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：

y_i

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

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 表□单个门店单天的销售额， $\hat{y}_i$ 表□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□百分□误差可以有效降低□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□或者某些特殊□□

的销售额肯定比平常要高出不少，如果采□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

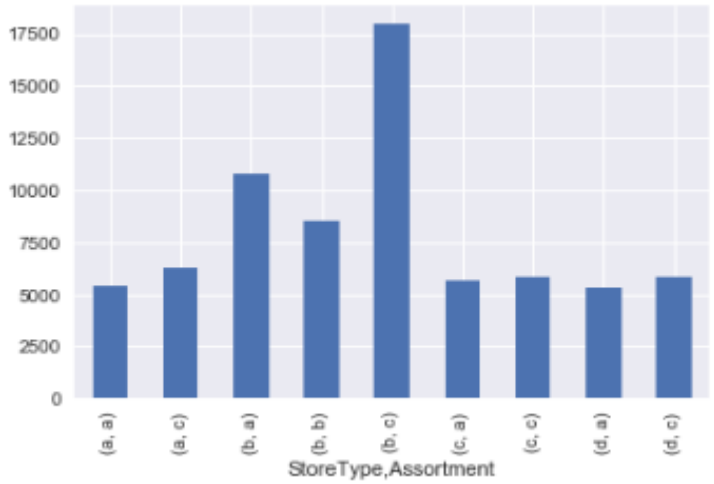
train.head().append(train.tail())

	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

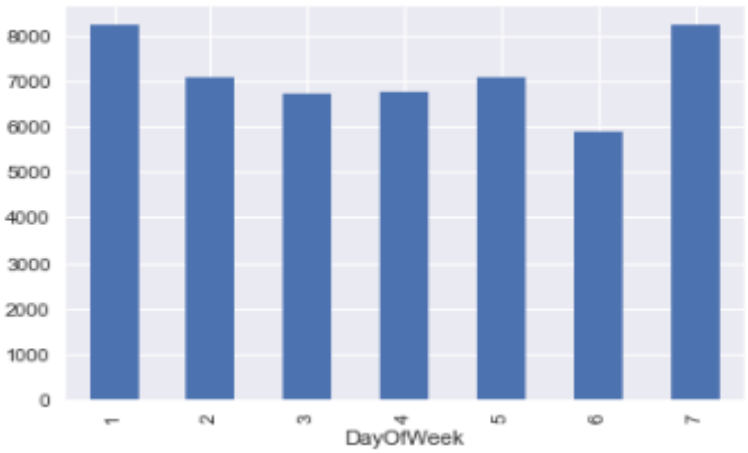
对于Store1做一个随着时间的可视化分析如下，可见要预测的标签是根据

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

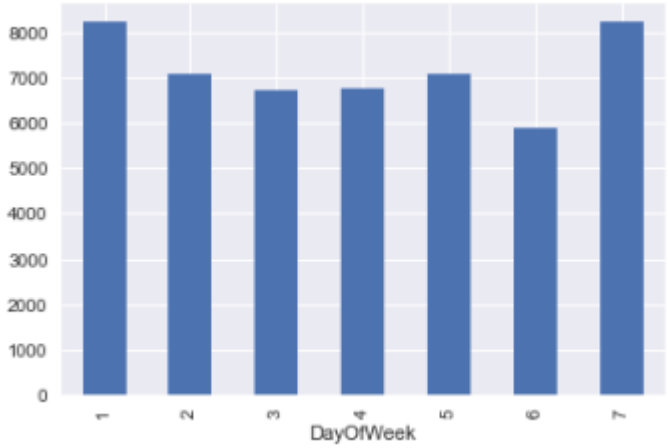


DayofWeek的平均客户数量，与平均销量有密切的关系。

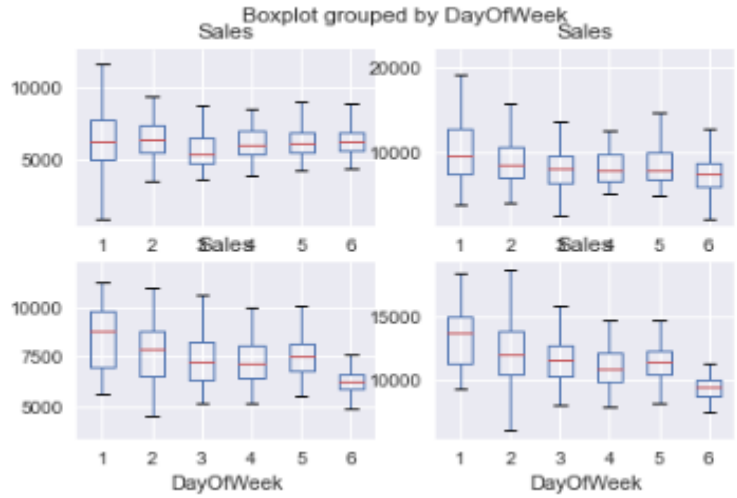
```
#Mean Customers per week

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

dow.plot('bar')
```



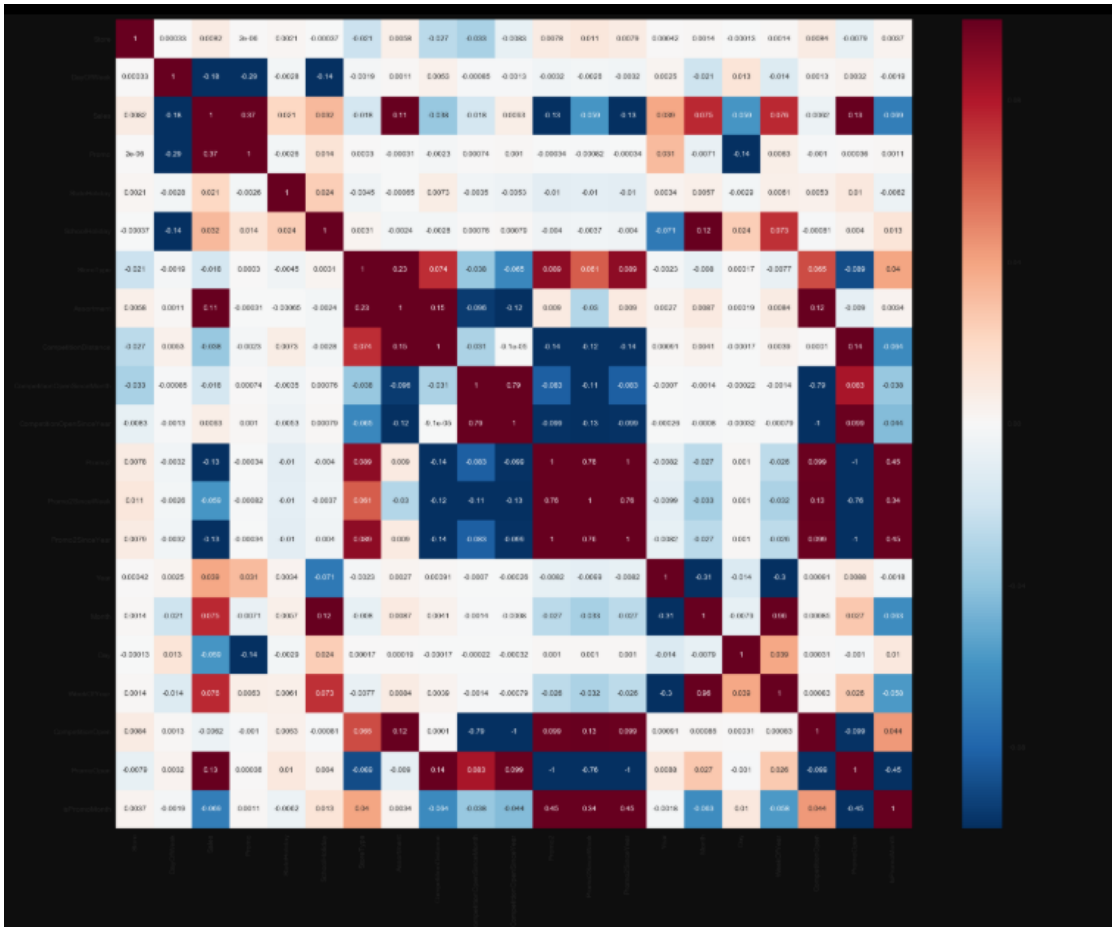
取四个店铺看平均销量，DayofWeek的销量差别很大：



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□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□ 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，□xgboost 相□与其他 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□对线性回归模型和 Gradient Boosting□法做□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□个通过属性的线性组合来进□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□最佳的拟合直线在因变量（销售额）和特征（自变量）之间建□□种关系。□最佳拟合直线的求解是通过最□□乘法来完成，在线性回归中，最□□乘法就是试图 找到□条直线，使所有样本到直线上的欧□距离之和最□。但是基于最□□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□常敏感，异常值会严重影响回归线，最终影响预测值。

2.2.2 Gradient Boosting

Gradient Boosting（梯度提升）是□种集成弱学习模型的机器学习□法。机器学习模型主要的目标是得到□个模型，使得预测值与真实值之间的误差尽可能小，Gradient Boosting 采取分层学习的方法，通过 m 个步骤来得到最终模型，其中第 m 步学习□个较弱的模型 Fm，在第 m+1 步时，不直接优化 Fm，而是学习□个基本模型 h(x)，使得其拟合残差项 y-Fm，这样就会使 m+1 步的模型预测更接近于真实值变成了如何找到 h(x)=Fm+1-Fm，最终就是要找到某类函数空间中的

$$F(x) = \sum_{i=1}^{M} \gamma_i h_i(x) + const$$

□组 h(x)使得 。

Gradient Boosting 算法中最典型的基学习器是决策树。Gradient Boosting Decision Tree(GBDT)就是□种结合决策树的 Gradient Boosting 算法实现。这□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□般都较□。另外□种很常用的 Gradient Boosting 算法的实现是 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□到□阶导数信息，xgboost 则对代价函数进□了□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□过拟合。xgboost 在进行完□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。

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

```
#查看数据缺失值
display(train.isnull().sum(),test.isnull().sum(),store.isnull().sum())
```

test 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。

```
test.fillna(1, inplace=True)
```

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。

将商店类型，品类和是否国家节日转化为数字。将日期拆分为年月日。'PromoInterval'特征转化为'IsPromoMonth'，表明是否处于促销月。

3.2 执行过程

#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)
watchlist = [(dtrain, 'train'), (dvalid, 'eval')]
```

经过模型的训练，得到如下结果：

Train RMSPE	**Validation RMSPE**
0.070345	***0.127409***

前五行数据的结果分析：

```
res = pd.DataFrame(data = ho_ytest)
res['Prediction']=yhat
res = pd.merge(ho_xtest,res, left_index= True, right_index=True)
res['Ratio'] = res.Prediction/res.Sales
res['Error'] =abs(res.Ratio-1)
res['Weight'] = res.Sales/res.Prediction
res.head()
```

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。

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。

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，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 Rossmann_submission_1.csv，Rossmann_submission_2.csv，Rossmann_submission_3.csv。

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。

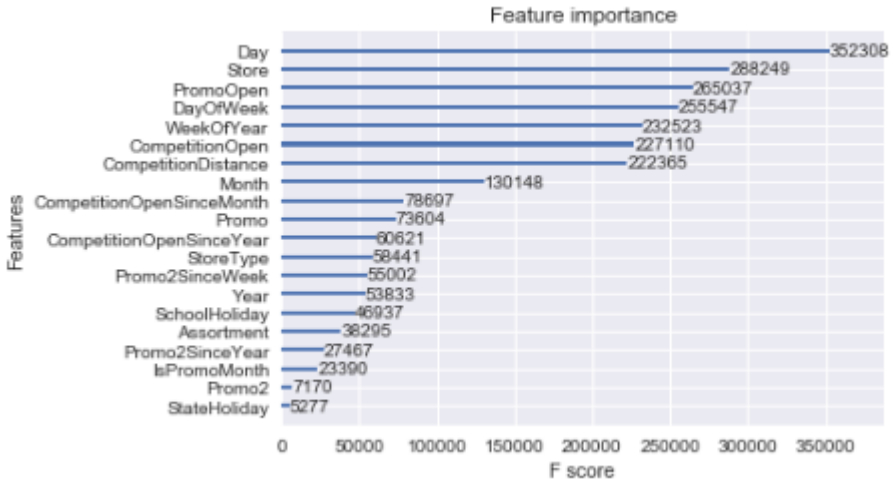
	Store	Ratio	Error	Store_new	Ratio_new	Error_new
264207	292	1.235983	0.235983	292	1.221274	0.221274
711449	782	1.184041	0.184041	782	1.172604	0.172604
827582	909	1.167560	0.167560	909	1.155518	0.155518
827591	909	0.862041	0.137959	909	0.852097	0.147903
797965	876	0.867589	0.132411	876	0.853161	0.146839
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。

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□ 较多的精□,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□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□有很多，下面举出□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□ 些特征，看看对模型的提升效果如何，或许这样的尝试投□ 产出□ 更□ ；
2. 在模型□ □ ，除了 xgboost 模型，还可以尝试使□ 其他的技术，□ 如深度学习，这个□ 赛的第三名就是使□ 的深度学习□ 法。