

# 基于 XGBosst 的销售预测

2020 年 1 月 5 日

---

# 目 录

<b>1</b>	<b>项目概述</b>	<b>4</b>
1.1	背景信息	4
1.2	问题陈述	4
1.3	解决方案与评价指标	4
<b>2</b>	<b>分析</b>	<b>5</b>
2.1	数据探索	5
2.2	可视化分析	8
2.3	算法和技术	11
2.4	基准模型	12
<b>3</b>	<b>方法</b>	<b>13</b>
3.1	数据预处理	13
3.2	执行过程	14
3.3	完善	16
<b>4</b>	<b>结果</b>	<b>17</b>
4.1	模型评价与验证	17
<b>5</b>	<b>项目结论</b>	<b>17</b>
5.1	思考	19
5.2	待改进点	20
<b>参考文献</b>		错误!未定义书签。
<b>致 谢</b>		<b>22</b>



# 1 项目概述

## 1.1 背景信息

Rossmann 在 7 个欧洲国家中有超过 3000 家药店。该项目的主要目的是利用这些药店过去几年的销售数据，去预测未来 6 周各药店的销量。

商店的销量受非常多的因素影响，包括促销信息、竞争对手信息、开学信息和国家节日（包括季节性的节日和当地的节日），以往的销量预测依靠各商店主管对于周期的判断，预测的准确度也因人而异。

更加精准的预测有助于商店制定更加合理的制定备货与促销策略，增大利润的同时，更好的迎合消费者的需求，提升购买体验。

## 1.2 问题陈述

本项目主要是利用 1115 家 Rossmanns 商店的历史销售数据，去预测未来 6 周每个商店的销量，历史销售数据属于时序数据，总体上看该问题属于典型的监督学习回归问题。

主要涉及的数据主要分为两个部分：

1. 历史销售数据：为每个商店按日期的时序型数据，包括 1115 家门店的编号、日期、销售额、访问客户数、营业状态、国家节日类型、校园日、促销状态信息；
2. 商店数据：包括商店编号、商店类型、竞争对手信息、促销月份、促销间隔信息。

## 1.3 解决方案与评价指标

本项目参考了 Kaggle 上若干经典解决方案后，拟采用 XGboost 模型进行回归预测，评价指标选用 Kaggle 针对该项目的指定指标 RMSPE

RMSPE 属于 MSE 的衍生指标，MSE 全称为均方根误差，属于回归类分析的常见评价指标，其公式为（ $y_{true}$  表示真实值， $y_{pred}$  表示预测值）：

$$MSE = \frac{1}{N} \sum_{i=1}^N |y_{true} - y_{pred}|^2$$

RMSPE 的公式为：

$$RMSPE = \sqrt{\frac{1}{N} \sum_{i=1}^N \left| \frac{y_{true} - y_{pred}}{y_{true}} \right|^2}$$

## 2 分析

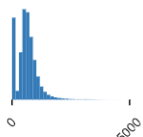


### 2.1 数据探索

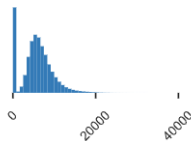
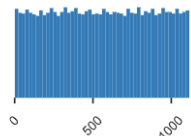
利用 Pandas\_Profiling 对销售数据集与商店数据集进行数据探索分析。

本项目的销售数据共计 1,017,209 行，共计 9 个特征，数据集前 5 行如下图所示：

First rows									
	Customers	Date	DayOfWeek	Open	Promo	Sales	SchoolHoliday	StateHoliday	Store
0	555	2015-07-31	5	1	1	5263	1	0	1
1	625	2015-07-31	5	1	1	6064	1	0	2
2	821	2015-07-31	5	1	1	8314	1	0	3
3	1498	2015-07-31	5	1	1	13995	1	0	4
4	559	2015-07-31	5	1	1	4822	1	0	5
5	589	2015-07-31	5	1	1	5651	1	0	6
6	1414	2015-07-31	5	1	1	15344	1	0	7
7	833	2015-07-31	5	1	1	8492	1	0	8
8	687	2015-07-31	5	1	1	8565	1	0	9
9	681	2015-07-31	5	1	1	7185	1	0	10

其中各特征类型如下：

Customers Numeric		Distinct count	4086	Mean	633.1459464	
		Unique (%)	0.4%	Minimum	0	
		Missing (%)	0.0%	Maximum	7388	
		Missing (n)	0	Zeros (%)	17.0%	
		Infinite (%)	0.0%			
		Infinite (n)	0			<a href="#">Toggle details</a>
Date Categorical		Distinct count	942		2015-07-30 1115	
		Unique (%)	0.1%		2013-06-27 1115	
		Missing (%)	0.0%		2015-07-03 1115	
		Missing (n)	0		Other values (939)	
					1013864	
						<a href="#">Toggle details</a>
DayOfWeek Numeric		Distinct count	7	Mean	3.998340557	
		Unique (%)	< 0.1%	Minimum	1	
		Missing (%)	0.0%	Maximum	7	
		Missing (n)	0	Zeros (%)	0.0%	
		Infinite (%)	0.0%			

Open Boolean	Distinct count Unique (%) Missing (%) Missing (n)	2 < 0.1% 0.0% 0	1 844392 0 172		<a href="#">Toggle details</a>
Promo Boolean	Distinct count Unique (%) Missing (%) Missing (n)	2 < 0.1% 0.0% 0	0 629129 1 388080		<a href="#">Toggle details</a>
Sales Numeric	Distinct count Unique (%) Missing (%) Missing (n) Infinite (%) Infinite (n)	21734 2.1% 0.0% 0 0.0% 0	Mean 5773.818972 Minimum 0 Maximum 41551 Zeros (%) 17.0%		<a href="#">Toggle details</a>
SchoolHoliday Boolean	Distinct count Unique (%) Missing (%) Missing (n)	2 < 0.1% 0.0% 0	0 835488 1 181		<a href="#">Toggle details</a>
StateHoliday Categorical	Distinct count Unique (%) Missing (%) Missing (n)	5 < 0.1% 0.0% 0	0 855087 0 131072 a 20260 Other values (2) 10790		<a href="#">Toggle details</a>
Store Numeric	Distinct count Unique (%) Missing (%) Missing (n) Infinite (%) Infinite (n)	1115 0.1% 0.0% 0 0.0% 0	Mean 558.4297268 Minimum 1 Maximum 1115 Zeros (%) 0.0%		<a href="#">Toggle details</a>

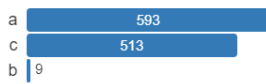
综上所述，销售数据集中共有 4 个数值型特征、2 个类别型特征和 3 个布尔型特征。缺失值方面，仅有 Customers 与 Sales 两个特征有 0 值存在，且个数相等，进一步分析发现缺失原因为该商店当日为非营业状态，后期在数据预处理阶段进行 0 值补缺。

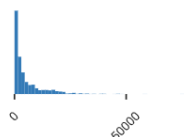
本项目的商店数据共计 1,115 行，共计 10 个特征，数据集前 5 行如下图所示：

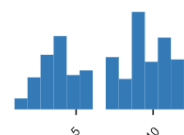
## First rows

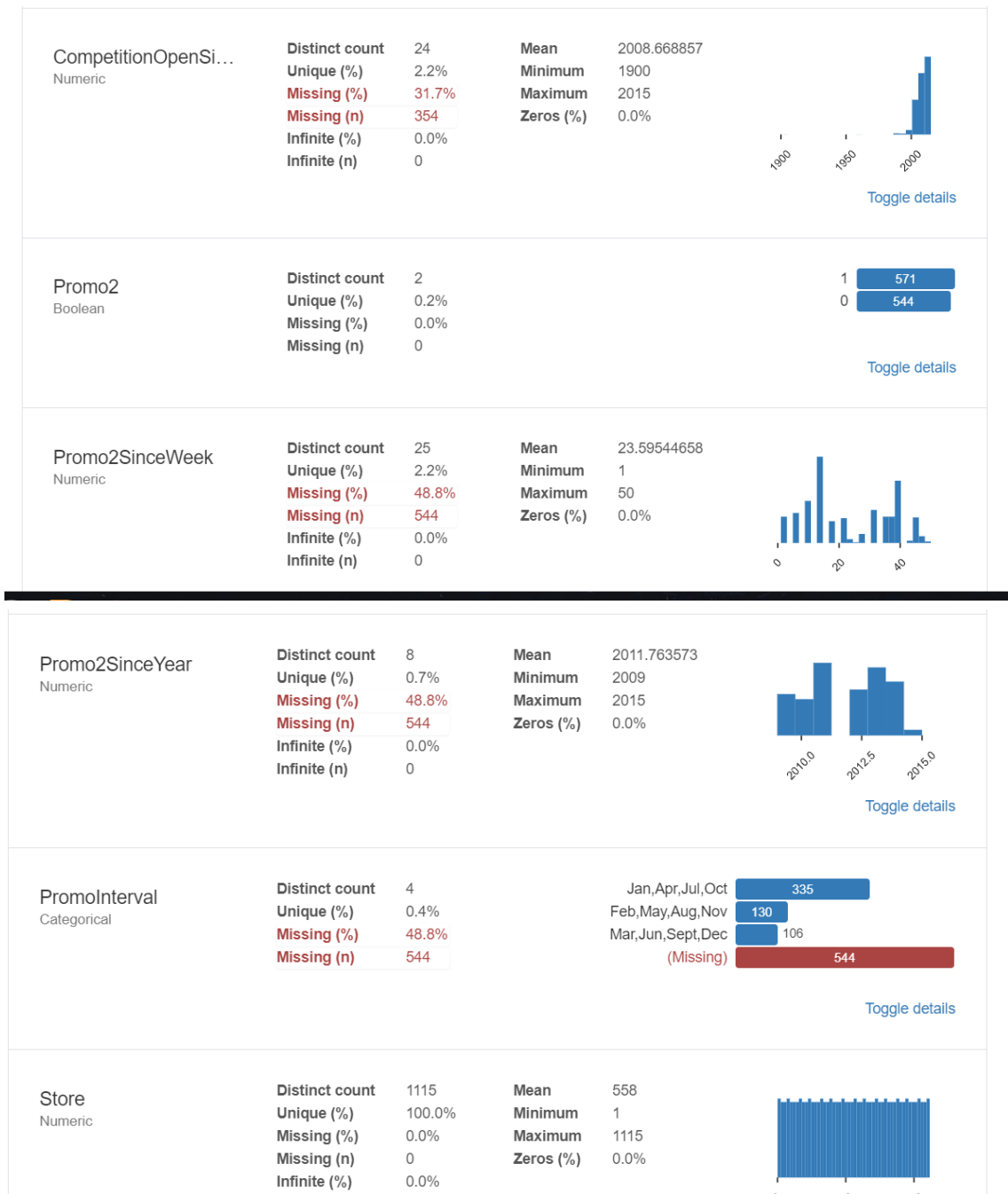
	Assortment	CompetitionDistance	CompetitionOpenSinceMonth	CompetitionOpenSinceYear	Promo2	Promo2SinceW
0	a	1270.0	9.0	2008.0	0	NaN
1	a	570.0	11.0	2007.0	1	13.0
2	a	14130.0	12.0	2006.0	1	14.0
3	c	620.0	9.0	2009.0	0	NaN
4	a	29910.0	4.0	2015.0	0	NaN
5	a	310.0	12.0	2013.0	0	NaN
6	c	24000.0	4.0	2013.0	0	NaN
7	a	7520.0	10.0	2014.0	0	NaN
8	c	2030.0	8.0	2000.0	0	NaN
9	a	3160.0	9.0	2009.0	0	NaN

其中各特征类型如下：

Assortment Categorical	Distinct count	3			
	Unique (%)	0.3%			
	Missing (%)	0.0%			
	Missing (n)	0			
<a href="#">Toggle details</a>					

CompetitionDistance Numeric	Distinct count	655	Mean	5404.901079	
	Unique (%)	58.7%	Minimum	20	
	Missing (%)	0.3%	Maximum	75860	
	Missing (n)	3	Zeros (%)	0.0%	
	Infinite (%)	0.0%			
	Infinite (n)	0			
<a href="#">Toggle details</a>					

CompetitionOpenSi... Numeric	Distinct count	13	Mean	7.224704336	
	Unique (%)	1.2%	Minimum	1	
	Missing (%)	31.7%	Maximum	12	
	Missing (n)	354	Zeros (%)	0.0%	
	Infinite (%)	0.0%			
	Infinite (n)	0			

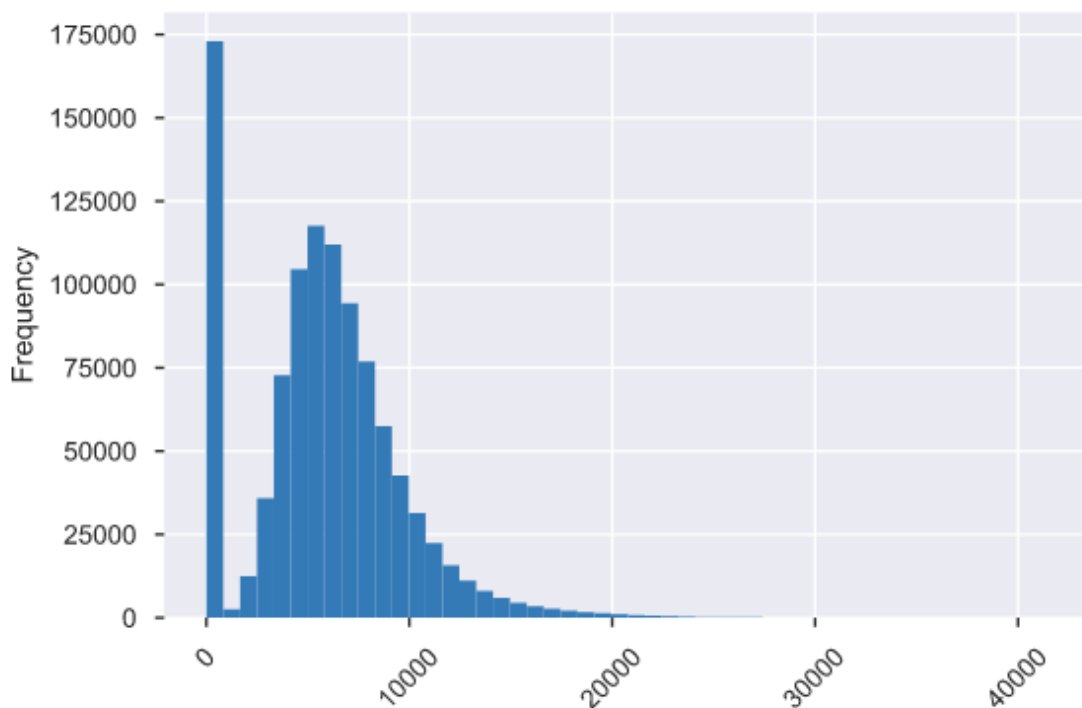


综上所述，商店数据集中共有 6 个数值型特征、3 个类别型特征和 1 个布尔型特征。缺失值方面，CompetitionOpenSinceMonth、CompetitionOpenSinceYear、Promo2SinceWeek、Promo2SinceYear、PromoInterval 含有一定比例的缺失值，主要原因为部分商店周边暂未有竞争对手或者未参与过促销活动，后期将对这些特征进行映射补零操作

## 2.2 可视化分析

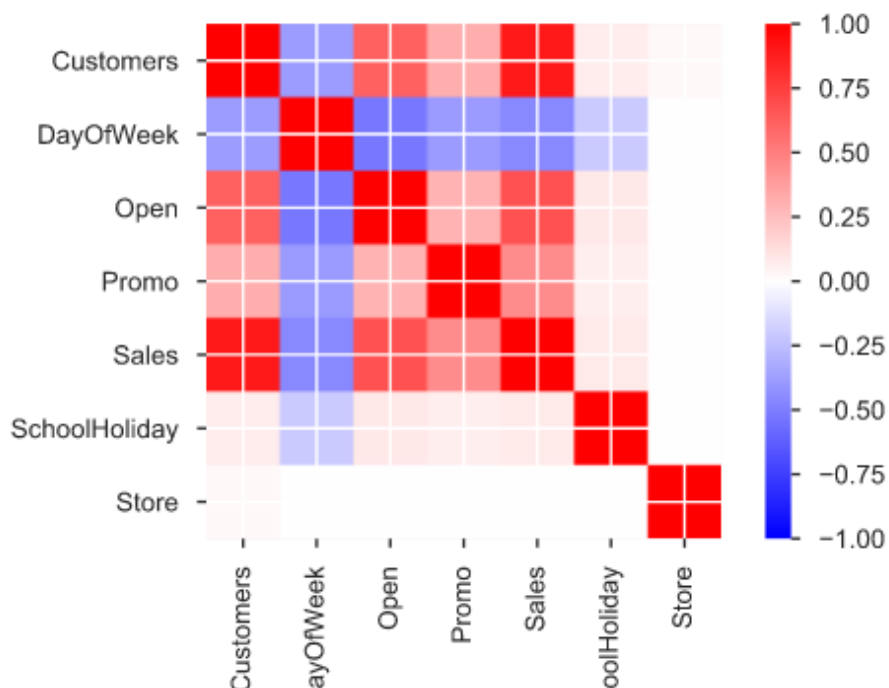


整体销量额偏态分布：



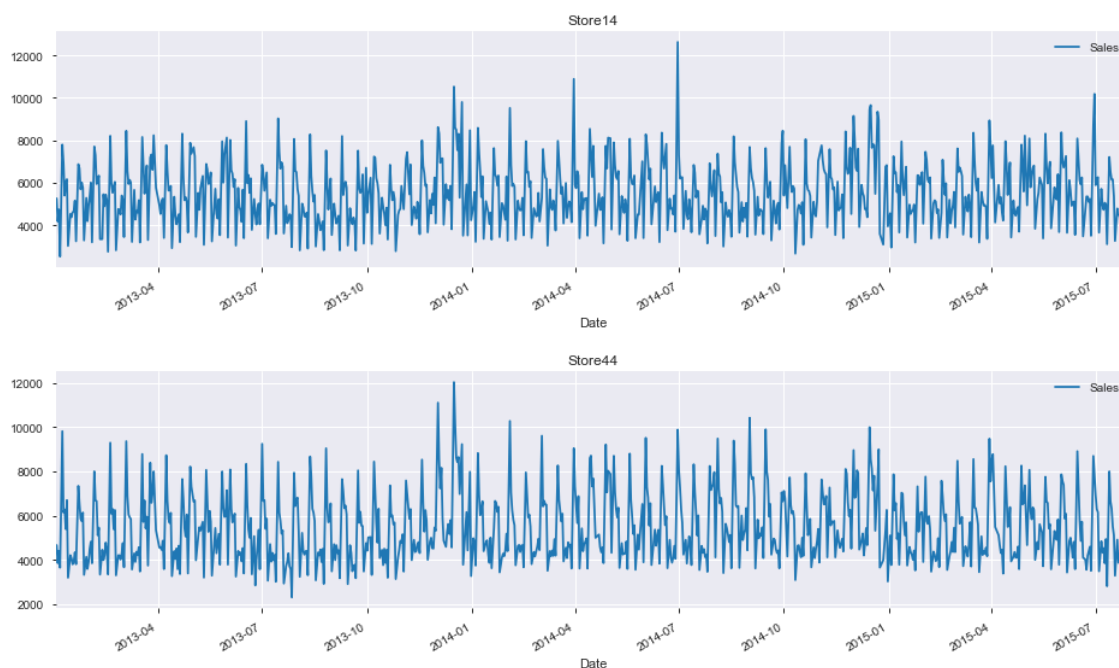
查看所有销售额的分布，会发现销售额成偏态分布，且有大量的 0 值存在，0 值记录应整体剔除。同时剔除后，应使用  $\log$  函数将销售额平滑成正态分布。

销售类数据集的特征间关联性：



对销售类数据集的绘制关联热图（如上图），颜色深浅表示的为各特征间 Pearson 关联系数的大小，红色表示正向关联，蓝色表示负向关联。由上图可见，除客户数量、营业状态外，销量还与促销状态成正相关。因此，在后期特征抽取时，应重点抽取促销类信息。

### 商店销售的周期性：



分别选取商店编号为 14、44 号商店绘制销量图，发现上述两家商店销售额呈现一定的周期性波动。



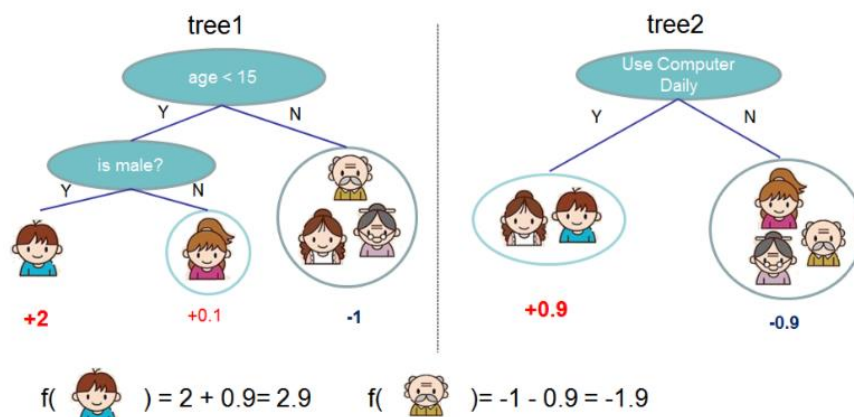


更进一步，查看选中 14 号商店的 2014 年 6-7 月销量和 7-9 月销量，发现销量波动呈现一定相似性。从波动规律可以看出，由于销量周期性的存在，短期销量预测具有可行性，同时在特征选取时，应保留时间特征，确保数据集中有特征可以体现时序性。

## 2.3 算法和技术

本项目将采用 XGboost 模型进行回归预测。XGboost 本质属于 Boosting 分类器模型，属于集成学习模型，它基本思想是把成百上千个分类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。这个模型会不断地迭代，每次迭代就生成一颗新的树。一般来说，gradient boosting 的实现是比较慢的，因为每次都要先构造出一个树并添加到整个模型序列中，而 XGBoost 的特点就是计算速度快，模型表现好[1]。

XGBoost 与 GBDT 一样，也是 boosting decision tree 模型。与 GBDT 一样，其最基本的组成部分为 CART 树，其根据训练特征及训练数据构建分类树，判定每条数据的预测结果。其中构建树使用 gini 指数计算增益，即进行构建树的特征选取。而一个 CART 树往往过于简单，一般不能做出有效预测，因此 boosting decision tree 模型均会采用多棵树进行组合预测，输入样本的最终预测结果等于各个树预测值之和，具体可见下图：



而与 GDBT 模型最大的不同之处在于的 XGboost 目标函数,其目标函数如下:

$$L(\theta) = \sum_i l(\hat{y}_i, y_i) + \sum_k \Omega(f_k)$$

目标函数分为两部分,第一个部分即是损失函数,第二个部分为正则化项。与 GDBT 不同之处在于 XGboost 的正则化项更加精细化[2]:

$$\Omega(\theta) = \gamma T + \frac{1}{2} \lambda \sum_{j=1}^T \omega_j^2$$

在 XGboost 里面,正则项又包含两部分:一部分是提升树叶子节点的数量  $T$ ,用于控制树的复杂度,可以达到剪枝的效果;另一部分为每颗树的叶子节点的权重  $w$  的平方和。正则项可以避免 over fitting。

XGboost 对目标函数  $l$  进行了泰勒展开,做一个近似,最终的目标函数  $l$  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,泰勒展开如下:

$$L^{(t)} \simeq \sum_i^n \left[ l(y_i, \hat{y}_i^{(t-1)}) + g_i f_t(X_i) + h_i f_t^2(X_i) \right] + \Omega(f_t)$$

本项目还将利用 GPU 加速 XGboost 的训练过程,完成参数的网格搜索,达到模型参数的最优化。

## 2.4 基准模型

本项目以 XGboost 初始参数的模型为基准模型[3],测得基准 RMSPE 得分 0.132837。同时,项目目标为结业最低要求,即测试集的最低 RMSPE 为 0.11773

```

In [27]: #基础模型
train_model(params, True)

[915] train-rmse:0.066823 eval-rmse:0.122004 train-rmspe:0.070816 eval-rmspe:0.132829
[916] train-rmse:0.066796 eval-rmse:0.122006 train-rmspe:0.070785 eval-rmspe:0.132832
[917] train-rmse:0.066781 eval-rmse:0.121995 train-rmspe:0.070777 eval-rmspe:0.132811
[918] train-rmse:0.066768 eval-rmse:0.121976 train-rmspe:0.070758 eval-rmspe:0.132786
[919] train-rmse:0.066751 eval-rmse:0.121986 train-rmspe:0.070739 eval-rmspe:0.132803
[920] train-rmse:0.066724 eval-rmse:0.122003 train-rmspe:0.070704 eval-rmspe:0.132828
[921] train-rmse:0.066698 eval-rmse:0.122014 train-rmspe:0.070668 eval-rmspe:0.132835
[922] train-rmse:0.066683 eval-rmse:0.122014 train-rmspe:0.070649 eval-rmspe:0.132835
[923] train-rmse:0.066671 eval-rmse:0.122012 train-rmspe:0.070637 eval-rmspe:0.132833
[924] train-rmse:0.066654 eval-rmse:0.122015 train-rmspe:0.070615 eval-rmspe:0.13284
[925] train-rmse:0.06664 eval-rmse:0.122015 train-rmspe:0.070601 eval-rmspe:0.132841
[926] train-rmse:0.066625 eval-rmse:0.12201 train-rmspe:0.070582 eval-rmspe:0.132837
Stopping. Best iteration:
[826] train-rmse:0.06864 eval-rmse:0.122022 train-rmspe:0.073117 eval-rmspe:0.132777

time is 144.319923 s.
testing...
RMSPE: 0.132837

Out[27]: (0.13283742588634764, <xgboost.core.Booster at 0x26105445648>)

```

## 3 方法

### 3.1 数据预处理

**Null 值处理。**商店数据的 Null 值主要集中于竞争信息相关与促销信息相关的特征, Null 值的主要原因为该商店附近未有竞争对手或未进行促销, 因此针对该部分 Null 的填充方法为使用 0 值填充。另外测试集中, 也存在 Null, 主要为营业状态, 对于该部分 Null 使用 1 值进行填充, 默认商店处于营业状态。

**日期格式转换。**销售数据集中, 存在日期格式的特征 Date。由可视化分析中可知, 日期特征是重要的时序型特征, 也是表达周期性的关键特征, 但由于日期型无法直接用于模型计算, 故将日期拆分为年、月、日三个整数型特征。

**销售数据集与商店数据集合并。**由于销售数据集与商店数据集都是模型训练的关键数据, 因此利用 pandas 的 merge 方法, 将两个数据集合并为一个数据集用于模型训练。

**拆分训练集与验证集。**将合并后的数据集进一步拆分为训练集与验证集, 由于本项目的目标是预测未来 6 周的销售额, 故将合并后的数据集中, 最后 6 周的 1115 家商店的数据拆分出来作为验证集, 其余数据作为训练集。同时, 由于当商店处于未营业状态时, 销售额为 0, 这些数据在训练或者验证集中都对模型训练具有一定的干扰, 故从上述数据集中剔除。

**字符型(分类型)特征进行映射转换。**针对 StateHoliday、Assortment 与 StoreType 三个字符型特征, 由于数据探索是发现这三类特征的值属于离散型, 且值不超过 5 种类型, 故针对 5 种取值进行映射转换, 分别映射为 0-5 的整数。

**其他特征变换。**针对特征竞争对手营业时间, 将其变换为竞争对手至今营业天数, 促销开始时间转换为促销存续时间, 同时针对促销月份信息, 生成新特征“是否处于促销月份”, 最大程度保留促销信息[4]。

### 3.3 所使用的特征

1. Store: 商店编号
2. DayofWeek: 日期处于该周的第几天
3. Promo: 是否促销标识
4. StateHoliday: 州节假日类型
5. StoreType: 商店类型 1
6. Assortment: 商店类型
7. CompetitionDistance: 竞争对手距离
8. CompetitionOpenSinceMonth: 竞争开始月份
9. CompetitionOpenSinceYear: 竞争开始年份
10. Promo2: 是否促销标识 2
11. Promo2SinceWeek: 促销 2 开始周
12. Promo2SinceYear: 促销 2 开始年份
13. Year: 交易年份
14. Month: 交易月份
15. Day: 交易日
16. WeekofYear: 交易日期为本年的第几周
17. CompetitionOpen: 交易发生时, 对手是否已开始营业
18. PromoOpen: 促销已开始天数
19. IsPromoMonth: 是否处于促销月份

### 3.4 执行过程

1. 使用 Google 云环境配置 GPU 服务器
2. 编写相关函数, 包括网格搜索函数与 RMPSE 函数
3. 设置模型初始参数值
4. 设定网格搜索参数, 循环调试得出最佳参数:
 

```
{'objective': 'reg:squarederror', 'learning_rate': 0.1, 'booster': 'gbtree',
  'max_depth': 10, 'min_child_weight': 2, 'subsample': 0.85, 'colsample_bytree':
  0.85, 'gamma': 0.0, 'silent': 1, 'seed': 10, 'tree_method': 'gpu_hist', 'gpu_id': 0,
  'reg_alpha': 1e-05, 'reg_lambda': 100}
```
5. 使用最佳参数训练模型

```

In [22]: train_model(params, True)

[1874] train-rmse:0.070228 eval-rmse:0.116249 train-rmspe:0.088187 eval-rmspe:0.12532
[1875] train-rmse:0.070223 eval-rmse:0.116248 train-rmspe:0.088182 eval-rmspe:0.125316
[1876] train-rmse:0.070213 eval-rmse:0.116244 train-rmspe:0.088116 eval-rmspe:0.125313
[1877] train-rmse:0.070204 eval-rmse:0.116243 train-rmspe:0.088109 eval-rmspe:0.125311
[1878] train-rmse:0.070197 eval-rmse:0.116239 train-rmspe:0.088118 eval-rmspe:0.125304
[1879] train-rmse:0.070192 eval-rmse:0.116246 train-rmspe:0.088112 eval-rmspe:0.125311
[1880] train-rmse:0.070183 eval-rmse:0.116245 train-rmspe:0.088043 eval-rmspe:0.125314
[1881] train-rmse:0.070173 eval-rmse:0.116243 train-rmspe:0.087989 eval-rmspe:0.125311
[1882] train-rmse:0.070168 eval-rmse:0.116242 train-rmspe:0.087984 eval-rmspe:0.125311
[1883] train-rmse:0.070161 eval-rmse:0.116241 train-rmspe:0.087973 eval-rmspe:0.125309
[1884] train-rmse:0.070155 eval-rmse:0.116241 train-rmspe:0.087959 eval-rmspe:0.125313
[1885] train-rmse:0.070145 eval-rmse:0.116242 train-rmspe:0.08795 eval-rmspe:0.125317
Stopping. Best iteration:
[1785] train-rmse:0.070954 eval-rmse:0.116201 train-rmspe:0.090557 eval-rmspe:0.125263

time is 286.852863 s.
testing...
RMSPE: 0.125317

Out[22]: (0.12531663578742377, <kgboost.core.Booster at 0x279ebe8e608>)

```

如上图所示，模型收敛轮次为 1785，train-rmspe 为 0.090557，val-rmspe 为 0.125263

## 6. 将训练后的模型在验证集进行验证

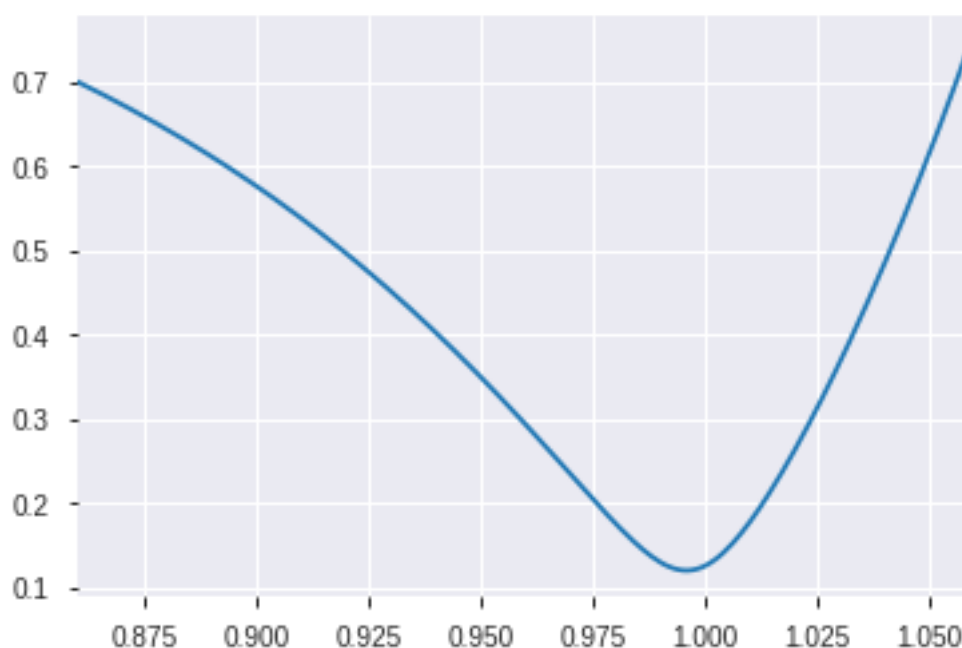
Rossmann_submission_1.csv	0.12732	0.11948	<input type="checkbox"/>
2 days ago by Taylor Tai			
<a href="#">add submission details</a>			

验证结果中，Private Score 为 0.12732，Public Score 为 0.11948

## 7. 将模型预测测试集，得到待提交数据

## 8. 根据验证集的预测结果，计算整体偏差，并根据修正参数对待提交数据修正分布

- 这里分布的调整的原因因为训练集与测试集非严格的独立同分布，在该项目中，在基本数据情况相似的情况下销售额分布会略有不同[5]，可能是受到经济周期的影响，类似的浮动和我们选取的数据集没有太大关系，为提升预测准确性，此处采用简单的做法，即将预测出的数值乘以一个权重数值将整体预测数值进行调整。
- 在验证集上添加模型预测的结果，设定权重数从 0.860 增加 1.06（步长为 0.001），每次权重数更新后，将预测结果乘以权重数，重新计算预测值，并计算此时验证集的 RMSPE，图标如下：



发现当权重数达到 0.996 时, RMSPE 能达到最低。

按照上述步骤, 按商店维度。每个商店数据执行上述操作, 并计算每个商店销量分布的调整权重数, 最后将计算得出的权重数应用到测试集中

9. 提交数据, 得到结果分数, Public Score 为 0.11431, Private Score 为 0.10887

Rossmann_submission_3.csv	0.11431	0.10887	<input type="checkbox"/>
2 days ago by Taylor Tai			
final try			

### 3.5 完善

1. 初始参数得到的模型, 其在验证集的 RMPSE 得分为 0.13457
2. 后续分别对 max\_depth、max\_depth、gamma、subsample 与 colsample\_bytree、reg\_alpha、reg\_lambda、learning\_rate 参数进行了网格搜索, 最终确定最佳参数[6]
3. 第一次网格搜索训练时, 使用 CPU 进行训练速度较慢, 后经查阅资料发现 XGboost 支持 GPU, 故配置 GPU 环境加速训练过程。整体网格搜索过程大约为 2 小时
4. 确定最佳参数后并训练模型, 得到第一版提交结果, 提交 Kaggle 后发现 RMPSE 得分为 0.11948, 与最低要求相差 0.002 分
5. 进一步查阅相关资料, 应用验证集数据进行偏差修正, 再次提交后达到分数要求

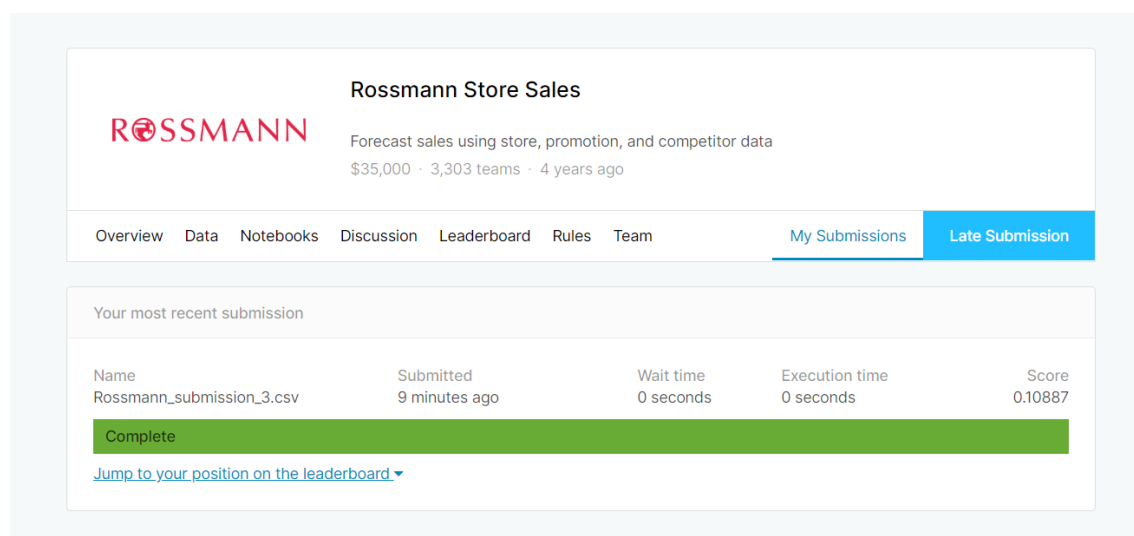


## 4 结果

### 4.1 模型评价与验证

最终模型各参数经网格搜索后，设置较为合理，除 `gamma` 值外，所有模型初始参数均得到优化，最终模型的 RMSPE 得分明显高于基准模型，同时解决了项目设定的目标。

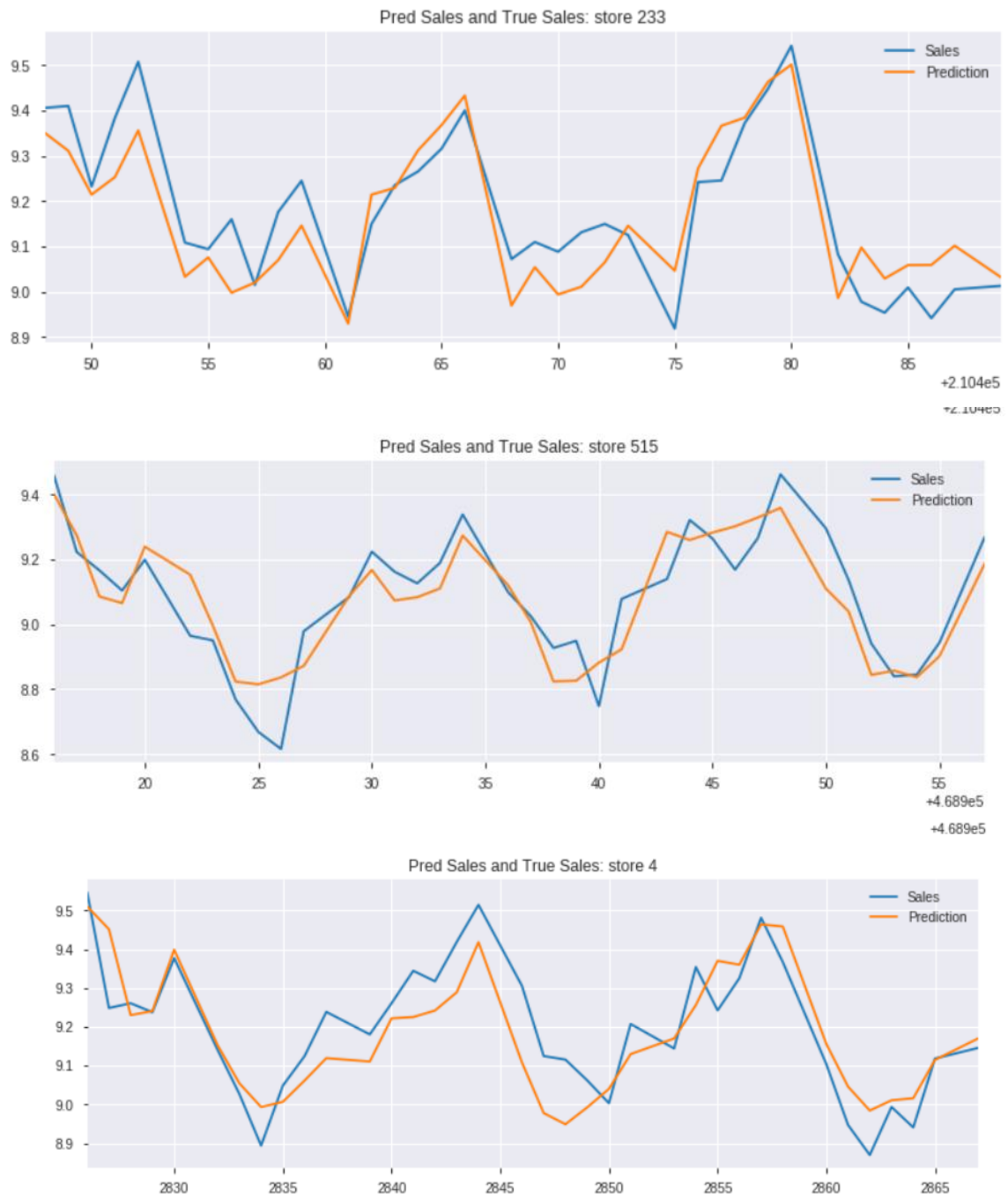
经 Kaggle 评分，模型结果直接得分为 0.11948，整体数据修正后达到 0.10887，达到项目结业最低要求，结论可靠。



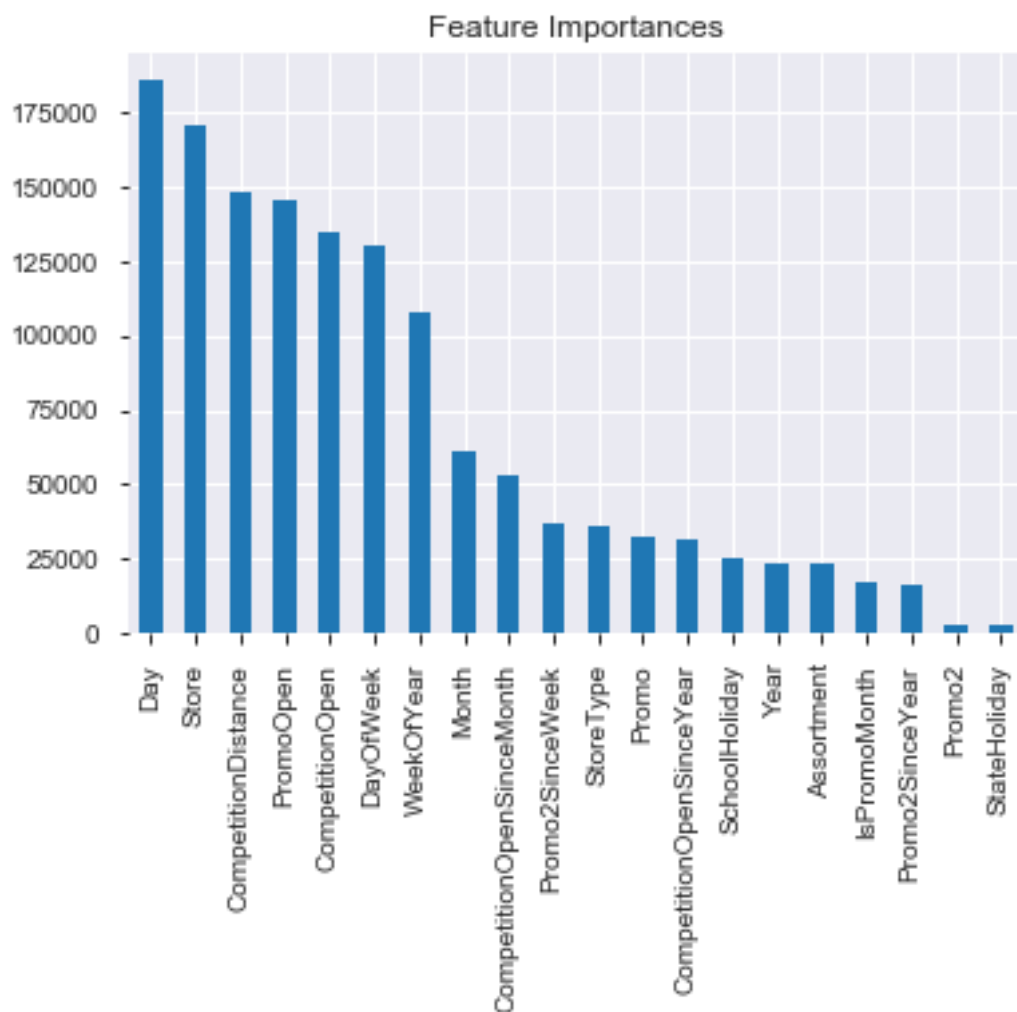
## 5 项目结论

### 5.1 结果可视化





上述四图为随机抽取验证集中四户商店，并对实际销售额与预测销售额进行对比，其中蓝线代表实际销售额，黄线代表预测销售额。由上图可见，模型在验证集中拟合能力较好，预测趋势与实际趋势同频，模型效果较好。



上图为最终模型的特征重要性表，Days 特征重要性排在第一位，表明影响最终销售额的特征中，时序性特征最为重要，因此后期改进时可以增加更多时序性特征。除开时序性特征，其次较为重要的特征为 CompetitionDistance、PromoOpen，说明影响竞争对手的距离与促销持续时间也是影响销量的较为重要的指标，后续可以结合时序性特征，综合竞争对手信息或者促销信息构建新的特征。

从业务上来看，竞争对手的距离直接影响客户流入量，距离近极易分流商店客户，导致销售额下降。另一方面，促销持续时间的长度，也会影响顾客的购买欲，销售会受到直接影响。业务制定销量提升方案时，要注意竞争对手与商店的距离，同时制定合理的促销天数也能给销量带来一定增长。

## 5.2 思考

本项目的第一个困难点在于如何在较短时间内训练出合理的模型。本次项目第一次在深度学习以外的场景使用 GPU 加速训练模型，加快了模型训练速度，因此尝试了网格搜索方法进行调参。若采用 CPU 进行训练，在较短时间内无法完成

调参工作。

自主编写网格搜索函数。虽然 XGboost 也提供了 Sci-kit learn 的 API，但在调试过程和 GPU 支持过程中还是遇到很多问题，故重新编写了简单调参函数与调参流水线。

该项目对时序性数据进行回归预测有很大的启发。时序的时间预测按照思维惯性一般会采用时序预测模型（例如 ARIMA），本次使用 XGBoost 这种提升树模型对时序性数据进行预测，且效果良好，为后续该类场景应用打开了新的思路。

### 5.3 待改进点

引入外部特征数据。其他案例中，引用包括温度在内的其他有效信息，可进一步提升模型预测准确度。

抽取更多特征信息。参照比赛第一名的方案，作者抽取了包括客均单价、上季度销量等在内的统计特征，作为新特征加入训练集，得到了较好的结果。

## 参考文献

- [1] Kaggle 神器 xgboost( <https://www.jianshu.com/p/7e0e2d66b3d4>)
- [2] Introduction to Boosted Trees  
<https://xgboost.readthedocs.io/en/latest/tutorials/model.html>
- [3] [6]XGboost 数据比赛实战之调参篇(完整流程)  
([https://blog.csdn.net/sinat\\_35512245/article/details/79700029](https://blog.csdn.net/sinat_35512245/article/details/79700029))
- [4] XGBoost Feature Importance( <https://www.kaggle.com/cast42/xgboost-in-python-with-rmspe-v2/code>)
- [5] Kaggle Rossmann 销售额预测 <https://zhuanlan.zhihu.com/p/67731440>

## 致 谢

本人在毕业设计以及学习课程的时间里，得到了 Udacity 的导师、群学友的热心帮助和支持，在此向他们表示诚挚的感谢。

台 亮

二〇二〇年一月