如说，可以从store.csv数据集获取更多店铺的信息，店铺的分类水平(Assortment)，促销活动(Promo)以及竞争对手店铺的距离(ComprtitionDistance)等都是影响销售额的影响因素。参考Gert[3]和Anton[4]的工作来构造更多的特征。

评价指标

本项目的评价指标由kaggle给出， Root Mean Square Percentage Error (RMSPE)，而且有比赛的结果作为对比。因此，选择RMSPE作为评价指标，定义为：

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

其中y_i表示真实的销售额，$\hat{y}_i$表示销售额的预测值。当销售额为0时，不带入计算。该指标的含义是预测值与真实值之间的误差占真实值比例的反映，RMSPE的值越小代表预测越准确。

II. 分析

数据的探索

本项目主要使用两个数据集：train.csv（train）和store.csv（store）。其中train代表店铺的销售数据，store代表店铺的信息。

	Store	DayOfWeek	Date	Sales	Customers	Open	Promo	StateHoliday	SchoolHoliday
0	1	5	2015-07-31	5263	555	1	1	0	1
1	2	5	2015-07-31	6064	625	1	1	0	1
2	3	5	2015-07-31	8314	821	1	1	0	1
3	4	5	2015-07-31	13995	1498	1	1	0	1
4	5	5	2015-07-31	4822	559	1	1	0	1

表 1 train的前6条数据

表1展示的是train的前6条数据，从表中可以看出该数据集一共包含了9个字段，每个字段的含义为：Store（店铺ID），DayOfWeek（星期几，5代表周五），Date（日期），Sales（销售额），Customers（到店人数），Open（是否开业，1代表开业，0代表未开业），Promo（是否有促销，1代表有促销活动，0反之）等，关于个字段的含义详见Rossmann Store Sales Data。

本项目要预测的是Sales，由于店铺会有不营业的情况出现，为了体现其销售额真实的统计量，因此先去掉Sales=0的数据，其最小值、最大值、均值、中值和标准差分别为：

表 2 Sales的描述性统计

最大值	最小值	均值	中位数	标准差
41,551.00	46.00	6,955.96	6,369.00	3,103.82

图1给出了Sales频率的频率直方图

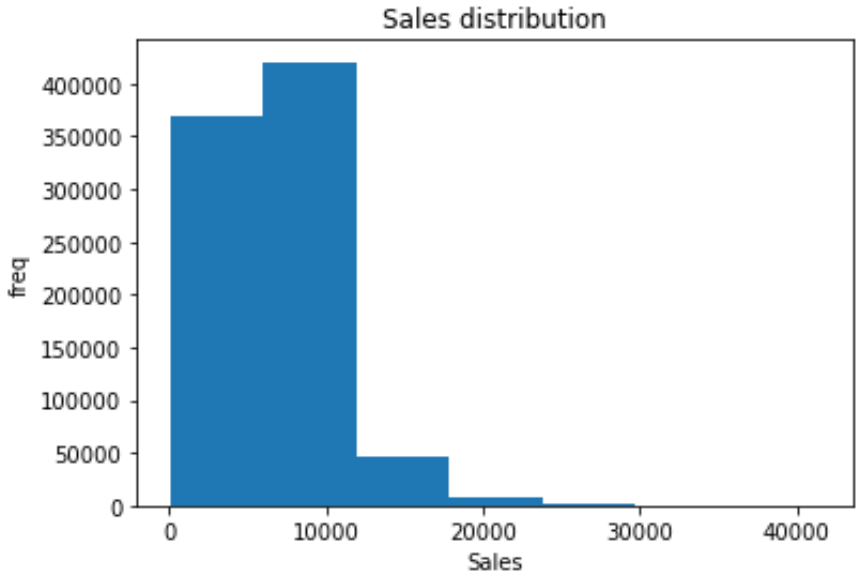


图 1 Sales的频率直方图

从表2和图1可以发现，大部分店铺的销售额在10000左右。

	Store	StoreType	Assortment	CompetitionDistance	CompetitionOpenSinceMonth	CompetitionOpenSinceYear	Promo2	Promo2SinceWeek	Promo2SinceYear
0	1	c	a	1270.0	9	2008	0	NaN	NaN
1	2	a	a	570.0	11	2007	1	13.0	2010.0
2	3	a	a	14130.0	12	2006	1	14.0	2011.0
3	4	c	c	620.0	9	2009	0	NaN	NaN
4	5	a	a	29910.0	4	2015	0	NaN	NaN

PromoInterval
NaN
Jan,Apr,Jul,Oct
Jan,Apr,Jul,Oct
NaN
NaN

表 3 store的前6条数据

表3展示的是store的前6条数据，从表中可以看出，该数据集一共包含了10个字段，每个字段的含义为：Store（店铺ID），StoreType（陈列水平），Assortment（摆放水平）等，对于每个字段包含的详细信息，详见Rossmann Store Sales Data。

算法和技术

本项目选择xgboost模型来解决该问题，xgboost是以CART作为基分类器的。CART能够解决回归问题，并且决策树算法能够给出每个特征的重要程度，有利于模型的特征筛选。同时，xgboost模型借鉴了随机森林的思想，支持列抽样，不仅能够防止过拟合，还能减少计算量。本项目的一个难点就在于特征的构造，然而可以利用xgboost的特点，来找出影响销售额的重要变量。

根据前面的分析可知，该问题是一个时间序列的回归问题。用于回归的算法有很多，比如ARIMA模型、支持向量回归、均值回归、神经网络和CART等，这些算法统称为机器学习算法或统计学习算法。本项目使用xgboost模型来解决该问题，xgboost是以CART作为基分类器的。CART能够解决回归问题，并且决策树算法能够给出每个特征的重要程度，有利于模型的特征筛选。而xgboost属于boosting模型，它基本思想是把成百上千个准确率较低的树模型组合起来，成为一个准确率很高的模型。这个模型会不断的迭代，每次迭代就生成一颗新的树。下面就来介绍一下xgboost模型[5]。

监督学习[6]

在讲解xgboost之前要从监督学习介绍，下面从模型，参数和目标函数三个方面来介绍监督学习。

模型是指给定输入x_i如何去预测输出y_i。比较常见的模型如线性模型（包括线性回归和logistic regression）采用了线性叠加的方式进行预测$\hat{y_i} = \Sigma w_j x_{ij}$。对于这里的预测值$y$，根据应用场景的不同，可以给出不同的解释，比如说它既可以是回归目标的输出，也可以通过sigmoid函数的变换得到概率，也可以作为排序的指标等。在模型中，我们需要学习的就是参数，而这个线性模型中，参数是指线性系数w。

目标函数通常的表示方法为$Obj(\theta) = L(\theta) + \Omega(\theta)$，主要由两部分组成，损失函数$L(\theta)$和正则化项$\Omega(\theta)$。在给定输入的情况下，可以通过模型和参数本身来预测，但是没有给定一个比较好的参数来实现比较准确的预测，这个时候就需要目标函数了。目标函数的作用就在于用来优化出模型最优的参数，从而使得模型实现较为准确的预测结果。常见的损失函数有平方损失$L = \Sigma(y_i - \hat{y_i})^2$，logistic损失函数$L = \Sigma(y_i ln(1 + e^{-\hat{y_i}}) + (1 - y_i) ln(1 + e^{\hat{y_i}})$等。而对于线性模型常见的正则化项有$L2$正则和$L1$正则。这样目标函数的设计来自于统计学习里面的一个重要概念叫做Bias-variance tradeoff。比较感性的理解，Bias可以理解为假设我们有无限多数据的时候，可以训练出最好的模型所拿到的误差。而Variance是因为我们只有有限数据，其中随机性带来的误差。目标中误差函数鼓励我们的模型尽量去拟合训练数据，这样相对来说最后的模型会有比较少的 bias。而正则化项则鼓励更加简单的模型。因为当模型简单之后，有限数据拟合出来结果的随机性比较小，不容易过拟合，使得最后模型的预测更加稳定。

优化算法是在给定目标函数后，来解决如何学习出最优参数的方法。常见的优化算法有梯度下降算法、拟牛顿算法插值法等。

Boosted Tree[6][7]

Boosted tree 最基本的组成部分叫做回归树(regression tree)，也叫做CART。CART假设决策树是二叉树，内部节点特征的取值为“是”和“否”，这样的决策树等价于递归的二分每个特征，将输入空间划分为有限个单元，并在这些单元上确定预测的概率分布，也就是在输入给定的条件下输出的条件概率分布。CART有特征选择、树的生成及剪纸组成，即可以用于分类也可以用于回归。

一个CART往往过于简单无法有效地预测，因此一个更加强力的模型叫做tree ensemble。对于tree ensemble，可以比较严格的把我们的模型写成是：

$$\hat{y_i} = \Sigma f_k(x_i), f_k$$

其中每个f是一个在函数空间F里面的函数，而F对应了所有CART的集合。设计的目标函数也需要遵循前面的主要原则，包含两部分

$$Obj(\theta) = \Sigma l(y_i, \hat{y_i}) + \Sigma \Omega(f_k)$$

其中第一部分$\Sigma l(y_i, \hat{y_i})$是训练误差，第二部分$\Sigma \Omega(f_k)$是每棵树的复杂度的和。因为现在模型的参数可以认为是在一个函数空间里面，不能采用传统的如SGD之类的算法来学习该模型，因此采用一种叫做additive training的方式。每一次保留原来的模型不变，加入一个新的函数f到模型中。

$$\begin{aligned}
\hat{y}_i^{(0)} &= 0 \\
\hat{y}_i^{(1)} &= f_1(x_i) = \hat{y}_i^{(0)} + f_1(x_i) \\
\hat{y}_i^{(2)} &= f_1(x_i) + f_2(x_i) = \hat{y}_i^{(1)} + f_2(x_i) \\
&\cdots \\
\hat{y}_i^{(t)} &= \sum_{k=1}^{t} f_k(x_i) = \hat{y}_i^{(t-1)} + f_t(x_i)
\end{aligned}$$

加入一个新的函数

第t轮的模型预测

保留前面t-1轮的模型预测

选取一个f来使得我们的目标函数尽量最大地降低。

$$\begin{aligned}
Obj^{(t)} &= \sum_{i=1}^{n} l(y_i, \hat{y}_i^{(t)}) + \sum_{i=1}^{t} \Omega(f_i) \\
&= \sum_{i=1}^{n} l\left(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)\right) + \Omega(f_t) + constant
\end{aligned}$$

目标：找到 f_t 来优化这一目标

更加一般的，会采用如下的泰勒展开近似来定义一个近似的目标函数，方便进行这一步的计算。

- 目标 $Obj^{(t)} = \sum_{i=1}^{n} l\left(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)\right) + \Omega(f_t) + constant$
- 用泰勒展开来近似我们原来的目标
  - 泰勒展开：$f(x + \Delta x) \simeq f(x) + f'(x)\Delta x + \frac{1}{2} f''(x)\Delta x^2$
  - 定义：$g_i = \partial_{\hat{y}^{(t-1)}} l(y_i, \hat{y}^{(t-1)})$，$h_i = \partial^2_{\hat{y}^{(t-1)}} l(y_i, \hat{y}^{(t-1)})$

$$Obj^{(t)} \simeq \sum_{i=1}^{n} \left[l(y_i, \hat{y}_i^{(t-1)}) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \Omega(f_t) + constant$$

当把常数项移除之后，会发现如下一个比较统一的目标函数。这一个目标函数有一个非常明显的特点，它只依赖于每个数据点的在误差函数上的一阶导数和二阶导数。

$$\sum_{i=1}^{n}\left[g_i f_t(x_i)+\frac{1}{2}h_i f_t^2(x_i)\right]+\Omega(f_t)$$

- where $g_i=\partial_{\hat{y}^{(t-1)}}l(y_i,\hat{y}^{(t-1)}),\quad h_i=\partial^2_{\hat{y}^{(t-1)}}l(y_i,\hat{y}^{(t-1)})$

接下来定义树的复杂度，复杂度包含了一棵树里面节点的个数，以及每个树叶子节点上面输出分数的***L2***。

$$\Omega(f_t)=\underbrace{\gamma T}_{\text{叶子的个数}}+\underbrace{\frac{1}{2}\lambda\sum_{j=1}^{T}w_j^2}_{\text{w的L2模平方}}$$

在这种新的定义下，我们可以把目标函数进行如下改写，其中I被定义为每个叶子上面样本集合 $I_j=i|q(x_i)=j$

$$\begin{aligned}Obj^{(t)}&\simeq\sum_{i=1}^{n}\left[g_i f_t(x_i)+\frac{1}{2}h_i f_t^2(x_i)\right]+\Omega(f_t)\\&=\sum_{i=1}^{n}\left[g_i w_{q(x_i)}+\frac{1}{2}h_i w^2_{q(x_i)}\right]+\gamma T+\lambda\frac{1}{2}\sum_{j=1}^{T}w_j^2\\&=\sum_{j=1}^{T}\left[(\sum_{i\in I_j}g_i)w_j+\frac{1}{2}(\sum_{i\in I_j}h_i+\lambda)w_j^2\right]+\gamma T\end{aligned}$$

这一个目标包含了T个相互独立的单变量二次函数。我们可以定义

$$\begin{aligned}Obj^{(t)}&=\sum_{j=1}^{T}[(\sum_{i\in I_j}g_i)w_j+\frac{1}{2}(\sum_{i\in I_j}h_i+\lambda)w_j^2]+\gamma T\\&=\sum_{j=1}^{T}[G_j w_j+\frac{1}{2}(H_j+\lambda)w_j^2]+\gamma T\end{aligned}$$

上述的主要内容是算法设计的推到和技术。xgboost是大规模并行boosted tree的工具，在数据科学方面，有大量kaggle选手选用它进行数据挖掘比赛，其中包括两个以上kaggle比赛的夺冠方案。

参数调节[8]

本项目中将会使用Python里sklearn库中的xgboost模型来训练，xgboost方法中涉及的参数众多，这里介绍以下几个重要参数。

- min_child_weight：这个参数默认是 1，是每个叶子里面 h 的和至少是多少，对正负样本不均衡时的 0-1 分类而言，假设 h 在 0.01 附近， min_child_weight 为 1 意味着叶子节点中最少需要包含 100 个样本。这个参数非常影响结果，控制叶子节点中二阶导的和的最小值，该参数值越小，越容易 overfitting。
- eta：shrinkage 参数，用于更新叶子节点权重时，乘以该系数，避免步长过大。参数值越大，越可能无法收敛。把学习率 eta 设置的小一些， 小学习率可以使得后面的学习更加仔细。
- max_delta_step：如果设立了该值，对叶子节点的权重值做了约束在 [max_delta_step,max_delta_step]。以防在某些 loss 下权重值过大，默认是 0（其实代表 inf)。可以试试把这个参数设置到 1-10 之间的一个值。这样会防止做太大的更新步子，使得更新更加平缓。
- lambda ： 控制模型复杂程度的权重值的 L2 正则项参数， 参数值越大，模型越不容易 overfitting。

- alpha： 控制模型复杂程度的权重值的 L1 正则项参数， 参数值越大，模型越不容易 overfitting。
- gamma： 后剪枝时，用于控制是否后剪枝的参数。
- max_depth：gbdt 每颗树的最大深度，树高越深，越容易过拟合。
- num_round：gbdt 的棵数，棵数越多，训练误差越小，但是棵数过多容易过拟合。需要同时观察训练 loss 和测试 loss，确定最佳的棵数。

对于模型参数设置，本文参考Abhilash Awasthi[9]的脚本来设计。

基准模型

在kaggle上，Gert在本项目上获得了第一名的成绩。以RMSPE为评价指标，其得分为0.10021。Gert[2]使用的就是集成的xgboost 模型，也就是多个采用不同特征的xgboost模型集成。该比赛获得金奖的指标最低得分是0.11037，获得银奖的指标最低得分是0.11552，获得铜奖的最低得分是0.11773。由于能够参考很多前人的研究，特别是Gert给出了自己获得第一名的比赛文档，本项目设定的目标为RMSPE指标得分在0.11552以内。

III. 方法

数据预处理

根据数据集已有的字段，进行预处理，构造特征，并且观察模型的训练效果。本项目主要利用两个数据集，train.csv和store.csv。

针对train.csv和store.csv做如下特征处理：

- train.csv有数据缺失，数据集的时间范围内，并不是每个店铺的每一天都有数据的，对缺失的补充0。
- 通过CompetitionOpenSinceYear和CompetitionOpenSinceMonth来得到竞争对手开业的时间CompetitionOpenDate
- 把DayOfWeek，Assortment，StateHoliday处理成one-hot-vector
- CompetitionOpen判断竞争对手是否开门：Date >= CompetitionOpenDate和Open两个一起判断。也就是说首先判断CompetitionOpenDate是否早于Date，如果是，与该日期本店铺的Open一致，如果不是则为未开门0。
- Promo2Active是店铺持续促销活动的在当天是否在进行，而Promo2是一些店铺是否参与持续促销活动，这样一来Promo2就可以不用使用了。判断方法是：当前日期是否大于开始促销的日期，当前月份是否在促销的月份里。如果都满足则为1，否则为0。最后外面还要乘以Promo2

参考Gert[2]的工作，考虑一些特征：

- week:根据Date计算当前日期是这一年的第几周
- year: 根据Date得到当前日期的年份，年份作为特征是否需要数值化进行处理。
- day: 根据Date得到当前日期的是哪一天，因为销售额与日期会有很大的关系，该特征可以体现出周期性。
- month：同day一样，能够从月的角度来体现周期性。

删除一些不需要的特征：

有一些特征通过上面利用到了，需要删除。为了划分训练集合测试集，要保留Date为了划分数据集使用，但是训练和预测时不使用。

- 需要删除的特征有： Promo2SinceWeek, Promo2SinceYear, PromoInterval, Promo2, CompeteOpenDate, CompetitionOpenSinceYear, CompetitionOpenSinceMonth

训练和预测

划分数据集

对数据集all_data进行分割，原则是validation data和test data分别保留6周的数据，因此按照时间划分后，从从2013-01-01到2015-05-08的数据作为train，从2015-05-09到2015-06-19的数据作为valid，从2015-06-20到2015-07-31的数据作为test。

评价指标

本项目的评价指标由kaggle给出， Root Mean Square Percentage Error (RMSPE)，而且有比赛的结果作为对比。因此，选择RMSPE作为评价指标，定义为：

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

其中y_i表示真实的销售额，$\hat{y}_i$表示销售额的预测值。当销售额为0时，不带入计算。该指标的含义是预测值与真实值之间的误差占真实值比例的反映，RMSPE的值越小代表预测越准确。

模型训练

为了减小被解释变量y_i的区间范围，对y_i进行以e为底的对数(ln)处理，在评价指标中将其还原后在计算指标值。模型输入的数据包含的特征有，每个特征的含义如上所述：

```
['Store', 'Sales', 'Customers', 'Open', 'Promo', 'SchoolHoliday',
       'DayOfWeek_1', 'DayOfWeek_2', 'DayOfWeek_3', 'DayOfWeek_4',
       'DayOfWeek_5', 'DayOfWeek_6', 'DayOfWeek_7', 'StateHoliday_0',
       'StateHoliday_a', 'StateHoliday_b', 'StateHoliday_c', 'Assortment',
       'CompetitionDistance', 'StoreType_0', 'StoreType_a', 'StoreType_b',
       'StoreType_c', 'StoreType_d', 'CompeteOpen', 'Promo2Active', 'week',
       'month', 'year', 'day']
```

其中`Sales`作为被解释变量，其余特征为解释变量。该项目使用主要尝试了xgboost模型，使用的参数，参考了Abhilash Awasthi[9]的工作。参数设置如下：

```
param = {'max_depth':10, 'eta':0.02, 'subsample':0.9,
'colsample_bytree':0.7, 'silent':1,  'objective':'reg:linear' , 'booster':
"gbtree"}
```

`num_round = 3000`设置迭代次数为3000。训练时使用的数据为train，为了输出训练时的评价指标结果，将valid作为`watchlist`。

为了能够使用自定义的评价指标evalerror，特别设置了参数feval=evalerror。

IV. 结果

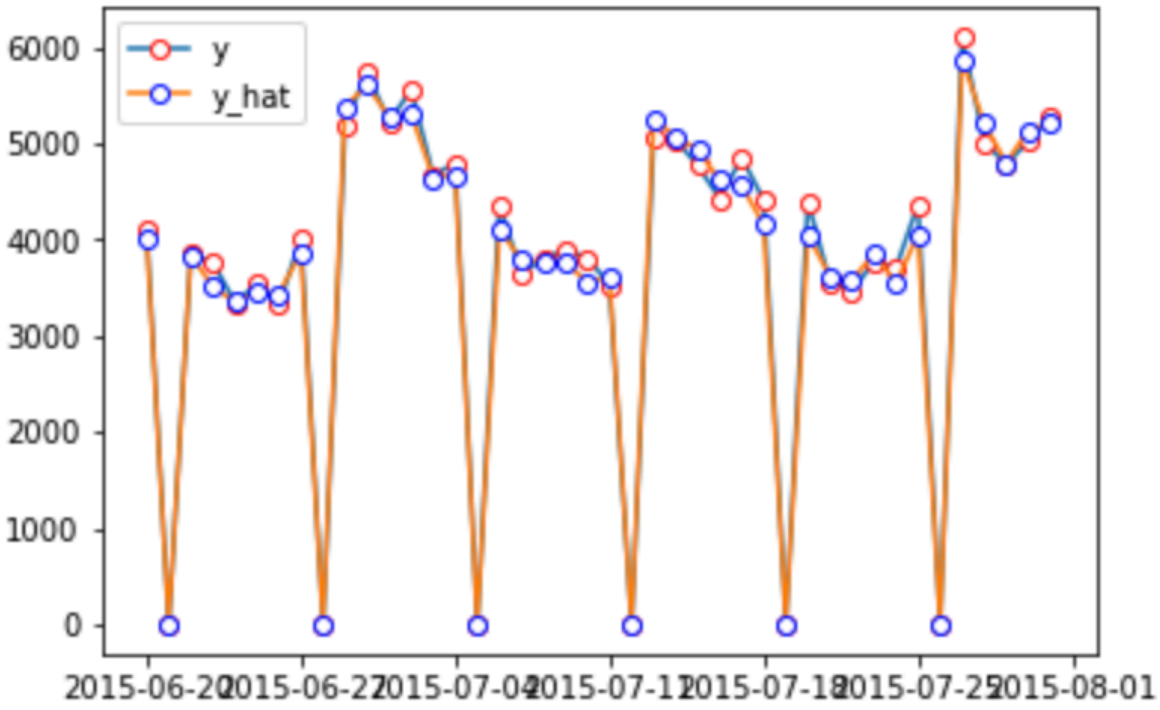
模型的评价与验证

在训练时由于打印出了模型在valid数据集上的评价指标结果，可以看出train数据集上的拟合效果很好。第999次迭代时，在train数据集上的RMSPE结果为0.0672，在valid数据集上的RMSPE的结果为0.0772。为此，可以在test数据集上验证模型的预测效果。

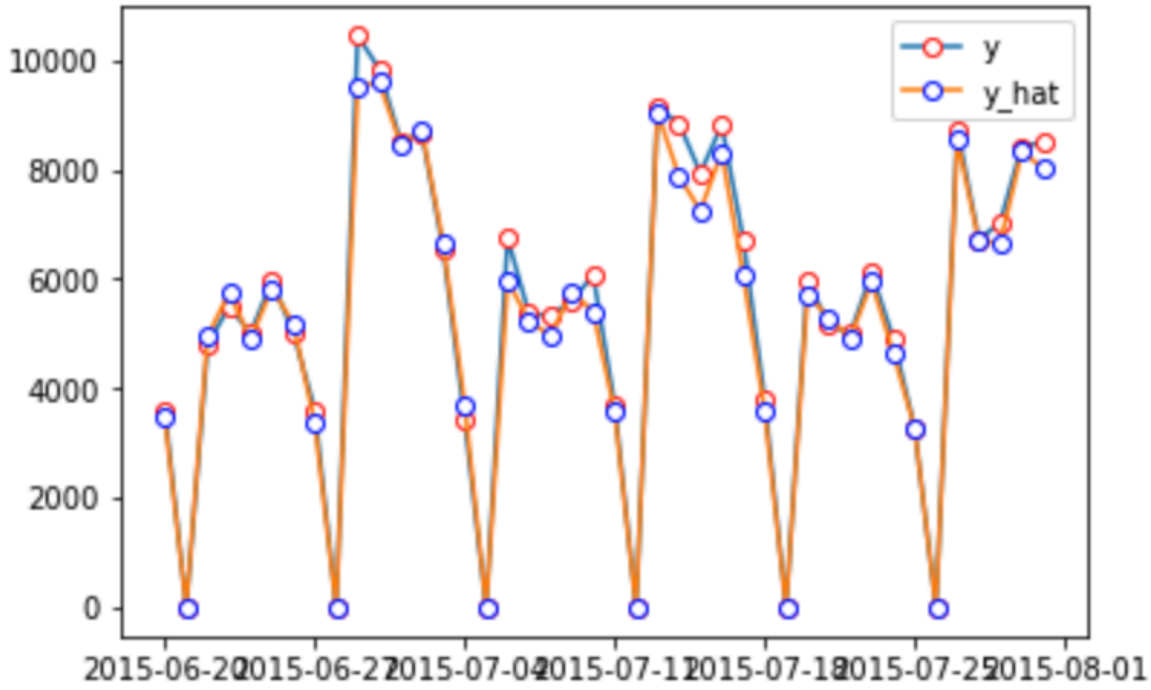
该模型在test数据集上的预测效果为：0.0828。从预测结果上，可以看出，符合逻辑，在测试集上的RMSPE值要高于train和valid上的RMSPE值。

进一步展示模型预测结果与真实值之间的差距，对店铺1和店铺8进行可视化。

图2展示的是店铺1在2015-06-20至2015-07-31销售额的真实值和预测值，其中红色的圈代表真实值，蓝色的圈代表预测值，从图中可以看出，真实值与预测值之间的误差很小。



同样，图3对店铺8的销售额进行了可视化，我们观察店铺8在2015-06-20至2015-07-31销售额的真实值和预测值，其中红色的圈代表真实值，蓝色的圈代表预测值，从图中可以看出，真实值与预测值之间的误差同样很小。



从上述测试集test上的RMSPE和可视化结果，可以看出，该模型达到了预期的效果，并且超过了预期的基准模型的效果。

V. 项目结论

对项目的思考

对Rossmann连锁药妆店销售额的预测是一个典型的机器学习项目，项目的流程很清晰，主要分为3个步骤：问题的定义（明确要解决的问题），数据预处理（包括数据清理，特征工程等），算法选型（包括算法选择，模型训练等）。其中大部分时间花费在数据预处理上面，由于参考了Anton Lebedevich[10]的R代码和Abhilash Awasthi[9]的脚本，节省了很多自己探索的时间。在参考和学习的过程中，也有很多的收获。

- 将时间序列数据处理为横截面数据时，如何利用时间信息处理为特征？可以将`Date`时间处理为`week`，`month`，`year`，`day`这些特征来作为时间特征。
- 当特征的数值比较离散时，可以进行对数处理。
- 对Pandas库的使用更加熟悉了。
- 提升了自己对于xgboost模型的理解和运用。

Reference

1. Chen T, Guestrin C. XGBoost: A Scalable Tree Boosting System[C]// ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2016:785-794. ↩

2. Gert's model doucmentation.↩↩↩

3. https://www.kaggle.com/c/rossmann-store-sales/discussion/18024↩

4. http://mabrek.github.io/↩

5. https://cosx.org/2015/03/xgboost↩

6. http://www.52cs.org/?p=429 ↩↩

7. 统计学习方法 ↩

8. xgboost导读与实战 ↩

9. https://www.kaggle.com/abhilashawasthi/xgb-rossmann?scriptVersionId=94227 ↩↩↩

10. github.com/mabrek/kaggle-rossman-store-sales ↩