机器学习工程师　纳米学位

林翔　毕业项目报告

毕业项目：预测 Rossmann 未来的销售额

1　问题的定义

1.1 项目概述

项目源自 2015 年的 Kaggle 比赛 Rossmann Store Sales。Rossmann 是欧洲一家大型的连锁药店。Rossmann 是欧洲的一家连锁药店，目前一共拥有大约 3,000 家药店， 它们分布在 7 个欧洲国家。现在的任务是通过历史数据，预测 6 周后的每日销售量。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。 每个人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。

可靠的销售预测对门店的精细化运营具有非常大的帮助。销售量的预测能够使商店的管理人员创建有效的员工时间表 ,提高生产力。此外 ,良好的销售预测模型，还可以让管理人员调整供应链，制定合理的促销策略与竞争策略。另外还可以帮助门店，提前准备合适的人力资源和物资数量，降低成本，提高营业额以及用户体验。

1.2 问题陈述

该项目要求基于三年的销量历史来预测门店未来六周的销售额 ,按照机器学习对

于问题的分类方法，该问题属于回归问题，同时也属于时间序列问题。我们需要从所给的数据集中提取可能对销售额有影响的特征，进行有效的回归模型进行预测。主要分为以下几步：

- ✓ 通过探索性分析(Exploratory Data Analysis)来观测数据的分布情况，还有相应的缺失值的情形，也同时为后面的特征工程(Feature Engineering)作参考和准备。
- ✓ 数据预处理(Pre-processing data)来处理诸如类别信息，缺失值，时间序列等相应信息，以便后续的特征提取。
- ✓ 特征工程(Feature Engineering)最大限度地提取出特征以便后续使用。
- ✓ 根据提取的特征来进行回归模型的建立，建模过程中尝试特征选择和特征融合(Feature selection and model ensemble)来取得比较好的预测效果。

最终希望能够根据这些特征建立的模型相对准确的对预测日的销售额进行预测。

1.3 指标评价

评价指标选用的是百分比均方根误差（the Root Mean Square Percentage Error(RMSPE)）。目标是，XGBoost 得到的模型的 RMSPE 小于 0.11773。计算公式如下：

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

另外还需要评估各个特征间的相关度和对销量的相关度。其中 y_i 表示单个门店单天的销售额，$\hat{y}_i$ 表示对应门店对应单天的销售额预测值，对于单个门店单天的销售额为0的情况不予评价。采用百分比误差可以有效降低大尺度的数据对最终误差的影响，对于门店的销售额数据来说，某些节假日或者某些特殊

日的销售额肯定比平常要高出不少，如果采用均方根误差，那些很大的销售额数据就会对误差评估产生较大的影响，从而可能对模型的好坏出生误判。

2 分析

2.1 数据探索和可视化分析

数据集中train.csv有1,017,209行记录，跨度为13年1月1号到15年7月31号。另外test.csv为41,088行记录,时间跨度为15年8月1号到9月17号。数据字段分布如下：

Train	Store	Test
Store	Store	Id
DayOfWeek	StoreType	Store
Sales	Assortment	DayOfWeek
Customers	CompetitionDistance	Date
Open	CompetitionOpenSinceMonth	Open
Promo	Promo2	Promo
StateHoliday	Promo2SinceWeek	StateHoliday
SchoolHoliday	Promo2SinceYear	SchoolHoliday
	PromoInternal	

#查看训练集前五行跟后五行

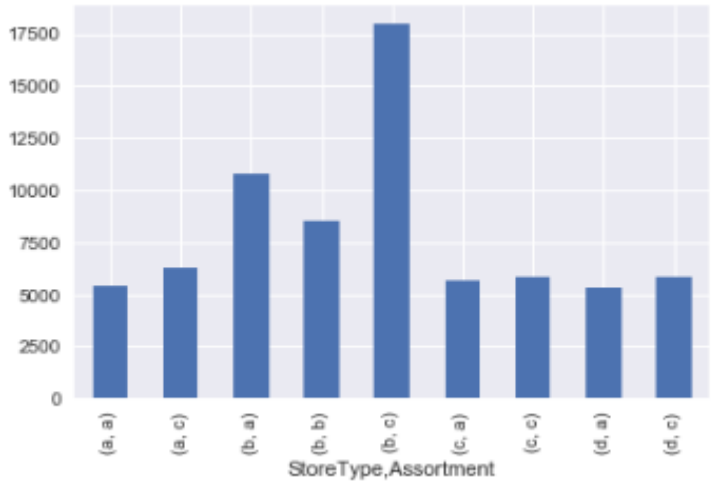
train.head().append(train.tail())

	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
1017204	1111	2	2013-01-01	0	0	0	0	a	1
1017205	1112	2	2013-01-01	0	0	0	0	a	1
1017206	1113	2	2013-01-01	0	0	0	0	a	1
1017207	1114	2	2013-01-01	0	0	0	0	a	1
1017208	1115	2	2013-01-01	0	0	0	0	a	1

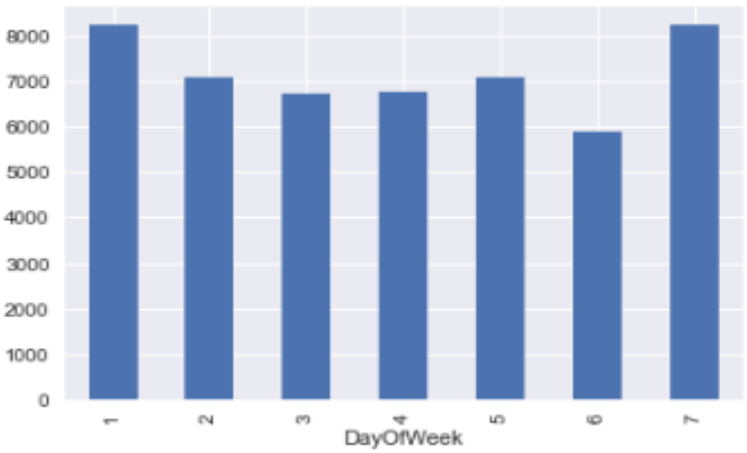
对于Store1做一个随着时间的可视化分析如下，可见要预测的标签是根据

时间有个时间性的规律，当年的11月和12月销量为最佳：

将训练集的平均销量按销售类型和品类做个分布如下，表明某些店铺的某些品类销量多：



DayofWeek的平均销量

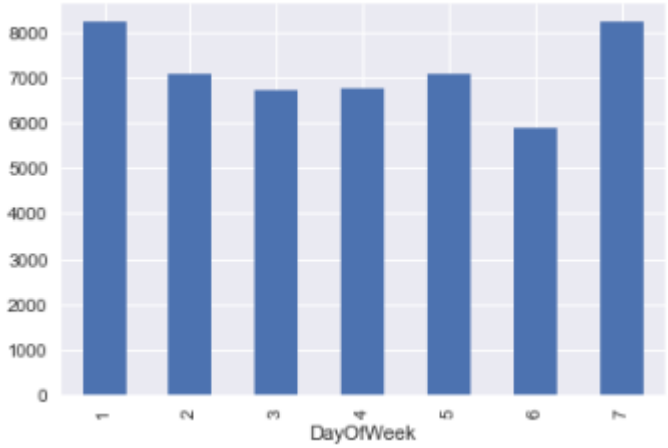


DayofWeek的平均客户数量，与平均销量有密切的关系。

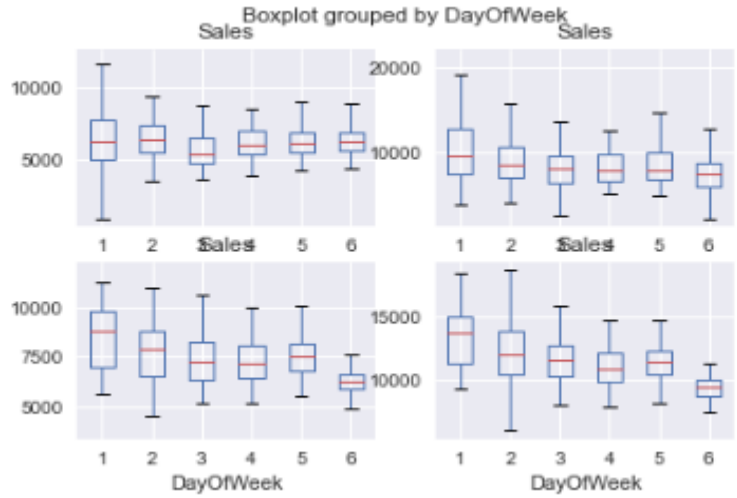
```
#Mean Customers per week

dow = train[(train[Customers]!=0)].groupby(['DayOfWeek']). Customers.mean()

dow.plot('bar')
```



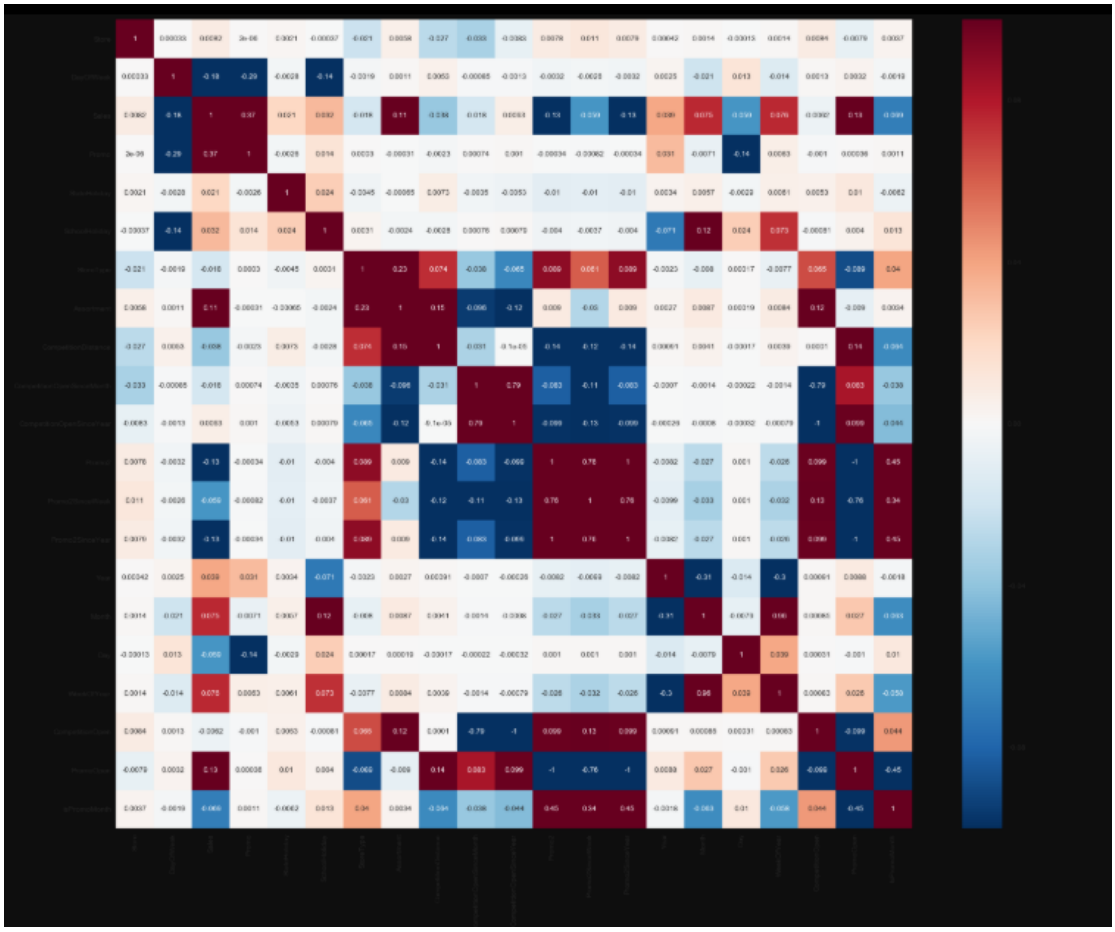
取四个店铺看平均销量，DayofWeek的销量差别很大：



此外从对 2014 年 6 月-9 月份的销量来看，6，7 月份的销售趋势与 8，9 月份类似，因为我们需要预测的 6 周在 2015 年 8，9 月份，因此我们可以把 2015 年 6，7 月份最近的 6 周数据作为 hold-out 数据集，用于模型的优化和验证。

#分析训练数据集中特征相关性以及特征与'Sales'标签相关性

可以看出，sales 与 promotion, Assortment, 店铺已促销时间密切相关



2.2 算法以及技术

由前面的分析可知，这是一个回归问题。所以首先想到的简单的线性回归模型进行拟合，观察拟合效果。鉴于给出的训练数据集中包含很多不同类型的特征数据，包括数值型、类别型等，于是尝试采用集成学习的回归模型就成了很自然的想法，常用的集成学习的模型有 Gradient Tree Boosting、Extreme Gradient Boosting（xgboost）等，而 xgboost 相比于其他 Gradient Boosting 的模型，速度更快，模型表现更好，所以该项目主要采用的就是基于 xgboost 的回归模型。下面对线性回归模型和 Gradient Boosting 方法做一个简单的介绍。

2.2.1 线性回归

给定数据集，其中线性回归（linear regression）试图学得一个通过属性的线性组合来进行预测的函数。对于本项目说，就是要从给定的数据集中提取特征，通过对这些特征的线性组合来预测销售额。线性回归使用最佳的拟合直线在因变量（销售额）和特征（自变量）之间建立一种关系。而最佳拟合直线的求解是通过最小二乘法来完成，在线性回归中，最小二乘法就是试图找到一条直线，使所有样本到直线上的欧氏距离之和最小。但是基于最小二乘的参数估计依赖于不同特征之间的独立性，希望不同特征之间的相关性越好，另外线性回归对异常值非常敏感，异常值会严重影响回归线，最终影响预测值。

2.2.2 XGBoost

Gradient Boosting 采取分层学习的方法，通过 m 个步骤来得到最终模。Gradient Boosting 算法中最典型的基学习器是决策树。Gradient Boosting

Decision Tree(GBDT)就是一种结合决策树的 Gradient Boosting 算法实现。这里的决策树是回归树，GBDT 中决策树是个弱模型，树的深度和叶子节点的数量一般都较小，很常用的 Gradient Boosting 算法的实现 xgboost，xgboost 是 Gradient Boosting 算法的一种高效实现。xgboost 中的基学习器除了可以是 CART(gbtree)，也可以是线性分类器(gblinear)。传统 GBDT 在优化时只用到□阶导数信息，xgboost 则对代价函数进行了二阶泰勒展开，同时用到了一阶导数和二阶导数。xgboost 在代价函数中加入了正则项，用于控制模型的复杂度，学习出来的模型更加简单，防止过拟合。xgboost 在进行完一次迭代后，会将叶子节点的权重乘上缩减（shrinkage）系数，主要是为了削弱每棵树的影响，让后面有更大的学习空间。XGBoost 学习模型主要的目标是得到一个模型，使得预测值与真实值之间的误差尽可能小。需要定义一个目标函数(Objective Function)，来衡量模型是否能很好的拟合训练数据。目标函数是 XGBOOST 的一个特点，为了防止过拟合，XGBoost 的目标函数由损失函数和复杂度组成。复杂度又由叶子数量和 L2 正则组成。

原理和数学推导过程如下：目标函数 = 损失函数 + 正则项。

$$\mathcal{L}(\phi) = \sum_i l(\hat{y}_i, y_i) + \sum_k \Omega(f_k)$$
$$\text{where } \Omega(f) = \gamma T + \frac{1}{2}\lambda \|w\|^2$$

对于每次迭代过程，可以将一棵树的训练目标函数写成形式如下：

$$\mathcal{L}^{(t)} = \sum_{i=1}^{n} l(y_i, \hat{y_i}^{(t-1)} + f_t(\mathbf{x}_i)) + \Omega(f_t)$$

输入是 t-1 轮后预测的值，真实值，用来拟合残差 f(x)。对于这个式子，我们不知道 loss 的具体形式，所以无法对 f(x)进行有效的最优估计。于是进行如下推导：首先将目标函数泰勒二阶展开近似

$$\mathcal{L}^{(t)} \simeq \sum_{i=1}^{n}[l(y_i, \hat{y}^{(t-1)}) + g_i f_t(\mathbf{x}_i) + \frac{1}{2}h_i f_t^2(\mathbf{x}_i)] + \Omega(f_t)$$

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

正则项展开为：

$$\tilde{\mathcal{L}}^{(t)} = \sum_{i=1}^{n}[g_i f_t(\mathbf{x}_i) + \frac{1}{2}h_i f_t^2(\mathbf{x}_i)] + \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2$$

$$= \sum_{j=1}^{T}[(\sum_{i\in I_j} g_i)w_j + \frac{1}{2}(\sum_{i\in I_j} h_i + \lambda)w_j^2] + \gamma T$$

这是一个二次项形式，所以最后使得目标函数最小的 w 为

$$w_j^* = -\frac{\sum_{i\in I_j} g_i}{\sum_{i\in I_j} h_i + \lambda}.$$

带入原方程最小值：

$$\tilde{\mathcal{L}}^{(t)}(q) = -\frac{1}{2}\sum_{j=1}^{T}\frac{(\sum_{i\in I_j} g_i)^2}{\sum_{i\in I_j} h_i + \lambda} + \gamma T.$$

那么最后求得的 w 就是目标函数在一个样本集合条件下的最优解。

2.3 基准

基准模型是该门店销售额的中位数。用于跟Xgboost模型进行对比。采用与 Kaggle 一样的评估指标，RMSPE，即均方根误差。目标是，XGBoost 得到的模型的 RMSPE 小于 0.11773。

3 方法

3.1 数据预处理

3.1.1 查看数据缺失值

```
display(train.isnull().sum(),test.isnull().sum(),store.isnull().sum())
```

测试文件有 11 个店铺是否 open 的缺失数据，都来自于 622 号店铺，从周 1 到周 6 而且没有假期，所以我们认为这个店铺的状态应该是正常营业的。店铺促销信息的缺失是因为没有参加促销活动，所以我们以 0 填充，竞争对手的缺失不明，也以 0 来填充。我们将 test 中的 open 数据补为 1，即营业状态。CompetitionDistance, CompetitionSinceMonth, CompetitionSinceYear 等竞争对手信息在实验中用中值效果不太好，缺失值都用 0 来代替。

3.1. 2 特征的值或类型的转换

- ✓ 将存在其他字符表示分类的特征转化为数字，包括 StoreType, Assortment 和 StateHoliday。注： mappings = {'0':0, 'a':1, 'b':2, 'c':3, 'd':4}。也可以将

这些特征转化为独热编码，但是训练效果不如直接 Label encoding。

- ✓ 将时间特征进行拆分为年月日，并加入'WeekOfYear'特征。
- ✓ 新增'CompetitionOpen'和'PromoOpen'特征,计算某天某店铺的竞争对手已营业时间和店铺已促销时间，用月为单位表示。将'PromoInterval'特征转化为'IsPromoMonth'特征,表示某天某店铺是否处于促销月，1 表示是，0 表示否。
- ✓ 将目标变量 Sales 进行对数处理，便于模型 RMSPE 误差计算。

3.2 执行过程

3.2.1 建立模型

对于模型训练集和交叉验证集的划分 ,考虑到模型最终是要预测未来连续六周的销售额，所以采用了 train 中最后六周的训练数据作为交叉验证集，其余的数据用来做训练，这样的策略更符合模型的目标。使用历史销售数据来预测未来的销售数据。而不是像一般采用的将训练集随机划分为训练集和交叉验证集的策略。数据方面只采用店铺为开而且销售额大于 0 的数据。

XGboost 初始模型构建

#参数设定

```
params = {"objective": "reg:linear",
          "booster" : "gbtree",
          "eta": 0.03,
          "max_depth": 10,
          "subsample": 0.9,
          "colsample_bytree": 0.7,
          "silent": 1,
          "seed": 10
          }
```

num_boost_round = 6000
dtrain = xgb.DMatrix(ho_xtrain, ho_ytrain)
dvalid = xgb.DMatrix(ho_xtest, ho_ytest)

一般参数:
booster[default=gbtree]选择基分类器 gbtree、gblinear 树或线性分类器
silent [default=0] 是否输出详细信息 0 不输出 1 输出
nthread [default to maximum number of threads available if not set]线程数默认最大
Tree Booster 参数：
1. eta [default=0.3]: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。
2. min_child_weight [default=1]: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。
3. max_depth [default=6]: 每颗树的最大深度，树高越深，越容易过拟合。
4. gamma [default=0]：在树的叶子节点上作进一步分区所需的最小损失减少。越大，算法越保守。[0,∞]
5. max_delta_step [default=0]：这个参数在更新步骤中起作用，如果取 0 表示没有约束，如果取正值则使得更新步骤更加保守。可以防止做太大的更新步子，使更新更加平缓。 通常，这个参数是不需要的，但它可能有助于逻辑回归时，类是非常不平衡。设置它的值为 1-10 可能有助于控制更新。
6. subsample [default=1]：样本随机采样，较低的值使得算法更加保守，防止过拟合，但是太小的值也会造成欠拟合。
7. colsample_bytree [default=1]：列采样，对每棵树的生成用的特征进行列采样.一般设置为： 0.5-1
8. lambda [default=1]：控制模型复杂度的权重值的 L2 正则化项参数，参数越大，模型越不容易过拟合。
9. alpha [default=0]:控制模型复杂程度的权重值的 L1 正则项参数，参数值越大，模型越不容易过拟合。
10. scale_pos_weight [default=1]如果取值大于 0 的话，在类别样本不平衡的情况下有助于快速收敛。
11. tree_method[default='auto']可选 {'auto', 'exact', 'approx'} 贪心算法(小数据集)/近似算法(大数据集)
学习任务参数：
objective [default=reg:linear]定义最小化损失函数类型
最常用的值有：
binary:logistic 二分类的逻辑回归，返回预测的概率(不是类别)。
multi:softmax 使用 softmax 的多分类器，返回预测的类别(不是概率)。
在这种情况下，你还需要多设一个参数：num_class(类别数目)。
multi:softprob 和 multi:softmax 参数一样，但是返回的是每个数据属于各个类别的概率。
seed [default=0]随机种子
eval_metric[根据目标 objective 默认]
对于有效数据的度量方法。
对于回归问题，默认值是 rmse，对于分类问题，默认值是 error。
num_round 迭代次数/树的个数

3.2.1 执行和分析

将此 xgboost 模型经过 6000 次迭代，100 次为一轮，如果在某轮中

eval-rmspe 没有得到提升则停止。

经过模型的训练，得到如下结果：

```
[3176]  train-rmse:0.065798   eval-rmse:0.117261    train-rmspe:0.0
69603   eval-rmspe:0.127442
[3177]  train-rmse:0.065792   eval-rmse:0.117261    train-rmspe:0.0
69595   eval-rmspe:0.127442
[3178]  train-rmse:0.065785   eval-rmse:0.117261    train-rmspe:0.0
69587   eval-rmspe:0.127442
[3179]  train-rmse:0.065779   eval-rmse:0.117258    train-rmspe:0.0
69581   eval-rmspe:0.127438
[3180]  train-rmse:0.065776   eval-rmse:0.117259    train-rmspe:0.0
69577   eval-rmspe:0.127439
[3181]  train-rmse:0.065769   eval-rmse:0.117258    train-rmspe:0.0
69569   eval-rmspe:0.127438
[3182]  train-rmse:0.06576    eval-rmse:0.117259    train-rmspe:0.0
69554   eval-rmspe:0.12744
[3183]  train-rmse:0.065755   eval-rmse:0.117267    train-rmspe:0.0
69549   eval-rmspe:0.127452
[3184]  train-rmse:0.06575    eval-rmse:0.117269    train-rmspe:0.0
69542   eval-rmspe:0.127454
[3185]  train-rmse:0.065742   eval-rmse:0.117267    train-rmspe:0.0
69533   eval-rmspe:0.127452
Stopping. Best iteration:
[3085]  train-rmse:0.066367   eval-rmse:0.117232    train-rmspe:0.0
70345   eval-rmspe:0.127409

Training time is 4991.144717 s.
validating
RMSPE: 0.127452
```

在 hold-out 的交叉验证集上进行前五行数据的结果分析，误差和权重比还是比较小的：

IsPromoMonth	Sales	Prediction	Ratio	Error	Weight
0	8.568646	8.581182	1.001463	0.001463	0.998539
0	8.521384	8.531733	1.001214	0.001214	0.998787
0	8.472823	8.470778	0.999759	0.000241	1.000241
0	8.519590	8.488213	0.996317	0.003683	1.003697
0	8.716536	8.571804	0.983396	0.016604	1.016885

分析所有店铺的预测结果

```
Mean Ratio of predition and real sales data is 1.0024289773551884: store all
```

分析保留数据集中任意三个店铺的预测结果

```
Mean Ratio of predition and real sales data is 0.9995037008441101: store 979
Mean Ratio of predition and real sales data is 1.0010418779153278: store 788
Mean Ratio of predition and real sales data is 0.9962927423119652: store 295
```

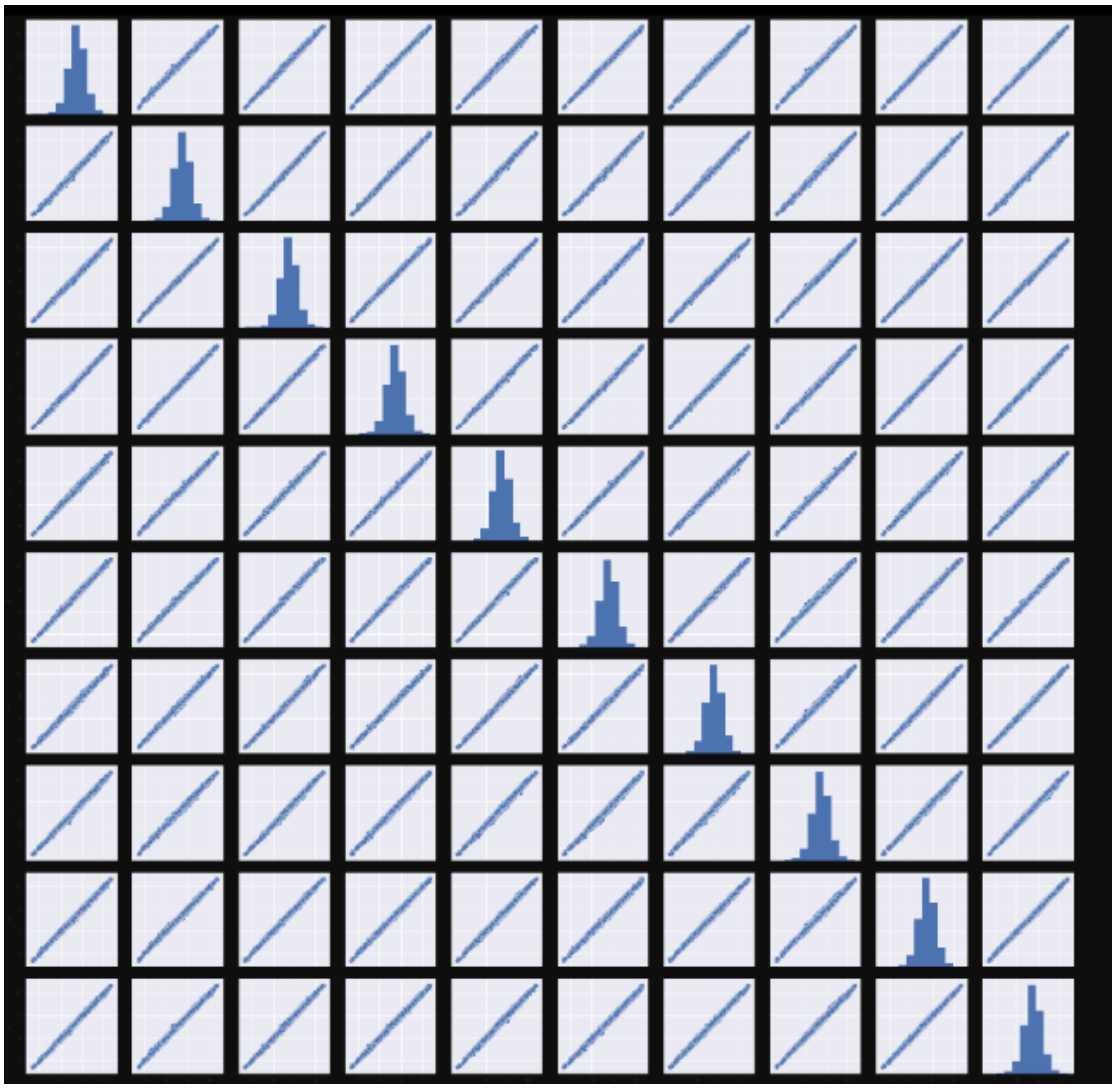
3.3 完善

分析偏差最大的 10 个预测结果我们的初始模型已经可以比较好的预测 hold-out 数据集的销售趋势，但是相对真实值，我们的模型的预测值整体要偏高一些。从对偏差数据分析来看，偏差最大的 3 个数据也是明显偏高。因此我们可以以 hold-out 数据集为标准对模型进行偏差校正。

- ✧ 把已经训练好的模型进行权重校正，当校正系数为 0.995 时，hold-out 集的 RMSPE 得分最低 :0.120686 ,相对于初始模型 0.127452 得分有很大的提升。记为模型 B。
- ✧ 以不同的店铺分组进行细致校正，每个店铺分别计算可以取得最佳 RMSPE 得分的校正系数和权重，细致校正后的 hold-out 集的得分为 0.114002 ,相对

于整体校正的 0.120686 的得分又有不小的提高。用初始和校正后的模型对训练数据集进行预测，得到三个数据集。记为模型 C。

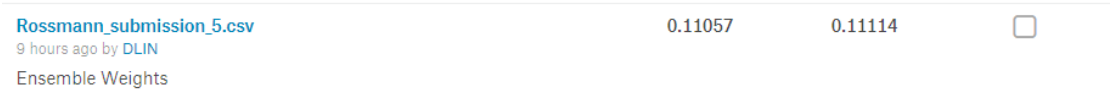
- 然后我们用不同的 seed 训练 10 个模型,每个模型单独进行细致偏差校正后进行融合. 模型融合可以采用简单平均或者加权重的方法进行融合。从下图来看，这 10 个模型相关性很高，差别不大，所以权重融合我们只考虑训练中单独模型在 hold-out 模型中的得分情况分配权重



- 均值融合的 public score 到达 0.11122。

Submission and Description	Private Score	Public Score	Use for Final Score
Rossmann_submission_4.csv 9 hours ago by DLIN Mean Ensemble	0.11053	0.11122	☐

✧ 加权融合的算法将 RMSPE 的值提高到 0.112982，Kagger 的 public Score 也可以到达 0.11114



由此，训练融合模型，并进行加权融合后将 RMSPE 的值提高到 0.112982。

4 结果

4.1 模型评价及验证

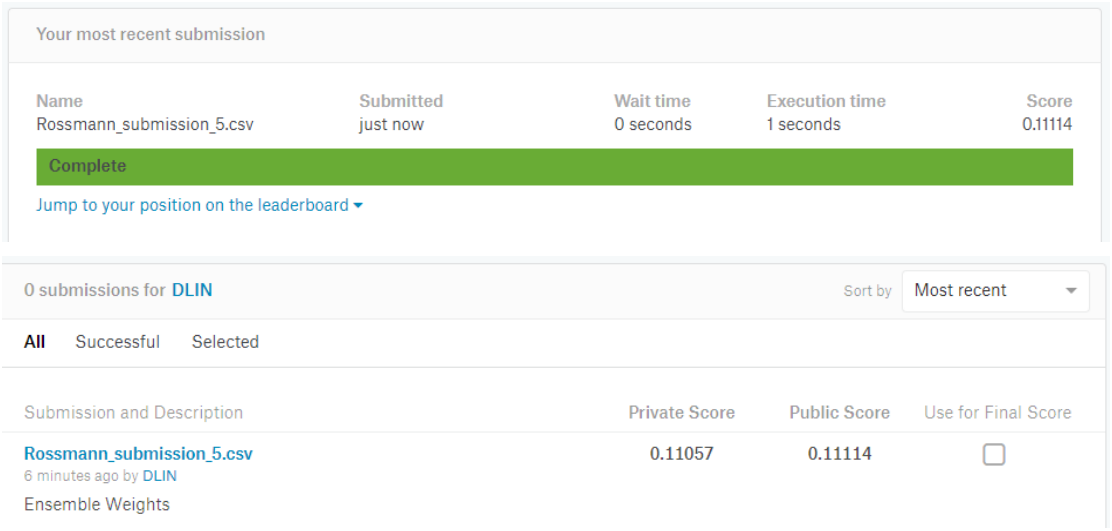
从新旧模型预测结果最大的几个偏差对比的情况来看，最终的融合模型在这几个预测值上大多有所提升，证明模型的校正和融合确实有效。

	Store	Ratio	Error	Store_new	Ratio_new	Error_new
264207	292	1.235983	0.235983	292	1.221274	0.221274
711449	782	1.184041	0.184041	782	1.172604	0.172604
827582	909	1.167560	0.167560	909	1.155518	0.155518
827591	909	0.862041	0.137959	909	0.852097	0.147903
797965	876	0.867589	0.132411	876	0.853161	0.146839
264218	292	0.879444	0.120556	292	0.871383	0.128617
264213	292	1.120542	0.120542	292	1.111659	0.111659
797963	876	0.880855	0.119145	876	0.873200	0.126800
711448	782	1.113012	0.113012	782	1.102314	0.102314
456286	501	1.112082	0.112082	501	1.096999	0.096999

4.1 合理性分析

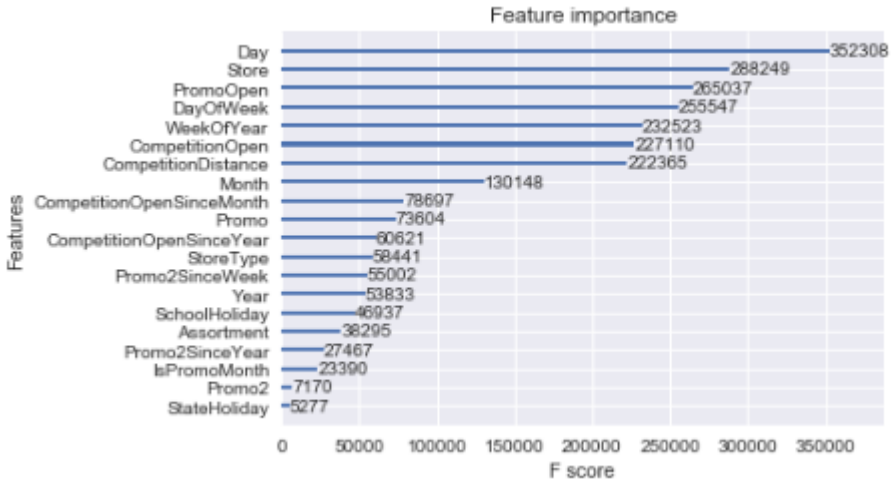
RMSPE 的值最终为 0.112982，小于基准模型 0.11773。合理。

将加权融合模型放在 Kaggle 里评价 Private Score 为 0.11057，Public Score 为 0.11114，结果应该说相当可观。截图如下：



5. 项目结论

5.1 结果可视化



模型特征重要性及最佳模型结果分析

从模型特征重要性分析，比较重要的特征有四类包括 1.周期性特征'Day'，'Day OfWeek'，'WeekOfYera'，'Month'等，可见店铺的销售额与时间是息息相关的，尤其是周期较短的时间特征；2.店铺差异'Store'和'StoreTyp'特征，不同店铺的销售额存在特异性；3.短期促销（Promo)情况:'PromoOpen'和'Promo'特征，促销时间的长短与营业额相关性比较大；4.竞争对手相关特征包括：'CompetitionOpen'，'CompetitionDistance'，'CompetitionOpenSinceMoth'以及'Competition OpenScinceyear'，竞争者的距离与营业年限对销售额有影响。作用不大的特征主要两类包括：1.假期特征：'SchoolHoliday'和'StateHoliday'，假期对销售额影响不大，有可能是假期店铺大多不营业，对模型预测没有太大帮助。2.持续促销(Promo2)相关的特征：'Promo2'，'Prom2SinceYear'以及'Prom2SinceWeek'等特征，有可能持续的促销活动对短期的销售额影响有限。

5.2 对项目的思考

本项目的整个流程主要包括一下几个步骤：

- 数据的探索和可视化
- 基准模型确立
- 数据预处理
- 特征提取
- Xgboost 模型选择，调参和融合
- 结果分析和提交

在整个项目过程中，特征提取花了比较多的精力，因为特征的好坏对模型最终的表现具有很大的影响，好的特征能大大的提升模型最终的效果。在模型方面，当

确定是使用 gboost 模型后，得到的第一个 xgboost 模型的表现相比于基准模型就有很大提升，但也伴随着较为严重的过拟合问题，于是想到用模型调参和模型融合的技术来降低过拟合的影响，模型融合的过程中就涉及到选择哪些模型容易进行的问题，需要进行很多的尝试，每次尝试之后，如果看到模型表现有了一定的提升，即使提升不是很显著，也是很有意思的一件事。模型最终的表现也比较符合预期。

5.3 项目的思考

本项目可以完善和尝试的地方有很多，下面举出一些：

1. 在模型调参和模型融合部分，会发现这些技术对模型的提升效果不是太明显，所以可以返回到特征提取部分， 尝试再提取一些特征，看看对模型的提升效果如何，或许这样的尝试拖入产出比更高。
2. 在模型方面，除了 xgboost 模型，还可以尝试使用其他的技术，比如深度学习，这个比赛的第三名就是使用的深度学习方法。

附：参考文献

[1] Kaggle Rossmann Store Sales Overview:

https://www.kaggle.com/c/rossmann-store-sales

[2] Introduction to Boosted Trees

https://xgboost.readthedocs.io/en/latest/tutorials/model.html

[3] First place interview for this competition

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