Rossmann Store Sales

I. 问题的定义

项目概述

Rossmann 是欧洲的一家连锁药店，在欧洲七个国家中经营着 3000 多家药店。

在这个源自 Kaggle 比赛 Rossmann Store Sales 中，我们需要根据 Rossmann 药妆店的信息（比如促销，竞争对手，节假日）以及在过去的销售情况，来预测 Rossmann 未来的销售额。

相关数据集包括：

- train.csv -销售额的历史数据，包含以下特征：

  "Store","DayOfWeek","Date","Sales","Customers","Open","Promo","StateHoliday","SchoolHoliday"

- test.csv-做测试集，包含以下字段：

  "Id","Store","DayOfWeek","Date","Open","Promo","StateHoliday","SchoolHoliday"

- store.csv - 关于商店的附加信息，包含以下字段：

  "Store","StoreType","Assortment","CompetitionDistance","CompetitionOpenSinceMonth"，

  "CompetitionOpenSinceYear","Promo2","Promo2SinceWeek","Promo2SinceYear","PromoInterval"

问题陈述

我们要根据商店销售的历史数据，如竞争对手、节假日、促销等因素来对商店未来六周的日销售量进行预测，特征都是有标签的，属于监督学习问题，需要预测的指标是销售额，是连续变量，我们这里需要处理的问题属于有监督的回归问题，我准备使用 xgboost 建模求解，具体步骤如下：

- *数据预处理 （缺失数据处理和非数值型特征的处理）*
- *将 store 与 train 和 test 数据合并起来，剔除和拆分部分特征*
- *对数据变量做可视化分析，去掉异常值*
- *将 Train 和 Store 合并处理后的数据拆分为训练集和测试集开始训练*
- *配置 xgboost 参数并开始第一次的训练*
- *通过配置 xgboost 参数的方式来寻找最佳参数*
- *记录训练的评估结果并保存模型*
- *用模型对测试集数据做预测并保存*

评价指标

在这里，我将使用 kaggle 上推荐的 RMSPE 指标对预测结果进行评价，该指标反映了预测值与实际值的误差，越小代表模型表现越好。

- $RMSPE=\sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\frac{yi-\hat{y}}{yi}\right)^2}$
- *其中 yi 是真实销售数据，yhat 是模型预测数据*

II. 分析

数据集初步探索：

1.Store.csv 前五行数据：

	Store	StoreType	Assortment	CompetitionDistance	CompetitionOpenSinceMonth	CompetitionOpenSinceYear	Promo2
0	1	c	a	1270.0	9.0	2008.0	0
1	2	a	a	570.0	11.0	2007.0	1
2	3	a	a	14130.0	12.0	2006.0	1
3	4	c	c	620.0	9.0	2009.0	0
4	5	a	a	29910.0	4.0	2015.0	0

Promo2SinceWeek	Promo2SinceYear	PromoInterval
NaN	NaN	NaN
13.0	2010.0	Jan,Apr,Jul,Oct
14.0	2011.0	Jan,Apr,Jul,Oct
NaN	NaN	NaN
NaN	NaN	NaN

总共有 1115 个门店的销售记录，其中缺失数据特征是：

CompetitionDistance、CompetitionOpenSinceMonth、

CompetitionOpenSinceYear、 Promo2SinceWeek、Promo2SinceYear、

PromoInterval。CompetitionDistance 为空值的有 3 条记录，就默认附近没有竞争对手，将距离填入 99999；CompetitionOpenSinceYear、 Promo2SinceWeek 缺失值均有 354 条，所以后续我将不考虑这两个特征；Promo2SinceWeek、Promo2SinceYear、PromoInterval 缺失数据一致，可以看出是与 Promo2 对应的，Promo2 为 0，则为空。商店信息中，StoreType、Assortment、PromoInterval 将使用 get_dummies 转换为哑变量，competitiondistance 跨度较大，将做对数转换来建模。

2.Train.csv 前五行数据：

	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

Train.csv:总共有 1017209 条记录，时间范围为 2013 年 1 月 1 日~2015 年 7 月 31 日，无缺失数据。

3.Test.csv 前五行数据：

	Id	Store	DayOfWeek	Date	Open	Promo	StateHoliday	SchoolHoliday
0	1	1	4	2015-09-17	1.0	1	0	0
1	2	3	4	2015-09-17	1.0	1	0	0
2	3	7	4	2015-09-17	1.0	1	0	0
3	4	8	4	2015-09-17	1.0	1	0	0
4	5	9	4	2015-09-17	1.0	1	0	0

test.csv 总共有 41088 条记录，时间范围为 2015 年 8 月 1 日~2015 年 9 月 17 日，open 缺失的有 11 条记录，根据 Date、Promo、StateHoliday、SchoolHoliday 将空值填入 1。

数据的探索

数据集特征：

Id - 测试集中表示一条记录的编号。

Store - 每个商店的唯一编号。

Sales - 任意一个给定日期的销售营业额。

Customers – 任意给定一天的消费者数。（由于测试集中没有该指标，为保持一致，将 dropout 该指标）

Open - 商店是否开门标志，0=关，1=开。

StateHoliday - 表明影响商店关门的节假日，正常来说所有商店，除了极少数，都会在节假日关门，a=所有的节假日，b=复活节，c=圣诞节，所有学校都会在公共假日和周末关门。（数据整理 a=1,b=2,c=3,0=0）

SchoolHoliday - 表明商店的时间是否受到公共学校放假影响。

StoreType - 四种不同的商店类型（a、b、c、d）。(a=1,b=2,c=3,d=4,0=0)

Assortment - 描述种类的程度，a = basic, b = extra, c = extended。（a=1,b=2,c=3,0=0）

CompetitionDistance - 最近的竞争对手的商店的距离。（空值认为是附近没有竞争对手，所以填充为 99999）

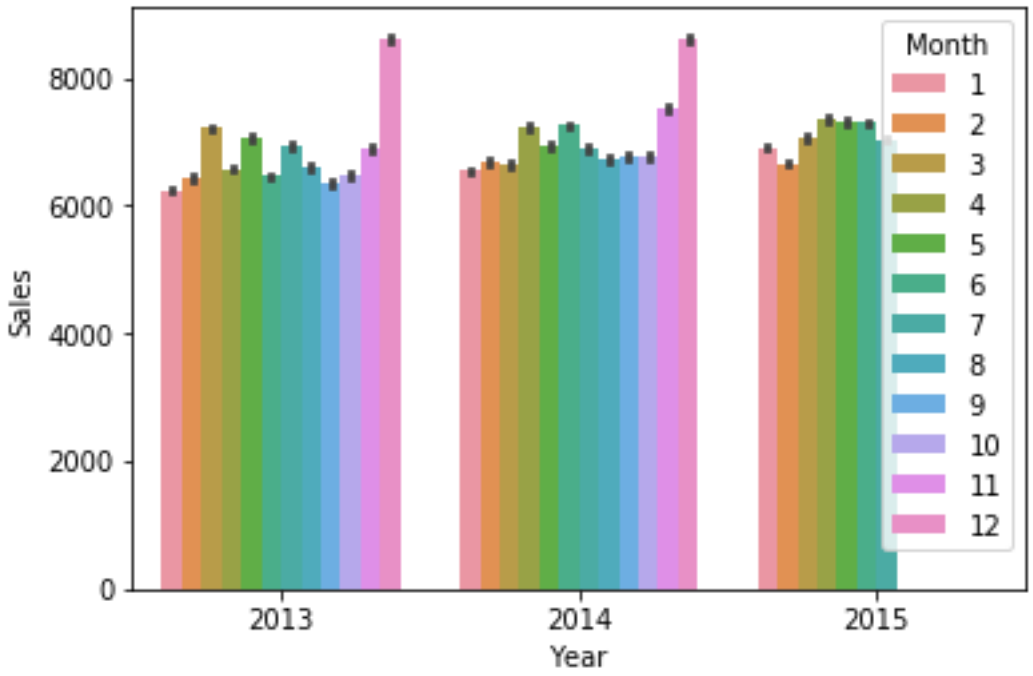
CompetitionOpenSince[Month/Year] - 最近的竞争者商店大概开业的年和月时间。

Promo - 表明商店该天是否在进行促销。

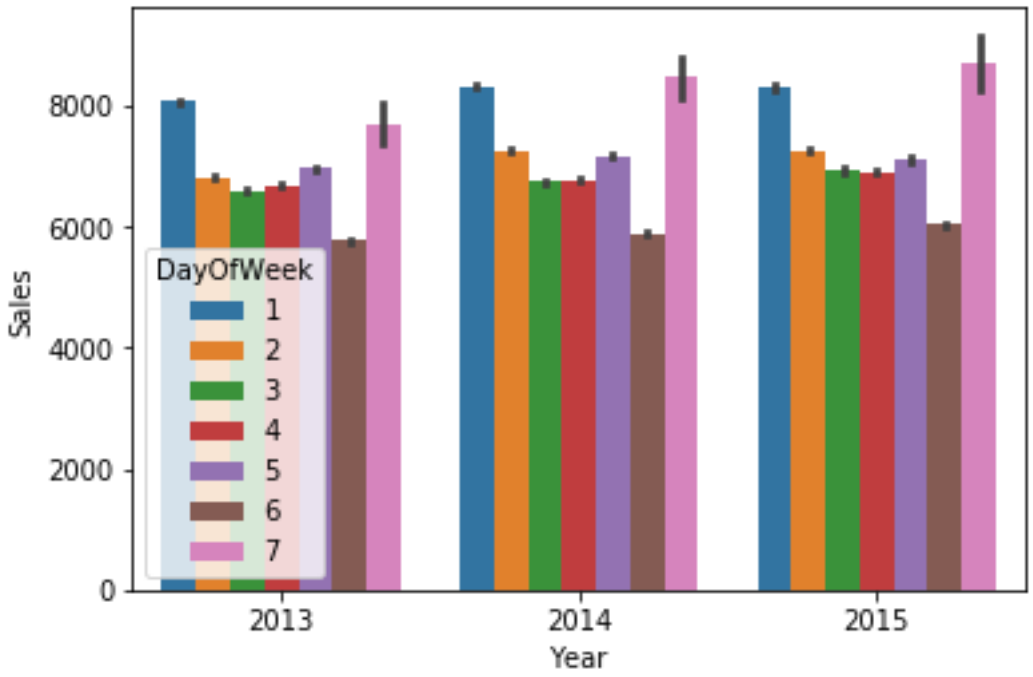
Promo2 - 指的是持续和连续的促销活动。: 0 =没有参加, 1 =参加。 （由于可视化图表观察是否有持续促销活动对销售额的影响影响不大，所以选择 dropout 相关指标）

探索性可视化

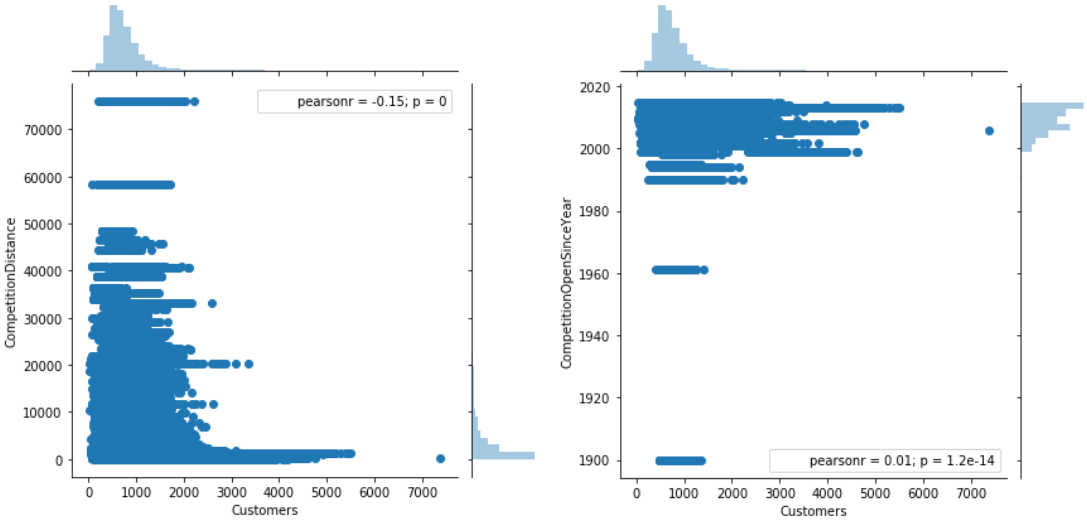
1.按每年的月份销售额分布来看，12 月销售明显高于其他月份，月份与销售量存在一定的关系。



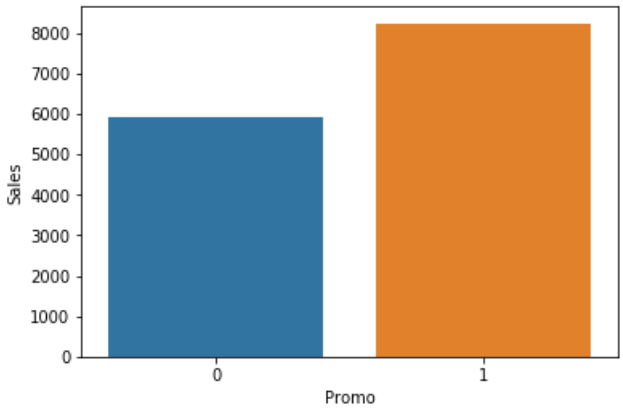
2.从按 2013~2015 年的每周的一天来看，周一和周天销售量都高于其他时间。



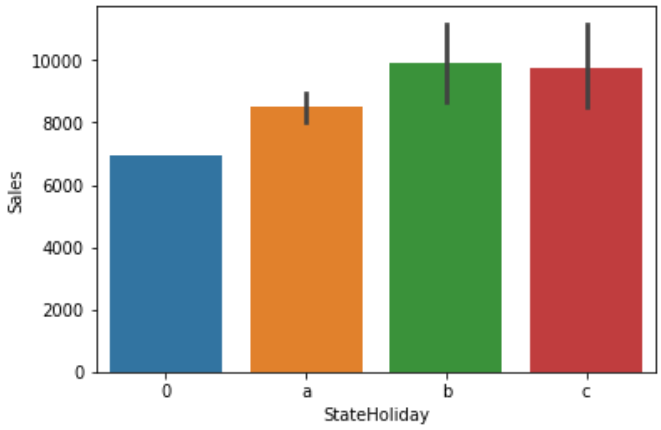
3.competitiondistance 数据高度倾斜，故对数据施加对数转换。



4.促销可以促进销售，达到销售额的增加。



5.可以看出节假日的销售额也高于非节假日。



算法和技术

Xgboost 算法：

xgboost 是在 GBDT 的基础上对 boosting 算法进行的改进，内部决策树使用的是回归树。其核心是每棵树通过对之前残差的学习，在每个叶子节点上找最优的权重。

传统 GBDT 以 CART 作为基分类器，xgboost 还支持线性分类器，这个时候 xgboost 相当于带 L1 和 L2 正则化项的逻辑斯蒂回归（分类问题）或者线性回归（回归问题）。

xgboost 在代价函数里加入了正则项，用于控制模型的复杂度。正则项里包含了树的叶子节点个数、每个叶子节点上输出的 score 的 L2 模的平方和。从 Bias-variance tradeoff 角度来讲，正则项降低了模型的 variance，使学习出来的模型更加简单，防止过拟合，这也是 xgboost 优于传统 GBDT 的一个特性。

xgboost的loss由两部分构成，前者优化经验误差，后者是控制泛化误差：

xgboost的目标函数如下：

$$Obj = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k)$$

Training loss　　Complexity of the Trees

相比较于传统的GBDT，xgboost正则更加的细化了，包括传统的L2正则以及叶子数目的正则项：

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

Number of leaves　　L2 norm of leaf scores

- *使用的参数：*

  Max_depth:树的最大深度，树越深，模型越复杂

  Learning_rate:学习率

N_estimators:迭代次数

Min_child_weight:叶子结点，需要的最小样本权重和

Subsample:构造每棵树的所用样本比例（样本采样比例）

Colsample_bytree:构造每棵树的所用特征比例

Reg:linear:线性回归

基准模型

对于销售额的预测 kaggle 上 private leaderboard 的 top 10%对于测试集 rmspe 为 0.11773 性能。这里我将提高基准得分至少达到 rmspe 为 0.11773。将预测错误率降低至百分之十二以内，我认为较合理。

III. 方法

数据预处理

- 1. 对 Date 数据转换，变为 Year、Month、Day、WeekofYear
- 2. 将 train.csv、test.csv 分别与 store.csv 合并
- 3. 选择 sales>0 的数据，将 open 为空的填充为 1
- 4. StateHoliday 除 0 外全部处理为 1，StoreType、Assortment、PromoInterval 转换为哑变量
- 5.CompetitionDistance 空值默认附没有竞争对手，填入 99999，由于该指标高度倾斜，故对数据做对数转换

- 6.对 sales 也进行对数转换
- 7. 'Promo2SinceYear','Promo2SinceWeek','Customers', 'CompetitionOpenSinceMonth','CompetitionOpenSinceYear'我选择 drop 掉，不考虑作为销售额的影响指标进行建模
- 8.关于训练集与验证集的划分，我考虑将 trian.csv 最后两周（2015-7-18~2015-7-31）数据拿来验证，其余用来训练。

执行过程

第一次执行 xgboost 算法：

```
#引入XGBoost
import xgboost as xgb
from xgboost.sklearn import XGBRegressor

#模型参数设置
clf = XGBRegressor(max_depth=10,
                   learning_rate=0.1,
                   objective='reg:linear',
                   n_estimators=100,
                   min_child_weight=1,
                   subsample=0.8,
                   colsample_bytree=0.8)
eval_set = [(X_test, y_test), (X_train, y_train)]
starttime = datetime.datetime.now()
clf.fit(X_train, y_train, eval_metric='rmse', early_stopping_rounds=10, verbose = True, eval_set = eval_set)
endtime = datetime.datetime.now()

yhat = np.exp(clf.predict(X_test))
rmspe = rmspe(yhat, np.exp(y_test))
print(endtime - starttime)
print('RMSPE: {:.6f}'.format(rmspe))
```

得到 RMSPE: 0.180234，上传 kaggle 得分 0.17452

完善

Xgboost 第一次优化，调整 learning_rate=0.05，n_estimators=1000，得到 RMSPE: 0.109197；上传 kaggle 得分为：0.11348，已经达到基准得分。

第二次调整，将 n_estimators 调整到 2000， RMSPE: 0.104664，耗时增加到 4149s，上传 kaggle 得分为：0.11217。

第三次，按照审阅者提议，将 n_estimators 调到 20000，发现模型产生明显的过拟合，耗时达到 12 个小时，RMSPE: 0.114455，上传 kaggle 得分为 0.12957。

IV. 结果

最终选定模型为 max_depth=10,n_estimators=2000， 提交 kaggle 得分为 0.11217。

```
clf2 = xgb.XGBRegressor(max_depth=10,
                        learning_rate=0.05,
                        objective='reg:linear',
                        n_estimators=2000,
                        min_child_weight=1,
                        subsample=0.8,
                        colsample_bytree=0.8)
eval_set = [(X_val, y_val), (X_train, y_train)]
starttime = datetime.datetime.now()
clf2.fit(X_train, y_train, eval_metric='rmse', early_stopping_rounds=10, verbose = True, eval_set = eval_set)
endtime = datetime.datetime.now()
yhat = np.exp(clf2.predict(X_val))
rmspe = rmspe(yhat, np.exp(y_val))
print('RMSPE: {:.6f}'.format(rmspe))
print (endtime - starttime).seconds
```

模型的评价与验证

使用 xgboost 建模第一次训练耗时 219s，RMSPE: 0.180234;
进行一次调参耗时 2078s，RMSPE: 0.109197;
第二次调参，训练耗时 4149s，RMSPE: 0.104664;
第三次调参，训练耗时 43040s, RMSPE: 0.114455;

第二次调参优化后上传 kaggle 的得分最高为 0.11217，已达到期待效果，运行时间上也较能接受，参数选择也较为合理，对于销售额的预测足够稳健可靠。

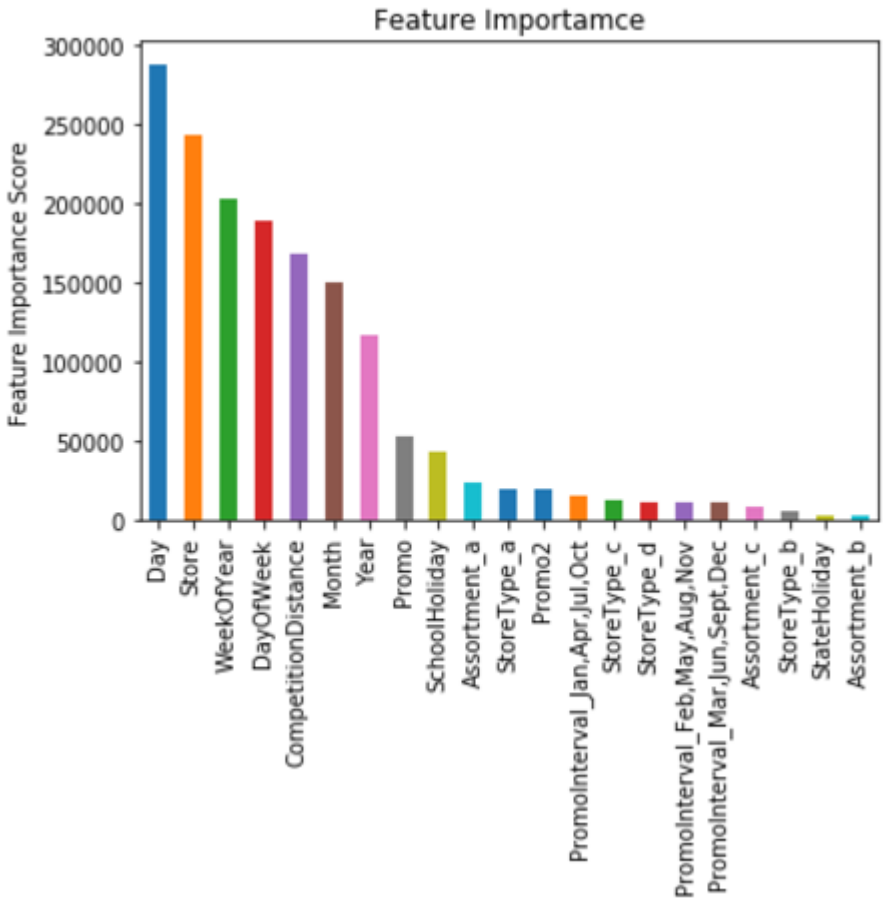
合理性分析

相比之前最开始使用的 xgboost 模型，调参后性能得到提升，最后得分

rmpse=0.11217，已达到一开始定制的 0.11773 的目标，预测模型的表现较好。

V. 项目结论

特征重要度可视化：



从影响销售额的前五项特征来看：Day 是一个重要因素，学校放不放假与时间有关，每周周末还是工作日，节假日都与 Day 相关，因此对销售额的影响程度较大。Competitiondistance 是竞争对手的一个重要指标，竞争对手与门店的距离也会对销售额产生较大的影响。

对项目的思考

第一次自己从数据清理开始做一个项目，还是挺具有挑战性，相比之前的填空式完成项目，这个花费了更多时间看书，还有网上查找资料。处理过程中，认为比较难的是对数据特征的选择，哪些特征要选择来建模，还有对于特征的处理，空值如何处理。最后是模型的选择，由于只尝试了 xgboost，虽然模型的效果较好，但是运算速度较之前做过的优达的项目速度较慢，而且参数的调整也只是手动调整了几个，对参数整体感知还不够。

需要作出的改进

考虑电脑性能等因素自己手动选择了几个参数进行调整，还有模型选择上，由于推荐用 xgboost，所以直接使用，没有尝试其他回归模型。Xgboost 耗时实在太长，未来可以考虑用 lightgbm 算法来建模。

参考：

1. https://www.kaggle.com/cast42/xgboost-in-python-with-rmspe-v2/code

2. Udacity-机器学习进阶 xgb 实现 P0 泰坦生还预测.ppt

3. https://homes.cs.washington.edu/~tqchen/pdf/BoostedTree.pdf

4. https://blog.csdn.net/sb19931201/article/details/52557382