Forecast Rossmann Store Sales

I. 问题的定义

项目概述

Rossmann 在 7 个欧洲国家经营着 3000 多家药店。目前，Rossmann 门店经理的任务是提前 6 周预测他们的日常销售。商店的销售受到许多因素的影响，包括促销、竞争、学校和国家假日、季节性和地区。由于成千上万的个人经理根据自己的特殊情况预测销售情况，结果的准确性可能会非常不同。

在他们的第一个 Kaggle 竞赛中，Rossmann 挑战通过在德国各地的 1,115 家商店的每日销售额来预测未来 6 周的销售额。可靠的销售预测使商店经理能够制定有效的员工时间表，提高生产力和积极性。通过帮助 Rossmann 创建一个健壮的预测模型，您将帮助商店经理专注于他们最重要的事情:他们的客户和他们的团队

问题陈述

Rossmann 是欧洲的一家连锁药店。 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。

评价指标

提交的内容是根据均方根百分比误差(RMSPE)计算的。RMSPE 被计算为

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

其中 yi 表示单个商店在一天内的销售额，yhat_i 表示相应的预测。任何一天和商店 0 的销售都被忽视得分。

II. 分析

数据的探索与可视化

输入的数据有两部分：

1. Rossmann 商店信息数据：
    1. Store（商店 ID）
    2. DayOfWeek（日期所在的周几）
    3. Date（销售时间）
    4. Sales（销售额）
    5. Customers（消费者数量）
    6. Open（商店当日是否开放）
    7. Promo（商店当日是否促销）
    8. StateHoliday（国家节假日，a、b、c 是国家节假日，0 表示非节假日）
    9. SchoolHoliday（国家公立学校节假日，1：节日，0 非节日）
2. 商店销售数据。
    1. Store（商店 ID）
    2. StoreType（商店类型）
    3. Assortment（商店级别）
    4. CompetitionDistance（竞争对手距离）
    5. CompetitionOpenSinceMonth（竞争对手开店月份，部分缺失）
    6. CompetitionOpenSinceYear（竞争对手开店年份，部分缺失）
    7. Promo2（是否持续促销）
    8. Promo2SinceWeek（持续促销的周）
    9. Promo2SinceYear（持续促销的年）
    10. PromoInterval（促销的月份）
3. 缺失数据

```
RangeIndex: 1115 entries, 0 to 1114
Data columns (total 10 columns):
Store                        1115 non-null int64
StoreType                    1115 non-null object
Assortment                   1115 non-null object
CompetitionDistance          1112 non-null float64
CompetitionOpenSinceMonth    761 non-null float64
CompetitionOpenSinceYear     761 non-null float64
Promo2                       1115 non-null int64
Promo2SinceWeek              571 non-null float64
Promo2SinceYear              571 non-null float64
PromoInterval                571 non-null object
dtypes: float64(5), int64(2), object(3)
memory usage: 87.2+ KB
```

图 1

如图 1 所示 CompetitionDistance 有三家数据缺失，Promo2 促销日有 571 天，竞争对手开店时间大量缺失,缺失 206 条数据，如图 2 所示。

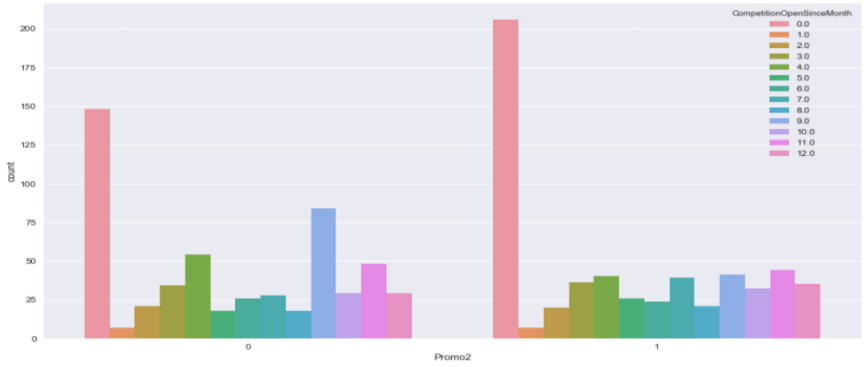

图 2

通过对一周数据探索发现第七天绝大部分店铺都没有开门，而很少的店铺开门，说明这第七天开门的数据可以算是异常数据。

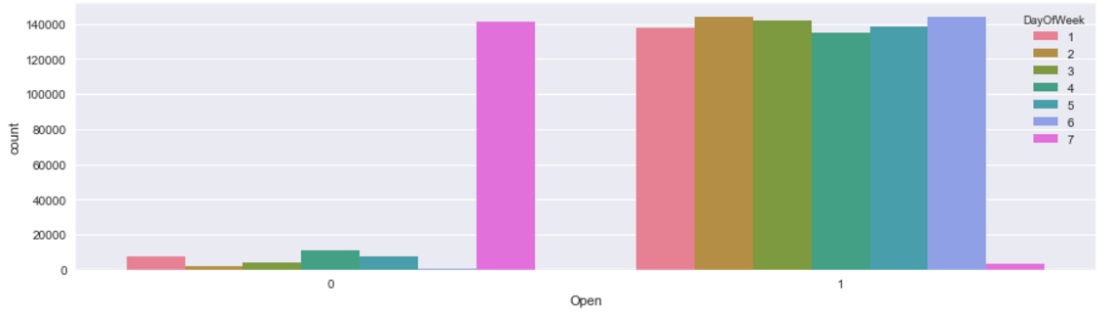

图 3

查看每月平均销售和销售百分比发现到 3、12 月份平均销售是个高峰，而这个时候一般是节日比较多。

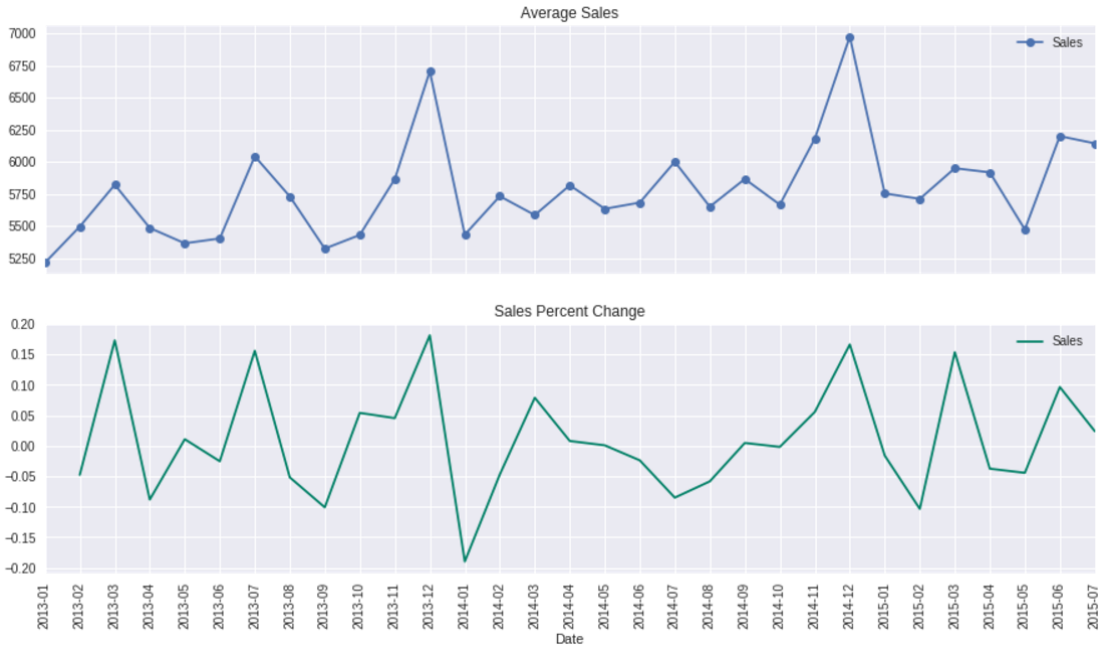

图 4

根据图 5 中可以看出销售额和用户数在促销和非促销的对比情况，促销状况下销售额和用户数都比较多。

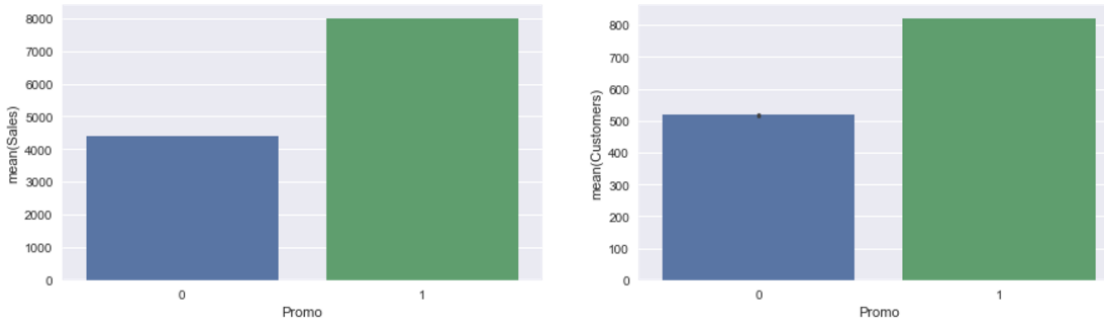

图 5

节假日和非节假日对比可以看出非节假日的平均销售额多，并且三种节日类型的销售额均值相差不大如图 6 所示。

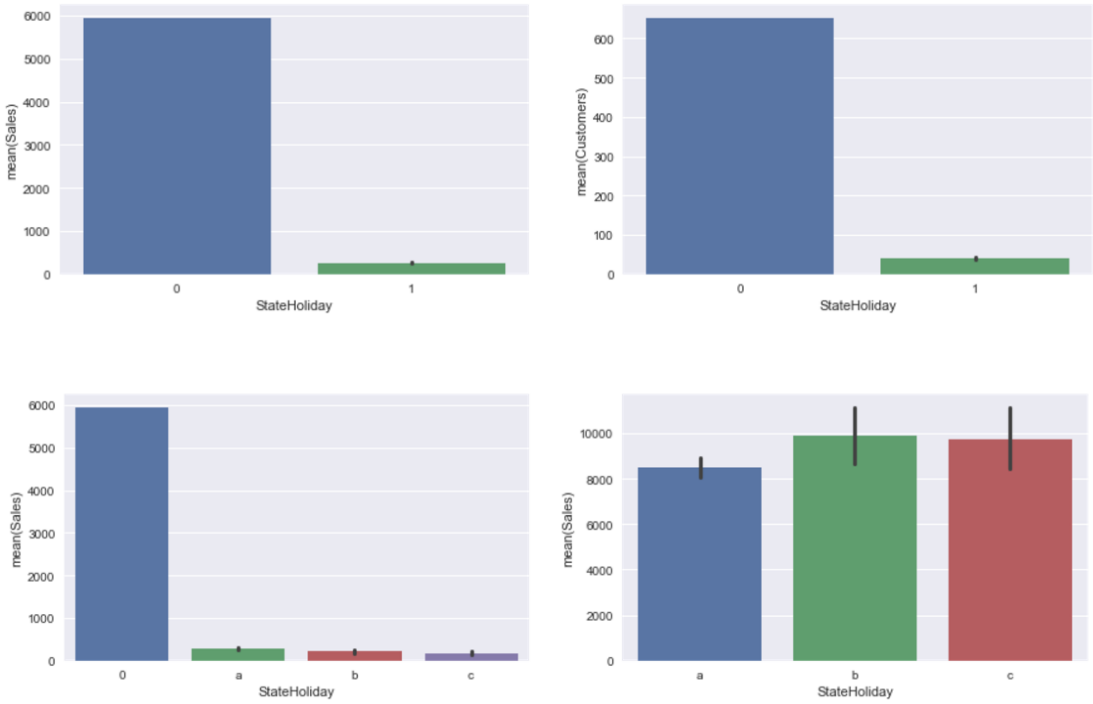

图 6

根据商店类型的统计量和各个商店类型的平均销售额和平均用户数，发现 b 类型商店数量少，但是销售额和用户数多如图 7 所示。

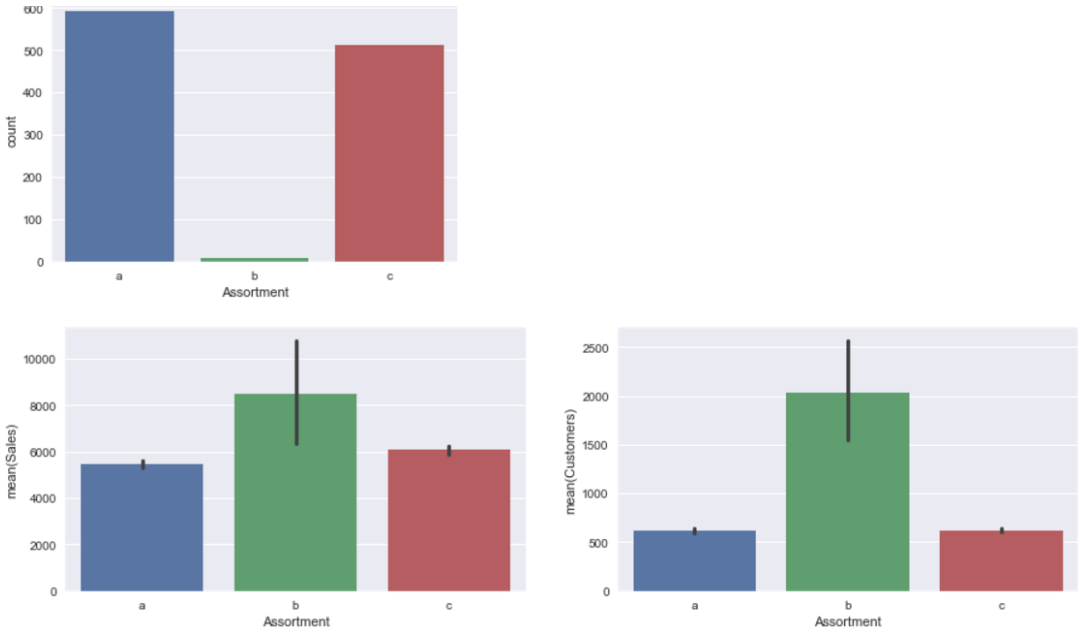

图 7

从竞争对手的距离可以可以发现销售额越高，竞争对手越近，并且竞争对手大部分集中在 2000~8000 距离之间如图 8 所示：

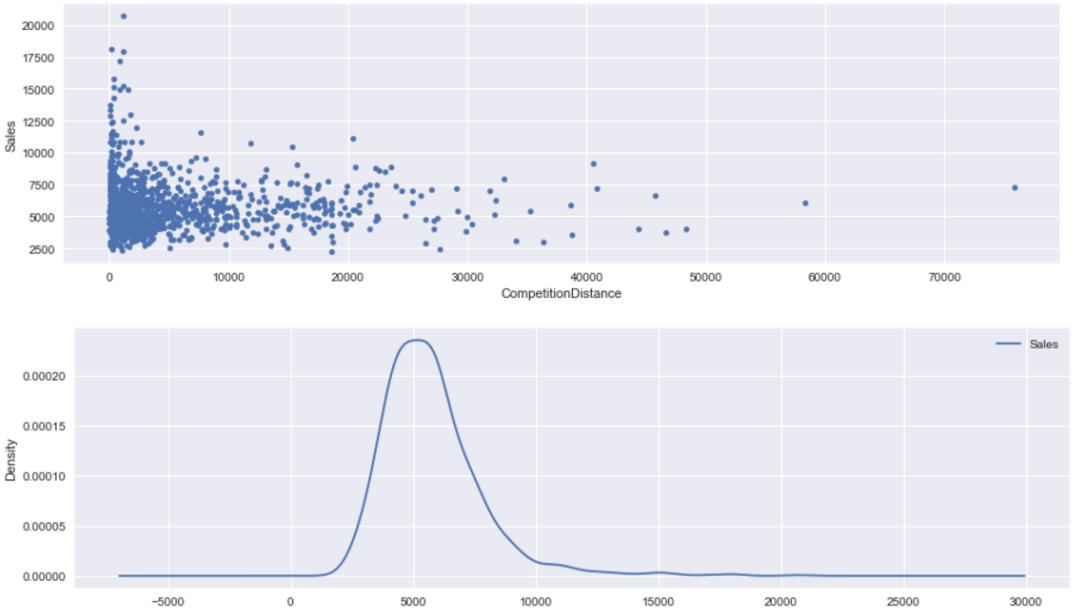

图 8

通过对前 5 个商店进行相关性分析如图 9 所示发现商店之间的相关性很大。

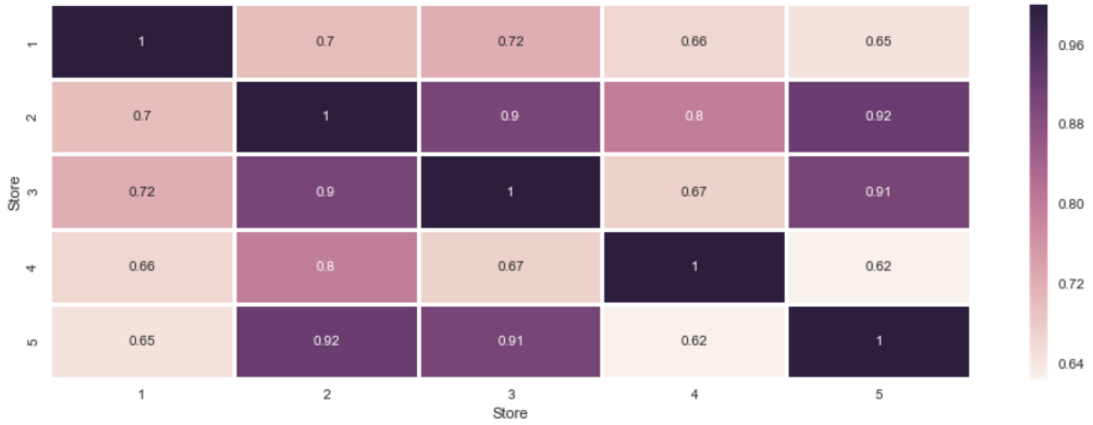

图 9

算法和技术

为了对数据建模来实现对商店销售额的准确的预测需要考虑以下三个方面因素：

1. 理论模型
2. 实际数据
3. 系统的实现

（1） 站在理论模型的角度

一个机器学习模型想要取得好的效果，这个模型需要满足以下两个条件：

1. 模型在我们的训练数据上的表现要不错：也就是 trainning error 要足够小。
2. 模型的 vc-dimension 要低。换句话说，就是模型的自由度不能太大，以防 overfit.(vc dimension 的值越大，参数的个数越多，模型越复杂)

1.对于 LR 这样的模型。如果 underfit，我们可以通过加 feature，或者通过高次的特征转换来使得我们的模型在训练数据上取得足够高的正确率。而对于 tree-enseble 来说，我们解决这一问题的方法是通过训练更多的“弱弱”的 tree. 所以，这两类模型都可以把 training error 做的足够低，也就是说模型的表达能力都是足够的。

2.在 tree-ensemble 模型中，通过加 tree 的方式，对于模型的 vc-dimension 的改变是比较小的。而在 LR 中，初始的维数设定，或者说特征的高次转换对于 vc-dimension 的影响都是更大的。所以，一不小心我们设定的多项式维数高了，模型就“刹不住车了”。这也就是我们之前说的，tree-ensemble 模型的可控性更好，也即更不容易 overfit。

（2）站在数据的角度

1.kaggle 比赛选择的都是真实世界中的问题。所以数据多多少少都是有噪音的。而基于树的算法通常抗噪能力更强。

2.除了数据噪音之外，feature 的多样性也是 tree-ensemble 模型能够取得更好效果的原因之一。特征的多样性也正是为什么工业界很少去使用 svm 的一个重要原因之一（因为会过拟合 overfitting），因为 svm 本质上是属于一个几何模型，这个模型需要去定义 instance 之间的 kernel 或者 similarity。

噪声问题：学习了天鹅的外形（全局特征）之后，你的例子都是白色的。计算机看见黑天鹅就认为不是天鹅。（黑色只是局部特征）

过拟合：把学习太彻底了，所有样本都学了，包括局部特征，所以识别新样本没有几个是对的。

解决方法：不要那么彻底，降低机器学习局部特征和错误特征机率。收集多样化样本，简化模型，交叉验证。

（3）站在系统实现的角度

除了有合适的模型和数据，一个良好的机器学习系统实现往往也是算法最终能否取得好的效果的关键。一个好的机器学习系统实现应该具备以下特征：

1. 正确高效的实现某种模型。我真的见过有些机器学习的库实现某种算法是错误的。而高效的实现意味着可以快速验证不同的模型和参数。

xgboost 高效的 c++实现能够通常能够比其它机器学习库更快的完成训练任务。

2. 系统具有灵活、深度的定制功能。

在灵活性方面，xgboost 可以深度定制每一个子分类器，并且可以灵活的选择 loss function（logistic，linear，softmax 等等）。

3. 系统简单易用。

xgboost 提供了各种语言的封装，使得不同语言的用户都可以使用这个优秀的系统。

4. 系统具有可扩展性, 可以从容处理更大的数据。

xgboost 提供了分布式训练（底层采用 rabit 接口），并且其分布式版本可以跑在各种平台之上，例如 mpi,yarn,spark 等等。

xgboost 能自动利用 cpu 的多线程，而且适当改进了 gradient boosting，加了剪枝，控制了模型的复杂程度

XGBoost 的建模思路：就是在每轮迭代中生成一棵新的回归树，并综合所有回归树的结果，使预测值越来越逼近真实值。

提到随机森林，就不得不提 Bagging，Bagging 可以简单的理解为：放回抽样，多数表决（分类）或简单平均（回归）,同时 Bagging 的基学习器之间属于并列生成，不存在强依赖关系。

Random Forest（随机森林）是 Bagging 的扩展变体，它在以决策树 为基学习器构建 Bagging 集成的基础上，进一步在决策树的训练过程中引入了随机特征选择，因此可以概括 RF 包括四个部分：1、随机选择样本（放回抽样）；2、随机选择特征；3、构建决策树；4、随机森林投票（平均）。

随机选择样本和 Bagging 相同，随机选择特征是指在树的构建中，会从样本集的特征集合中随机选择部分特征，然后再从这个子集中选择最优的属 性用于划分，这种随机性导致随机森林的偏差会有稍微的增加（相比于单棵不随机树），但是由于随机森林的‘平均’特性，会使得它的方差减小，而且方差的减小补偿了偏差的增大，因此总体而言是更好的模型。在构建决策树的时候，RF 的每棵决策树都最大可能的进行生长而不进行剪枝；在对预测输出进行结合时，RF 通常对分类问题使用简单投票法，回归任务使用简单平均法。

RF 的重要特性是不用对其进行交叉验证或者使用一个独立的测试集获得无偏估计，它可以在内部进行评估，也就是说在生成的过程中可以对误差进行无偏估计，由于每个基学习器只使用了训练集中约 63.2%的样本，剩下约 36.8%的样本可用做验证集来对其泛化性能进行‘包外估计’。

RF 和 Bagging 对比：RF 的起始性能较差，特别当只有一个基学习器时，随着学习器数目增多，随机森林通常会收敛到更低的泛化误差。随机森林的训练效率也会高于 Bagging，因为在单个决策树的构建中，Bagging 使用的是‘确定性’决策树，在选择特征划分结点时，要对所有的特征进行考虑，而随机森林使用的是‘随机性’特征数，只需考虑特征的子集。

随机森林的优点较多，简单总结：

1、在数据集上表现良好，相对于其他算法有较大的优势（训练速度、预测准确度）；

2、能够处理很高维的数据，并且不用特征选择，而且在训练完后，给出特征的重要性；

3、容易做成并行化方法。

RF 的缺点：

在噪声较大的分类或者回归问题上回过拟合。

XGBoost 的性能在 GBDT 上又有一步提升，而其性能也能通过各种比赛管窥一二。坊间对 XGBoost 最大的认知在于其能够自动地运用 CPU 的多线程进行并行计算，同时在算法精度上也进行了精度的提高。

由于 GBDT 在合理的参数设置下，往往要生成一定数量的树才能达到令人满意的准确率，在数据集较复杂时，模型可能需要几千次迭代运算。但是 XGBoost 利用并行的 CPU 更好的解决了这个问题。

其实 XGBoost 和 GBDT 的差别也较大，这一点也同样体现在其性能表现上，详见 XGBoost 与 GBDT 的区别。

3．XGBoost 在代价函数中加入了正则项，用于控制模型的复杂度。从权衡方差偏差来看，它降低了模型的方差，使学习出来的模型更加简单，放置过拟合，这也是 XGBoost 优于传统 GBDT 的一个特性。

4.Shrinkage（缩减），相当于学习速率（XGBoost 中的 eta）。XGBoost 在进行完一次迭代时，会将叶子节点的权值乘上该系数，主要是为了削弱每棵树的影响，让后面有更大的学习空间。（GBDT 也有学习速率）。

5.列抽样。XGBoost 借鉴了随机森林的做法，支持列抽样，不仅防止过 拟合，还能减少计算。

6.对缺失值的处理。对于特征的值有缺失的样本，XGBoost 还可以自动 学习出它的分裂方向。

7.XGBoost 工具支持并行。XGBoost 是一次迭代完才能进行下一次迭代的（第 t 次迭代的代价函数里包含了前面 t-1 次迭代的预测值）。XGBoost 的并行是在特征粒度上的。决策树的学习最耗时的一个步骤就是对特征的值进行排序（因为要确定最佳分割点），XGBoost 在训练之前，预先对数据进行了排序，然后保存为 block 结构，后面的迭代 中重复地使用这个结构，大大减小计算量。这个 block 结构也使得并行成为了可能，在进行节点的分裂时，需要计算每个特征的增益，最终选增益最大的那个特征去做分裂，那么各个特征的增益计算就可以开多线程进行。

基准模型

根据项目背景和问题，为未来 6 周各个门店的销售额进行预测，本项目开始参考 Omar El Gabry 的特征分析方法：

1. 分析不同类型商店的平均销售。
2. 分析一周七天的平均销售额。
3. 分析连续几个月的销售额。
4. 进行促销对比分析。
5. 进行持续促销对比分析。
6. 是否公共节假日对比分析。
7. 通过箱型图查看销售额，用户数分布。

8. 对商店类型进行对比分析。
9. 对商店等级进行对比分析。
10. 查看竞争对手距离分布。
11. 查看用户数分布。
12. 进行商店之间相关性分析。

通过分析发现一周的不同时间、节假日、促销、商店类型等对销售额有影响，通过 one-hot-encoding 编码实现特征工程，在根据 XGboost 算法分类器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回归树，在对每个商店单独建模预测数据，最后通过搜索优化提高模型的准确率。

III. 方法

数据预处理

1. 根据数据可视化的部分对每周第七天的数据进行删除。
2. 由于商店是在营业才能有销售额，所以对删除 open=0 的数据，对测试集中 open=0 的数据赋值为 0。
3. 由于销售额与销售日期有关，所以需要把当日时间的年月日拆分为三个年、月、日变量。
4. 对 PromoInterval 特征对应的促销月份转换到对应时间的月份。
5. 把字符型和浮点型的变量进行 one-hot 转换。

执行过程

根据项目要求可知本项目是建立一个回归预测模型，所以选择目前可靠性最好的几个回归模型线性回归、随机森林树回归、决策树回归、xgboost 回归通过参考 Omar El Gabry 对单个店（商店编号为 8）进行建模进行对比分析如表 1：

模型	R2
LinearRegression	0.53696335977534071
RandomForestRegressor	0.63042511608199558
DecisionTreeRegressor	0.57976807656483886
XGboost	0.72465912880987216

表 1

在对所有商店按商店编号进行建模求 R2 平均可靠值：

模型	R2
LinearRegression	0.246445535257
RandomForestRegressor	0.580155616319
DecisionTreeRegressor	0.50329742448
XGboost	0.626765270346

表 2

最后通过对全量数据进行建模求 R2 值进行比较如表 3 所示：

模型	R2
LinearRegression	-2.6091204389900047
RandomForestRegressor	0.87969411370509409
DecisionTreeRegressor	0.81536821084488542
XGboost	-1.3463269129403068

表 3

综合表 1 表 2 的各个模型对单个商店进行建模的结果得知，XGboost 模型效果最好，在对全量数据进行建模发现随机森林效果最好，由于这四个模型都是没有进行参数优化的模型，所以选择 xgboost 算法和随机森林算法进行参数优化，结果如表 4 所示，xgboost 算法效果最好。

模型	R2
RandomForestRegressor	0.589146638467
XGboost	0.626765270346

表 4

最终采用 xgboost 算法作为该项目最终建模。

完善

IV. 结果

模型的评价与验证

根据上几步的数据的处理、模型的比较以及参数的优化，随机选择一组参数 R2 score 为 0.9071，其中模型中各个参数的权重如图 10 所示，前九个 store 编号、Competition Distance、month、Promo、day 等特征对模型影响非常大。

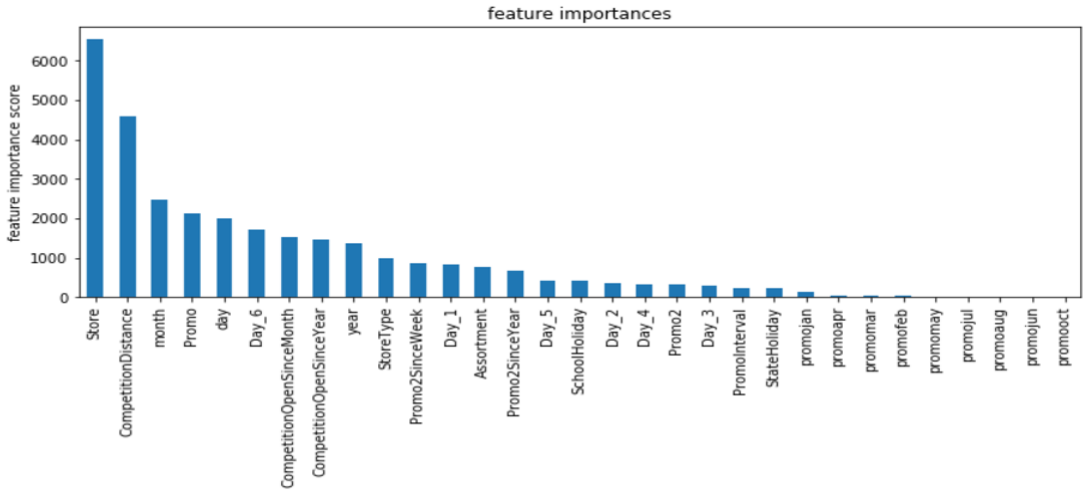

图 10

通过 sikit_learn 中的 GridSearchCV 函数对 xgboost 算法中 max_depth 和 min_child_weight 进行参数搜索，并通过 6 折交叉验证来减少误差。最终模型可靠性为 0.93。各个特征的权重如下图所示：

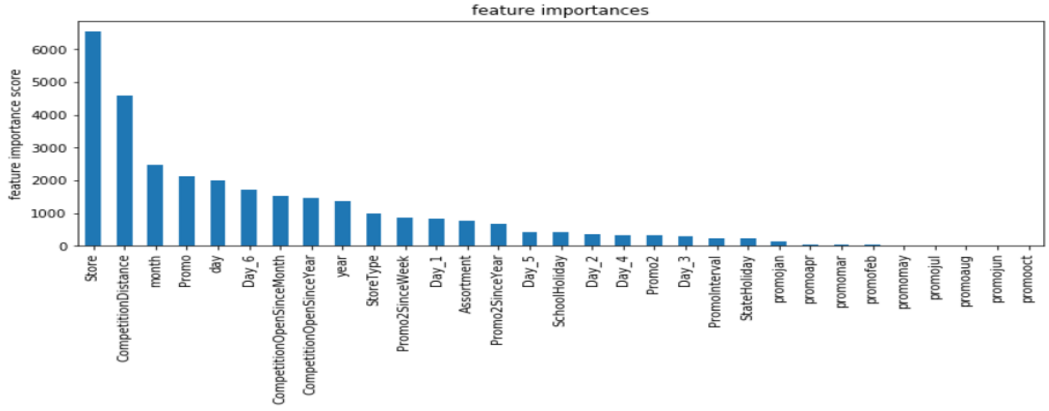

图 12

对参数 gamma 进行调优得到 gamma 等于 0 时模型可靠性为 0.9421

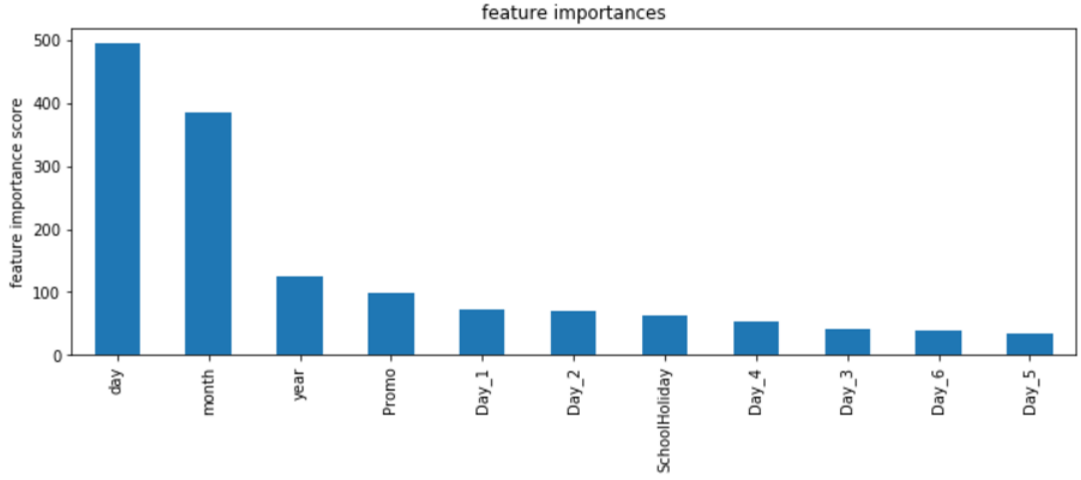

图 13

对 subsample 进行调优得到 subsample 为 0.7 时模型可靠性为 0.9541，其中模型中各个

特征权重如下图：

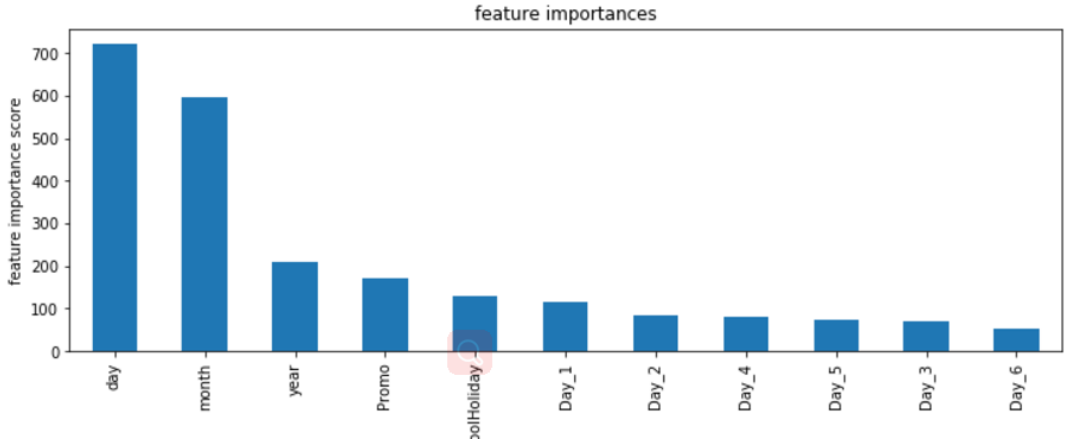

图 14

合理性分析

综合各个参数的优化，发现 max_depth 为 11、min_child_weight 为 5、gamma 为 0、subsample 为 0.7、ntrees 为 8000 时模型最优。

最后 xgboost.trian 函数进行梯度下降来实现整个模型的最优化，并且模型的各个参数权重如图 15 所示：

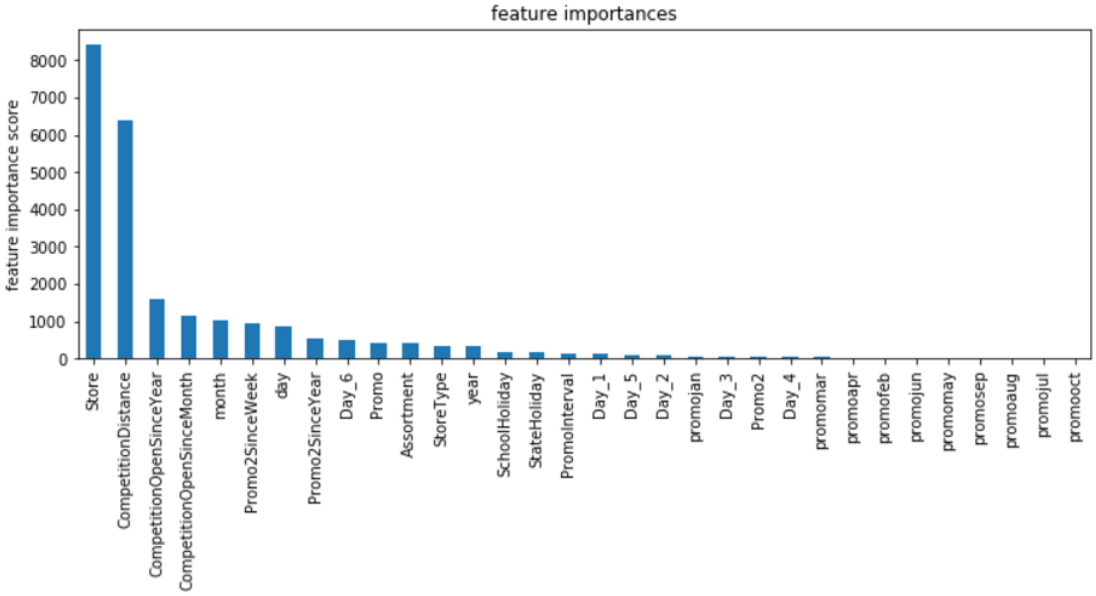

图 15

V. 项目结论

根据以上步骤的结果最终选择对模型影响最大的 13 个特征来进行模型的训练，并且使得模型的误差最小，从特征的重要性来看 store ID 对模型影响非常大说明各个商店区别还是很大的，去除商店 ID 竞争对手的距离和竞争对手的开店的年份和月份对销售额影响很大，

商店如果为了提高销售，可以通过与竞争对手进行差异化竞争来提高销售额。模型中商店的促销对销售额影响很大，说明商店促销是有用的，并且节假日周六的销售额比工作日的销售额多，说明如果进一步提高销售额可以从周日营业开始。

结果可视化

最后对模型的误差平法均值和标准差进行可视化如图 16、17 所示：

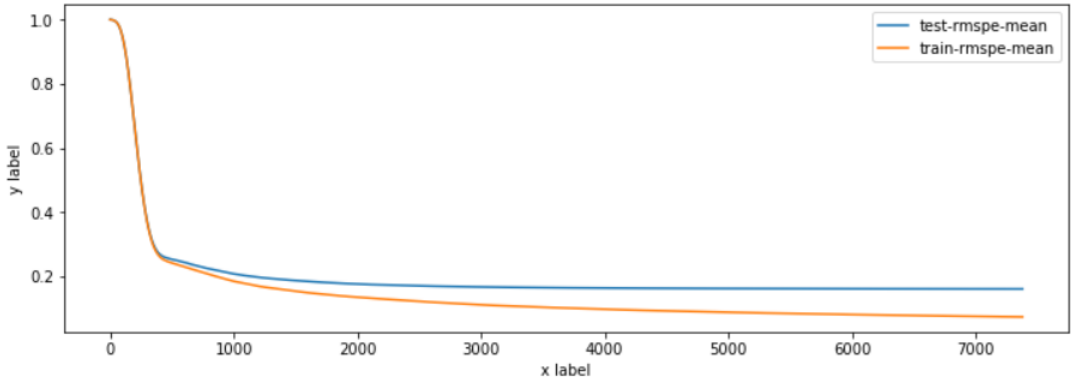

图 16

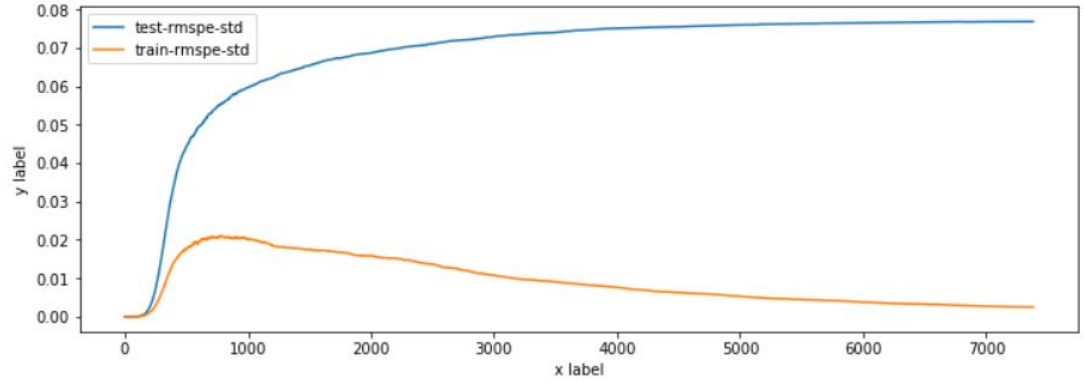

图 17

对项目的思考

首先通过可视化分析来探索数据，通过对商店的各个特征来分析从全部角度来观察商店销售情况，发现异常值、冗余值等来清除与预测无关的数据并且分析其中各个特征与输出特征 y 的值的关系，发现特征间的相关性。

最后发现模型还是有些过拟合，后续可以继续对数据进行探索发现数据中存在的问题，以及改善模型的参数来避免过拟合。

参考文献

[1]https://www.jianshu.com/p/0f96f97c3d5a

[2] https://www.jiqizhixin.com/articles/2017-12-20-2
[3] https://tech.meituan.com/machinelearning-data-feature-process.html