Udacity Machine Learning NanoDegree

毕业开题报告

作者：陈俞飞

时间：2019 年 5 月 24 日

一、项目背景

Rossman 是一个在欧洲 7 个国家有 3000 家分店的药品连锁店。现在各个 Rossman 连锁店的经理们正在研究如何提前预测接下来 6 周的日销售额。商店的销售额受到诸如促销，竞争，学校，节假日，区位等多种因素的影响。由于每家分店的经理们都根据各自分店的实际情况进行了销售额预测，导致预测结果的准确性很不一样。

Rossman 的管理者们希望可以通过本次项目得到一些优秀的模型和特征，以便其更准确地对各分店做出销售营业额预测，提高集团整体决策效率。

二、问题描述

根据 Rossman 方面的要求和其提供的数据，我们需要根据过去一段时间内商店的经营情况来对未来一段时间商店的营业额作出预测。

这个问题可以用机器学习-监督学习中的回归方法来进行解决，我们可以训练相关的机器学习回归模型（线性回归，随机森林，xgboost, lightgbm 等）来对商店未来一段时间的营业额作出预测，并进行模型预测结果的评估。

三、数据输入

本次项目中的文件列表如下：

- train.csv：具有 label 的历史销售数据，需要用于训练构建模型
- test.csv：无 label 的历史销售数据，需要用于测试训练的模型
- sample_submission.csv：提交的预测数据的正确格式样本文件
- store.csv： Rossmann 经营的 1115 家商店的信息

特征名称	描述	备注
Store	每家商店的编号	
Sales	表示当天的销售额	
Customers	表示当天的消费者数量	
Open	指示说明商店当天是否营业	0 表示商店关门，1 表示开门营业
StateHoliday	表明当天是否为国家法定节假日	a 表示公众假期;b 表示复活节假期;c 表示圣诞假期;0 表示不是假期
SchoolHoliday	表明当天是否为学校假期	
StoreType	说明商店的类型	a,b,c,d
Assortment	说明商店经营策略的类型	a 表示基础型,b 表示额外型,c 表示扩展型
CompetitionDistance	最近竞争商铺的距离	

CompetitionOpenSicne	竞争者从什么时候开始营业	CompetitionOpenSicneMonth=开始经营月份;CompetitionOpenSicneYear=开始经营年份
Promo	说明给定日期时商店是否有进行促销	
Promo2	说明商店是否有进行连续的促销活动	0 表示商店不参与连续促销;1 表示商店参与连续促销
Promo2Since	描述了开始参与连续性促销的日期	Promo2SinceWeek=参与促销的月份;Promo2SinceYear=参与促销的年份
PromoInterval	描述有连续性促销的间隔	哪些月份有连续性促销

```
train.head(10)
```

	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
5	6	5	2015-07-31	5651	589	1	1	0	1
6	7	5	2015-07-31	15344	1414	1	1	0	1
7	8	5	2015-07-31	8492	833	1	1	0	1
8	9	5	2015-07-31	8565	687	1	1	0	1
9	10	5	2015-07-31	7185	681	1	1	0	1

train 表中共有 9 个特征字段，其中 Sales 应单独作为 y 值与训练集其他特征分离开；Date 字段需要后续进行年份，月份和日期的分离，否则后续模型无法识别。

从特征类型上来看，Sales 和 Customers 属于连续性数值变量，而 StateHoliday，SchoolHoliday,和 DayOfWeek 为无序分类变量，后续需要通过独热编码后才能加入最后的模型。

```
In [8]: store
```

Store	StoreType	Assortment	CompetitionDistance	CompetitionOpenSinceMonth	CompetitionOpenSinceYear	Promo2	Promo2SinceWeek	Promo2SinceYear	PromoInterval
1	c	a	1270.0	9.0	2008.0	0	NaN	NaN	NaN
2	a	a	570.0	11.0	2007.0	1	13.0	2010.0	Jan,Apr,Jul,Oct
3	a	a	14130.0	12.0	2006.0	1	14.0	2011.0	Jan,Apr,Jul,Oct
4	c	c	620.0	9.0	2009.0	0	NaN	NaN	NaN
5	a	a	29910.0	4.0	2015.0	0	NaN	NaN	NaN
6	a	a	310.0	12.0	2013.0	0	NaN	NaN	NaN
7	a	c	24000.0	4.0	2013.0	0	NaN	NaN	NaN
8	a	a	7520.0	10.0	2014.0	0	NaN	NaN	NaN
9	a	c	2030.0	8.0	2000.0	0	NaN	NaN	NaN
10	a	a	3160.0	9.0	2009.0	0	NaN	NaN	NaN
11	a	c	960.0	11.0	2011.0	1	1.0	2012.0	Jan,Apr,Jul,Oct

store 表中共有 10 个特征字段，其中除了因不参与 Promo2 促销而有缺失值的字段外，Competition Distance, CompetitionOpenSinceMonth，CompetitionOpenSinceYear 这三个字段有缺失值。CompetitionDistance 可以用该列特征的最大值填充，CompetitionOpenSinceMonth

和 CompetitionOpenSinceYear 可以用最早的时间进行填充。

从特征类型上来看，StoreType，Assortment,PromotionInterval 为无序分类变量，后续需要通过独热编码后才能加入最后的模型。

四、解决方案：

关于特征工程的选择，我主要准备从两方面入手。

一方面我认为可以根据商店类型，经营策略，有无竞争对手等类别特征构造一些不同类型下的平均营业额和平均顾客数以及平均营业额和平均顾客数偏离该商店所在类型均值的差额。

另一方面我认为根据竞争商店开始的时间和 Promo2 开始的时间构造一些与时间相关的特征，比如商店的营业时间与竞争开始时间的差额，商店的营业时间与 Promo2 开始时间的差额等特征。

模型的选择方面，通过查看 kaggle 该项目页面讨论区的一些讨论，发现 xgboost 模型的表现十分优异，因此我决定采用 xgboost 模型来完成这个项目。

xgboost 是属于 boosting 模型大类 GBDT 模型中的一种。GBDT 和 xgboost 在竞赛和工业界使用都非常频繁，能有效的应用到分类、回归、排序问题。GBDT 是以决策树（CART）为基学习器的 GB 算法，GBDT(Gradient Boosting Decision Tree) 又名 MART (Multiple Additive Regression Tree)，是一种迭代的决策树算法，该算法由多棵决策树组成，所有树的结论累加起来做最终答案。GBDT 的原理是，首先使用训练集和样本真值（即标准答案）训练一棵树，然后使用这棵树预测训练集，得到每个样本的预测值，由于预测值与真值存在偏差，所以二者相减可以得到“残差”。接下来训练第二棵树，此时不再使用真值，而是使用残差作为标准答案。两棵树训练完成后，可以再次得到每个样本的残差，然后进一步训练第三棵树，以此类推。

xgboost 是 Gradient boosting 算法的高效实现，它扩展和改进了 GDBT，该算法更快，准确率也相对高一些。

五、评估标准：

基准模型：

采用 kaggle 竞赛中的前 10%作为本次项目的基准，也就是在该项目 private leaderboard 上的分数要低于 0.11773

评估指标：

根据 Kaggle 上项目发起人的要求，我们选择 RMSPE 作为我们本次项目的评估指标，计算公式如下：

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

其中 y_i 为真实值，y_i-hat 为预测值。

六、项目设计：

1. 数据解释性可视化分析（EDA）

在这一部分，我们对原始数据集进行描述性分析，包括极值的探查和概率分布情况等。

2. 数据预处理

将 train，test 和 store 三张表进行合并，对有关缺失值进行合理填充并剔除相关不合理的训练数据。

3. 特征工程

根据商业常识构建相关新的特征（主要分为店铺相关特征和时间相关特征），对无需分类变量进行独热编码

4. 模型训练

将原始训练集划分为训练和验证两部分，并将最后 7 周的数据作为模型的验证集。对分割出的训练集进行训练并在验证集上进行指标验证。

5. 模型调参

参照 XGBoost 的官方文档进行调参

例如：

max_depth: 树的最大深度，可以用来避免过拟合。该值越大，模型会学到更具体更局部的样本。

learning rate: 通过减少每一步的权重，可以提高模型的鲁棒性

subsample: 这个参数控制对于每棵树，随机采样的比例。减小这个参数的值，算法会更加保守，避免过拟合。

6. 模型评估

用 RMSPE 来对模型进行评估。

7. 结果可视化

对预测结果进行可视化分析，从而更好地了解模型的评估效果。

参考文献：

1. https://www.kaggle.com/c/rossmann-store-sales
2.https://blog.csdn.net/owenfy/article/details/79631144
3.https://zh.wikipedia.org/wiki/XGBoost
4.https://xgboost.readthedocs.io/en/latest/
5. http://www.cnblogs.com/wxquare/p/5541414.html
6. https://www.cnblogs.com/peizhe123/p/5086128.html