预测 Rossmann 未来的销售额

I 问题的定义

一 项目概述

本项目采用 Rossmann 的每天的销售情况的数据，通过对 Rossmann 历史销售数据的学习，预测未来的销售情况。从项目的定义来看，本项目为一个回归问题，属于监督学习，所以要采用监督学习的模型来对数据进行学习及预测。

Rossmann 是欧洲的一家连锁药店，在 7 个欧洲国家拥有 3,000 家药店。目前，Rossmann 店经理需要提前六周预测其日销量。 商店销售受到诸多因素的影响，包括促销，竞争，学校和国家假日，季节性和地点。 成千上万的个人经理根据其独特的情况预测销售量，结果的准确性可能会有很大的变化，可靠的销售额预测可以使得 Rossmann 店经理 制作员工时间表以提高生产力和动力。本项目的目标是帮助 Rossmann 店经理建立一个稳定的预测模型，预测未来的销售额，从而帮助 Rossmann 店经理分析出影响销售的关键要素。

本项目的数据集是从 Kaggle 网站上下载下来的，从数据集来看，分成三个部分：store.csv（店铺信息表），train.csv (训练集) 和 test.csv（测试集）

二 问题陈述

该项目的主要目标是预测 Rossmann 的销售额，是数据挖掘领域的问题，是监督学习的回归问题，从下载的数据看，影响销售额的因素包含几个特征：店铺周围的最近竞争者的距离，店铺的类型，是否在进行促销活动，销售的时间（哪一个月份），是否是国家节假日，是否是学生放假日等等，根据这些特征的值选取适当的机器学习模型对训练集的数据进行学习训练，训练后预测测试集中的销售额，并对测试结果进行评估，此项目主要 采用 Xgboost 模型进行数据的训

练和预测，采用 RMSPE 误差来评估测试效果，预测后预测结果上传到 Kaggle 网站上，得到最后的 RMSPE 得分，得分越小表示数据的预测将越准确，根据得分情况对学习模型的参数进行优化调整，进而得到比较满意的效果。

三 评价指标

该项目采用 RMSPE 数值来作为评估的标准

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，yi 表示记录的 label 值，即某个商店在某一天的销售额，yi_hat 表示对应的预测值

II 分析

一 数据的探索

本项目的数据集是从 Kaggle 网站上下载下来的，从数据集来看，分成三个部分：store.csv（店铺信息表），train.csv (训练集) 和 test.csv（测试集）

训练集数据 train.csv 中包，

其中 store.csv（店铺信息表），该表中有 1115 条记录，记录的是 1115 个店的信息，

其属性详细说明如下：

属性名	说明	备注
store	店 ID	
StoreType	店的类型 a,b,c,d	
Assortment	店的分类	a basic，b:extra,、c: extended

CompetitionDistance	与竞争者的距离	
CompetitionOpenSinceMonth	附近竞争者开始营业的月份	2：表示从 2 月份开始营业
CompetitionOpenSinceYear	附近竞争者开始营业的年份	2008：表示附近竞争者从 2008 年开始营业
Promo2	是否有连续的促销活动	0 没有，1 有
Promo2SinceWeek	店面开始促销活动的周	
Promo2SinceYear	店面开始促销的年	

train.csv（训练集）表里有 101，7210 条数据，包含了从 2013 年开始到 2015 年 7 月的每天的销售额。

其属性详细说明如下

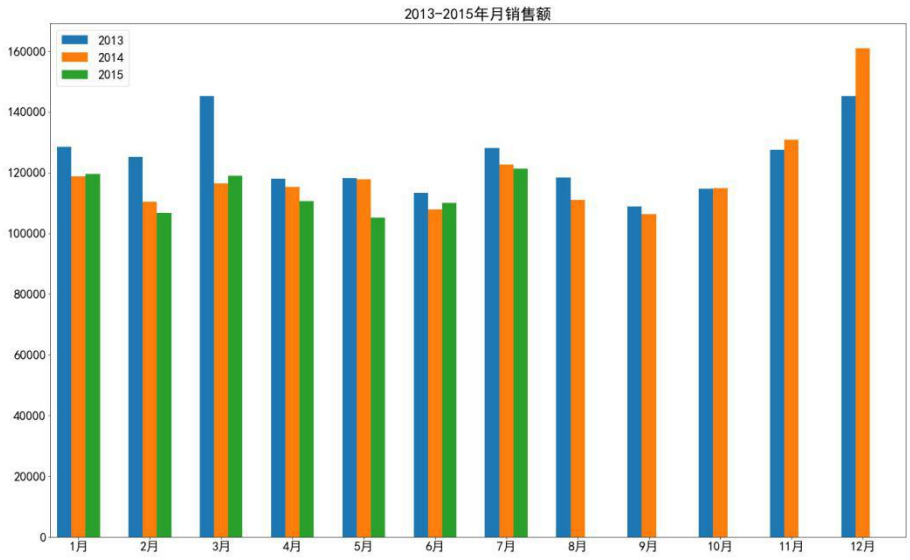
属性名	说明	备注
Store	店 ID	
DayOfWeek	星期几	1-7：星期一~星期日
Date	日期	
Sales	销售额	
Customers	顾客数量	
Open	是否营业	0 否 1 是
Promo	是否有促销活动	0 否 1 是
StateHoliday	国定假日	a = public holiday, b = Easter holiday, c = Christmas, 0 = None

SchoolHoliday	是否为学校假期	0 否 1 是

从上面两个表的数据属性来初步看，影响销售额的因素有很多，所处的日期月份，是否是节假日，学校是否放假，是否有促销活动，所在的日期月份，周围的竞争者等等 都构成影响销售额的因素，但具体的因素的重要性及影响性需要通过数据分析得出最后结论。

二 探索性可视化

从训练集数据来看，筛选 open=1 并且 Sales>0 的数据，作为训练数据，总共 844392 条数据，为了看下 整体销售情况，对每个月的销售额进行了汇总比较，下面的柱形图呈现其对比情况：



从上图中可以看出，每年 5,6,8,9 这几个月的销售额偏低一些，具体因素 还需进一步分析。

测试集 test.csv 总共 41,089 条数据，主要预测 2015 年 8,9 月的数据，其中 open 属性有为空的状态，填充默认值为 1，即为营业状态，测试时候选取 open=1 的数据进行预测，open=0 的时候 设置 Sales = 0。

三 算法和技术

本项目采用 xgboost（eXtreme Gradient Boosting） 学习模型进行训练，是由 GBDT（Gradient Boosting Tree） 发展而来，但从性能上有很大提高：

（1）xgboost 为了防止过拟合，加入了正则项，

（2）xgboost 损失函数是误差部分是二阶泰勒展开，GBDT 是一阶泰勒展开，因此损失函数近似的更精准；

（3）对每颗子树增加一个参数，使得每颗子树的权重降低，防止过拟合，这个参数叫 shrinkage；对特征进行降采样，灵感来源于随机森林，除了能降低计算量外，还能防止过拟合。

（4）对每个特征进行分块（block）并排序，使得在寻找最佳分裂点的时候能够并行化计算。这是 xgboost 比一般 GBDT 更快的一个重要原因。

本项目选用 xgboost 学习模型对数据进行学习训练，评价指标采用 RMSPE 误差，因此在程序实现过程中引入了 xgboost 程序包，首先将训练集数据划分为训练集和验证集，采用了两种划分方式并进行了对比，首先进行了随机划分的方式 采用了 test_size=0.2 的设置进行了划分，可以看出此时划分出来的训练集数据的 Date 是完全打乱的顺序，和原始的 train.csv 数据给出的数据顺序完全不一致，然后对 xgboost 的参数进行设置，主要我进行了几个参数的设置，简单说明如下：

```
num_boost_round = 70
watch_list= [(dtrain, 'train'), (dvalid, 'valid')]
params = {"objective": "reg:linear","booster": "gbtree", "eta": 0.3,"max_depth": 10,"min_child_weight":3} #"min_child_weight":5
print("start train data by xgboost")
xgboost_model = xgb.train(params, dtrain, num_boost_round,evals=watch_list)
print("valid....")
y_pre = xgboost_model.predict(dvalid)
```

参数名称	参数说明
num_boost_round	boosting 的轮数/迭代次数
objective	定义学习任务及相应的学习目标： “reg:linear” – 线性回归
booster	指定使用的 booster,本项目采用 gbtree
eta	学习率，默认是 0.3
max_depth	树的最大深度
min_child_weight	子节点最小的权重。非常容易影响结果，参数数值越大，就越保守，越不会过拟合
evals	设置估计数据,evals 可设置训练集和验证集，在每次迭代后用训练集和验证集代入模型，并给预测结果评分

(Xgboost 模型参数简单说明)

此外还自定义了误差函数 rmspe，代码如下图：

其中，参数有两个 y 和 y_pre, y 是测试集/验证集的真实值，y_pre 是对应的测试集/验证集的预测值

```
from math import sqrt
def rmspe(y,y_pre):
    print("y-y_pre/y",(y-y_pre)/y)
    print("(y-y_pre/y)**2",((y-y_pre)/y)**2)
    a = np.mean(((y-y_pre)/y)**2)
    return sqrt(a)
```

（自定义 rmspe 函数）

四 基准模型

由于该项目是 Kaggle 上的竞赛项目，采用 RMSPE 作为评估的标准，主要采用 Xgboost 模型进行训练和预测，在划分训练集和验证集方式上做了两种尝试，一个是随机划分的方式，一个是手动划分，不打乱数据顺序的情况下

在 eta=0.3 num_boost_round = 20 ,划分的数据数量相同的情况下，Kaggle 得分对比如下：

划分方式	Kaggle 得分 (Private)	Kaggle 得分 (Public)
随机划分（test_size=0.2）	0.23083	0.22617
手动划分([0:813766][813766::])	0.22020	0.20783

从得分上看，手动划分的 RMSPE 值略低些，误差小些，说明数据特征中 date 日期重要性还是比较重要的，从特征重要性的排序上也可以看出此结论。

III 方法

一 数据预处理

1 store 店铺数据进行清洗

```
fill_values = {'CompetitionOpenSinceYear': 0, 'CompetitionDistance': 1, 'CompetitionOpenSinceMonth': 0, 'CompetitionOpenSinceYear': 0,'Promo2Si
data_store.fillna(value=fill_values,inplace = True)
store_drop_columns = ['CompetitionOpenSinceMonth','CompetitionOpenSinceYear','Promo2SinceWeek','Promo2SinceYear','PromoInterval']
```

2 test 数据中 open 特征值进行处理，NaN 的值设置为 1

```
data_test.fillna(value={'Open':1},inplace=True)
data_test.head(10)
```

3 将 所有数据集中 Assortment，StoreType，StateHoliday 等特征值中的字母值转化成为数字类型的值

```
#将字符的属性转换成数字
replace_data = {'a':1,'b':2,'c':3,'d':4}
print(type(data_store))
print(type(data_store['Assortment']))
print(type(data_store.Assortment))
data_store['Assortment'].replace(replace_data,inplace=True)
data_store['StoreType'].replace(replace_data,inplace=True)

data_store.head(10)

data_data['StateHoliday'].replace(replace_data,inplace=True)
data_data['StateHoliday'] = data_data['StateHoliday'].apply(pd.to_numeric)
data_data.shape[0]

data_test['StateHoliday'].replace(replace_data,inplace=True)
data_test['StateHoliday'] = data_test['StateHoliday'].apply(pd.to_numeric)
print(data_test['StateHoliday'].unique())
```

4 归一化处理:将 CompetitionDistance 特征进行归一化处理

```
# 进行归一化 对CompetitionDistance

scaler = MinMaxScaler()
x = data_store['CompetitionDistance'].values.reshape(-1,1)
data_store['CompetitionDistance'] = scaler.fit_transform(x)
data_store.head(5)
```

5 独热编码处理：对 StoreType StateHoliday 等特征进行毒热编码处理，这里做了几个对比，

（1） 在加入 Assortment 进行独热编码处理和不加 Assortment 进行独热编码处理，在 Kaggle 网站的 RMSPE 值比较高,因此去掉了对 Assortment 的独热处理

（2） SchoolHoliday 特征 是否进行独热编码处理,在 Kaggle 网站的 RMSPE 值一样，因此去掉了对 Assortment 的独热处理

（3） DayOfWeek 特征 在进行独热编码处理，在 Kaggle 网站的 RMSPE 值 比较高，因此去掉了对 DayOfWeek 的独热处理

```
data_data = pd.get_dummies(data_data,columns=['StoreType','StateHoliday'])
print(data_data.shape)
print(data_data.columns.values)
print(data_data.head(5))
print("**************")
print(data_test.shape[0])
data_test = pd.get_dummies(data_test,columns=['StoreType','StateHoliday'])
data_test['StateHoliday_2'] = 0
data_test['StateHoliday_3'] = 0
data_data.sort_index(axis=1,inplace=True)
data_test.sort_index(axis=1,inplace=True)
print(data_test.shape)
print(data_test.columns.values)
print(data_test.head(5))
```

二 执行过程

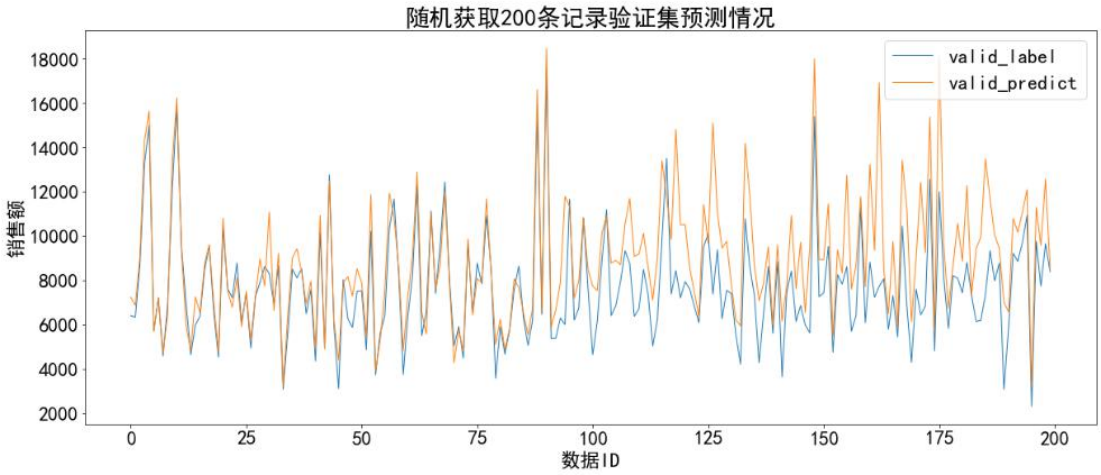
在数据训练、验证及测试过程中，遇到了几个问题并进行了相应处理：

（1） 对于训练集的划分方式进行了两种尝试：A 随机划分 B 手动划分（效果较好）

（2） 在进行独热编码处理后的特征名称可能出现训练集和测试集特征名称排序不一致的情况，进而程序报错，独热编码处理后进行了特征名称的排序处理，进而问题解决。

（3） 对测试集中 open=0 的处理，进行预测的时候将只针对 open=1 的测试数据进行预测，预测结束时候 将 open=0 的预测结果直接设置为 0，再将两部分数据合并，并对其进行排序，使记录顺序和测试集原始顺序完全一致。

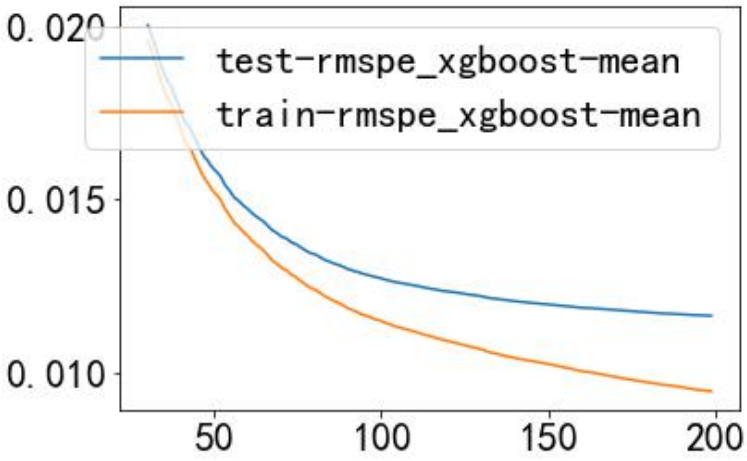
下面是在验证集中随机获取 200 个数据点，将其预测值与真实值对比绘制的折线图



三 完善

进行了几个参数(num_boost_round, eta, min_child_weight)的调优测试，最后的 Kaggle 得分是 0.11265

下面是 num_boost_round=200，eta=0.3，min_child_weight=3 的验证集与训练集的 rmspe 的值的变化曲线图：



下面附上 Kaggle 网站的最近的提交记录：

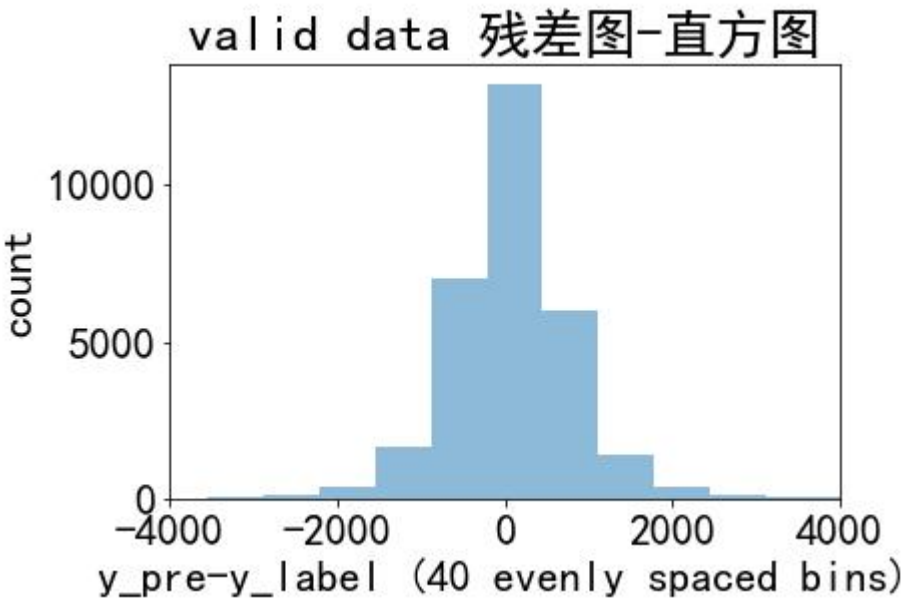
All Successful Selected

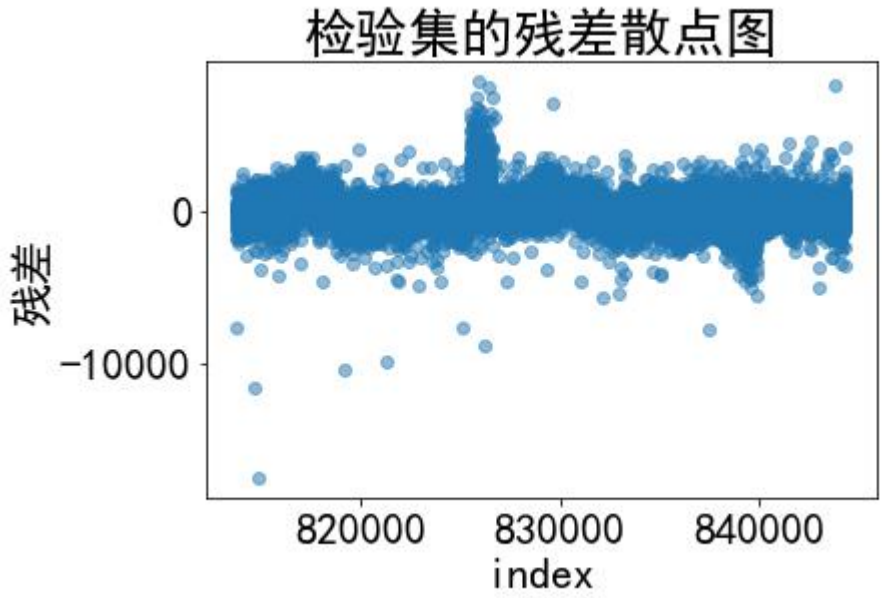
Submission and Description	Private Score	Public Score	Use for Final Score
result_1207_1527.zip a minute ago by hahajing num=200，eta=0.3，w=3	0.13196	0.11265	☐
result_1206_1651.zip a day ago by hahajing num=200,eta=0.3,w=1	0.13468	0.11352	☐
result_1206_1646.zip a day ago by hahajing num=200,eta=0.3,w=3	0.13196	0.11265	☐
result_1206_1630.zip a day ago by hahajing num=100, eta=0.3, w=5	0.13419	0.12120	☐
result_1206_1600.zip a day ago by hahajing num=100 eta=0.3 w=3	0.13601	0.11539	☐

IV 结果

一 模型的评价与验证

该模型从 RMSPE 的值来看是合理的，从特征的重要性排序上来看 和之前的预测基本吻合，下面两张是验证集的残差直方图和散点图。





从上面两张残差图的分析来看，除了个别的异常外，残差的直方图是正态分布的，散点图上基本是在 0 上下密集分布的，所以该 XGBOOST 模型在该数据集的表现是良好的。

二 合理性分析

通过调优参数测试，RMSPE 值由最初的 0.22617 下降到最终的 0.11265，从准确度来看比最初的模型要好

V 项目结论

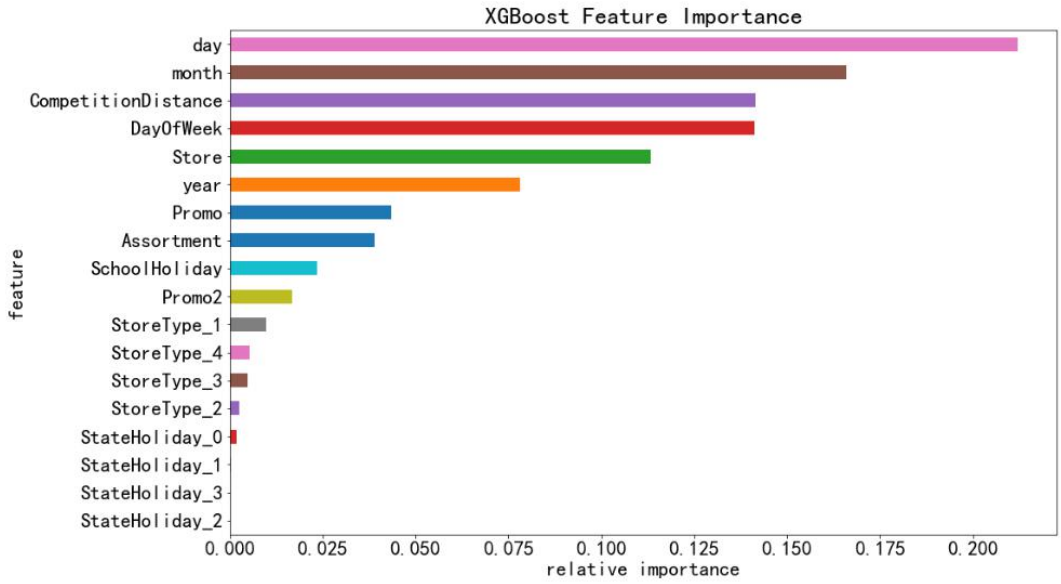
一 结果可视化

下面是在 kaggle 网站的得分情况

All Successful Selected

Submission and Description	Private Score	Public Score	Use for Final Score
result_1207_1527.zip a minute ago by hahajing num=200 , eta=0.3 , w=3	0.13196	0.11265	☐
result_1206_1651.zip a day ago by hahajing num=200,eta=0.3,w=1	0.13468	0.11352	☐
result_1206_1646.zip a day ago by hahajing num=200,eta=0.3,w=3	0.13196	0.11265	☐
result_1206_1630.zip a day ago by hahajing num=100, eta=0.3, w=5	0.13419	0.12120	☐
result_1206_1600.zip a day ago by hahajing num=100 eta=0.3 w=3	0.13601	0.11539	☐

手动划分训练集得出的特征重要性排序柱状图



二 对项目的思考

该项目是典型的监督学习的项目，通过这个项目的整个制作，进一步熟悉和理解了监督式学习项目的处理方法和流程，对数据的预处理有了更细致的了解，尤其是热度编码的处理，通过对不同特征的是否进行热度编码处理的对比，更深刻理解了热度编码适用的场景，重要的是学习了 Xgboost 模型的数据训练和预

测的过程，由于时间关系还没来得及对其他模型进行对比，需要进一步对数据的理解和对数据模型选取的学习和训练。

三 需要做出的改进

该项目从 Kaggle 预测得分上来看是基本达标，但是还需要改进的是如何使得模型更稳健，更通用，是否适合任何日期的销售额的预测，或者说这个训练集能满足多久的销售额的预测，这个有待考量。

文献引用

[1] https://www.cnblogs.com/timxgb/p/8231130.html

[2] https://blog.csdn.net/u010414589/article/details/51153310

[3] https://blog.csdn.net/han_xiaoyang/article/details/52665396

[4] https://juejin.im/entry/5a1fea03f265da432153d3d5

[5] https://blog.csdn.net/aliceyangxi1987/article/details/73598857

[6] https://www.jianshu.com/p/8346d4f80ab0

[7] https://zhuanlan.zhihu.com/p/31182879

[8] https://ask.hellobi.com/blog/lsxxx2011/10242

[9] https://www.kesci.com/home/competition/forum/56fd68bdef4b99e9098371d4

[10] https://blog.csdn.net/zhangf666/article/details/70174464

[11] http://wepon.me/files/gbdt.pdf

[12] https://www.cnblogs.com/willnote/p/6801496.html