

# 机器学习纳米学位报告

## 毕业项目

---

林叶

2017 年 02 月 22 日

## I. 问题的定义

---

### 项目概述

这个项目为 Kaggle 的一个比赛项目。网址为

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

Rossmann 公司有超过 3000 加分店，遍布欧洲 7 个国家。商店销售受许多因素的影响，包括促销，竞争，学校和州的节假日，季节性和地方性等情况都会影响销售情况的预测。这个项目希望通过帮助 Rossmann 创建一个强大的预测模型帮助更好的管理药品情况。

项目选择数据集是采用 Kaggle 的（Forecast Rossmann Store Sales）项目提供的数据。项目中的 `train.csv` 作为训练数据。使用训练好的模型对 `test.csv` 中的数据进行预测。并可以提交到 Kaggle 中进行模型评估。

### 问题陈述

项目选择 Kaggle 提供的 Rossmann 连锁药店的销售数据集。根据药店提供的历史销售记录以及对药店的一些扩展信息进行分析与预测。原始数据包括内容有：

店 ID, 周几, 日期, 客户数量, 是否开业, 是否有促销, 是否有节假日, 是否是学校假日。通过 store.csv 可以获取商店的基本信息作为扩展数据。

预测需要合理根据上述信息作为输入特征, 对 Sales 进行预测。之后通过训练结果, 将特征信息输入到模型中进行 Sales 的预测。并检验预测的 Sales 与实际销售金额是否接近。

这个项目在 Kaggle 中已经结束。最终获奖者采用的是 xgboost 作为模型进行预测。

获奖模型介绍地址: [https://kaggle2.blob.core.windows.net/forum-message-attachments/102102/3454/Rossmann\\_nr1\\_doc.pdf](https://kaggle2.blob.core.windows.net/forum-message-attachments/102102/3454/Rossmann_nr1_doc.pdf)

## 评价指标

通过 Kaggle 官网的 RMPSE 进行评分。

Kaggle 采用 RMPSE 模型进行评分。RMPSE 是 Root Mean Square

Percentage Error (RMSPE)。计算公式为:

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

其中  $y_i$  表示单个日子上的单个商店的销售额,  $\hat{y}_i$  表示相应的预测。任何有 0 个销售的日子和商店在得分时都会被忽略。

RMPSE 对数据中的极大极小值反应敏感，他可以很好的反应出预测模型的精密密度。对于预测销售量可以很好的表现出模型的预测效果。

## II. 分析

---

### 数据的探索

项目中原始内容为 1017209 条。

Sales 最小值: 0.00

Sales 最大值: 41,551.00

Sales 均值: 5,773.82

Sales 中值 5,744.00

Sales 标准差: 3,849.92

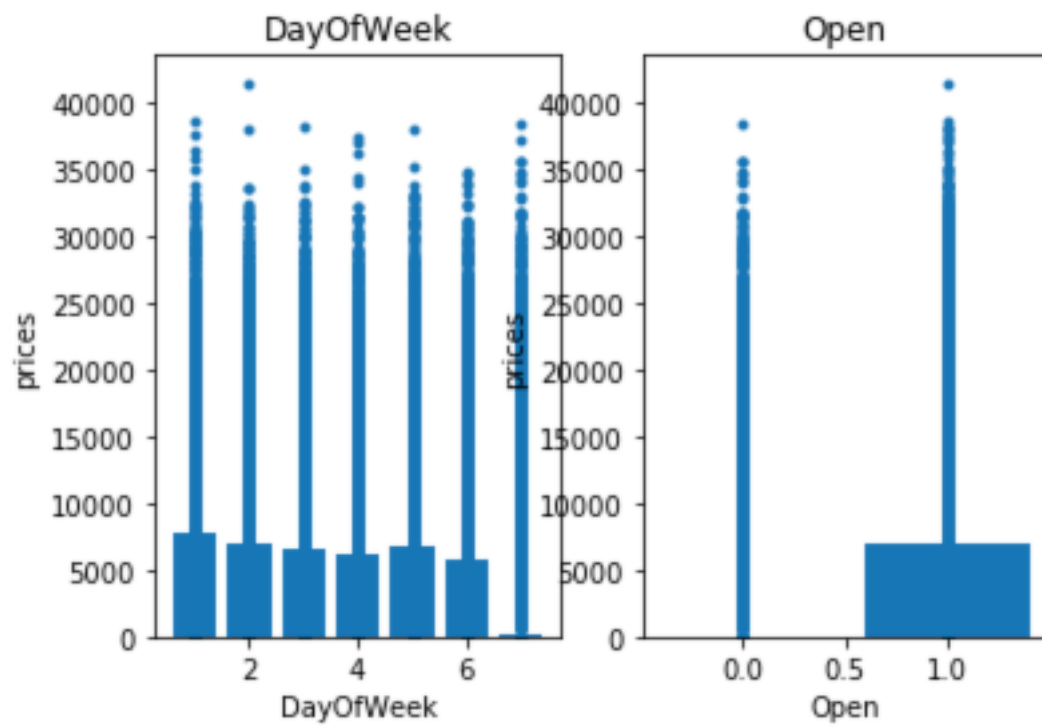
通过对 Sales 的统计，最小值是 0 最大值是 41551。平均值是 5773.82,中值是 5744，标准差是 3849.92。均值与中值非常接近。

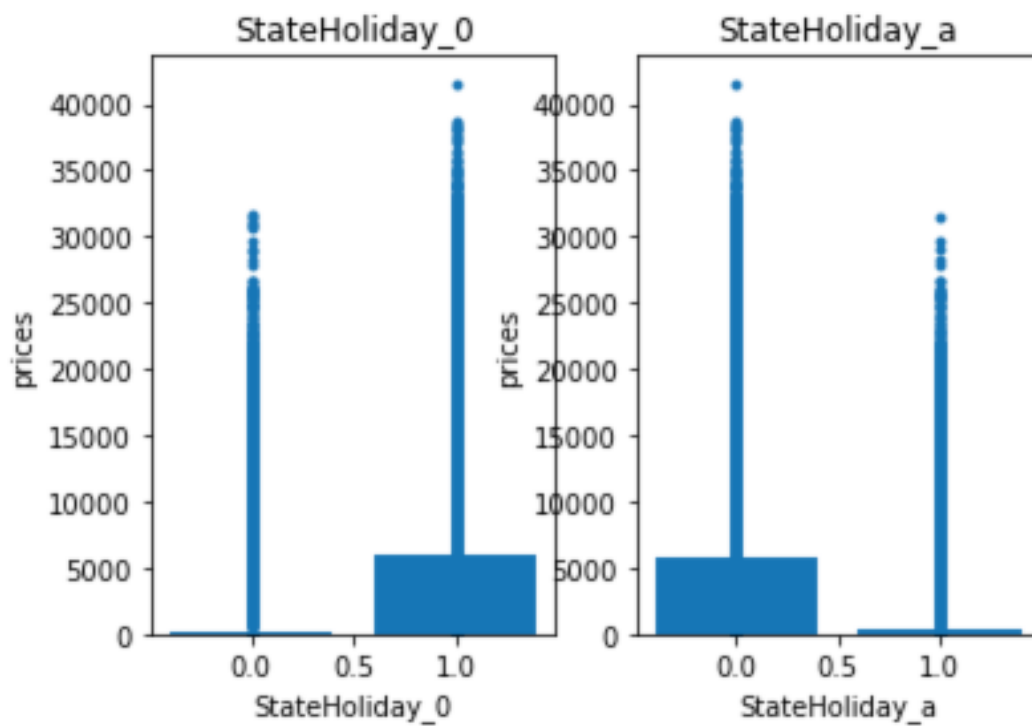
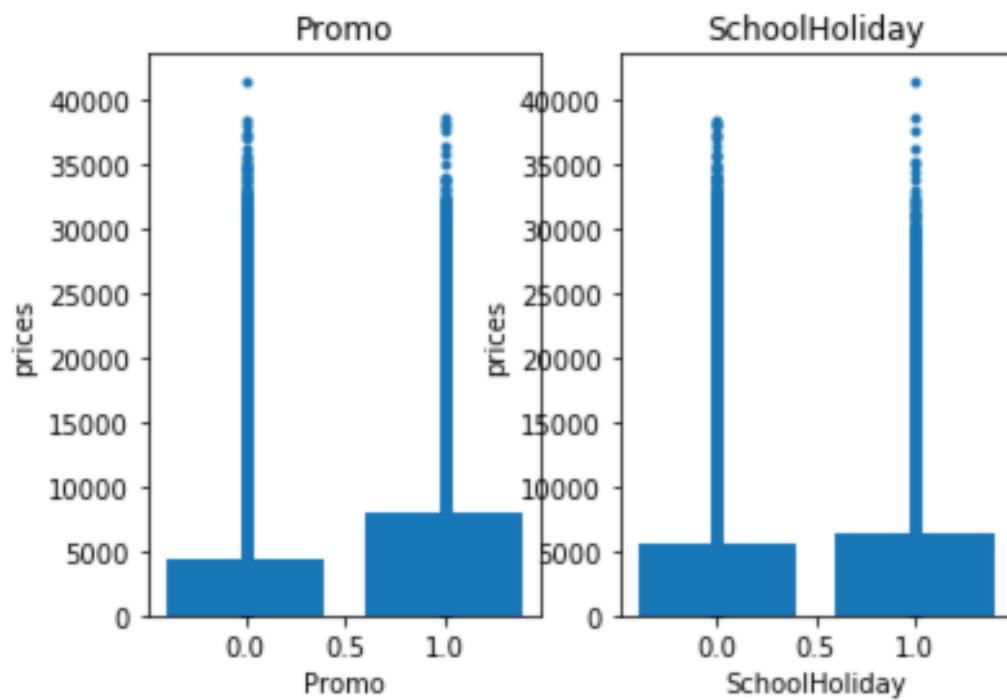
StateHoliday 数据是 0,a,b,c。将数据 a,b,c 转换成 1,2,3。生成根据特征生成图观察数据。

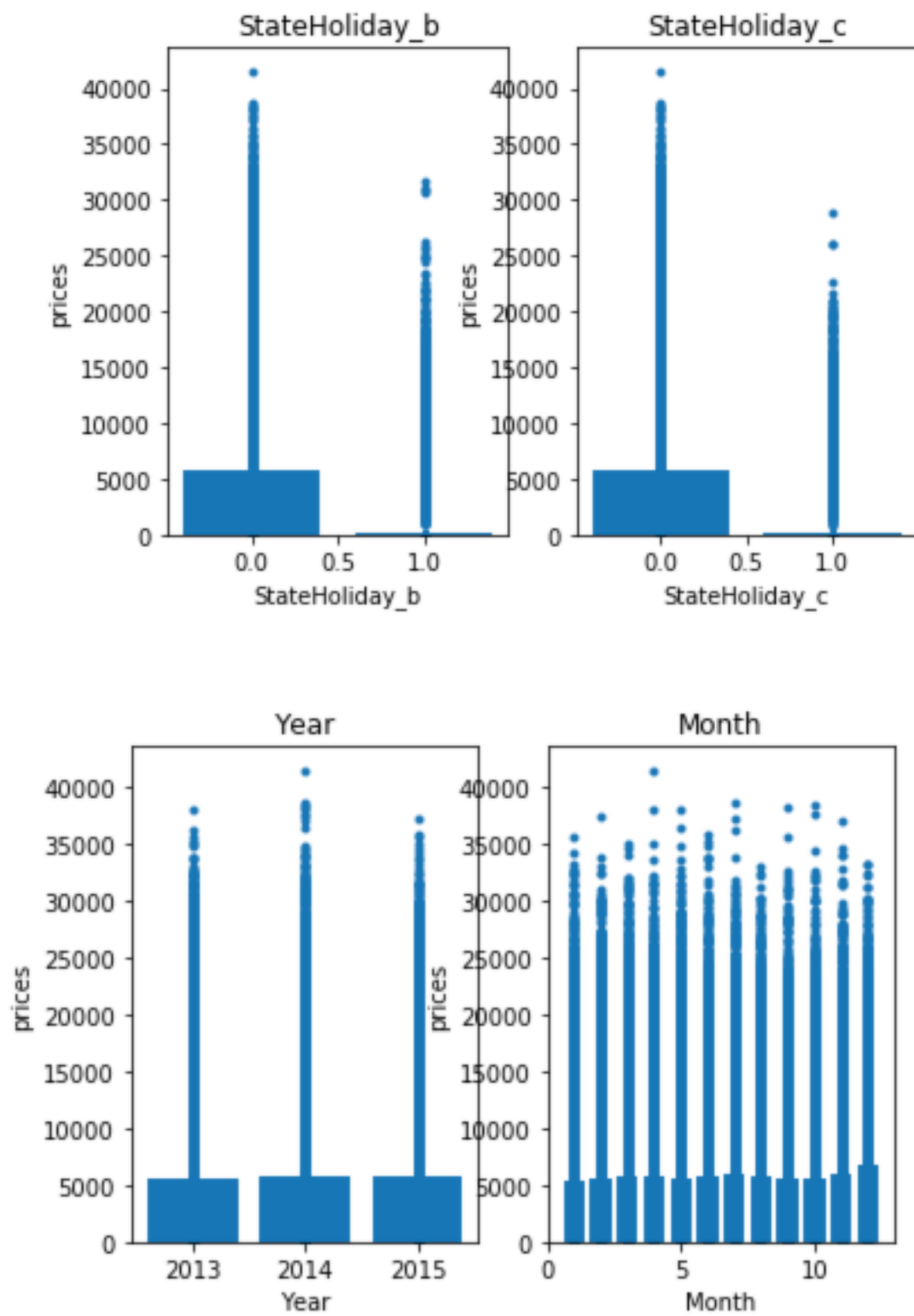
另外观察 store.csv 内容。store 包含信息如下：StoreType、Assortment、CompetitionDistance、CompetitionOpenSinceMonth、CompetitionOpenSinceYear、Promo2、Promo2SinceWeek、Promo2SinceYear、PromoInterval。

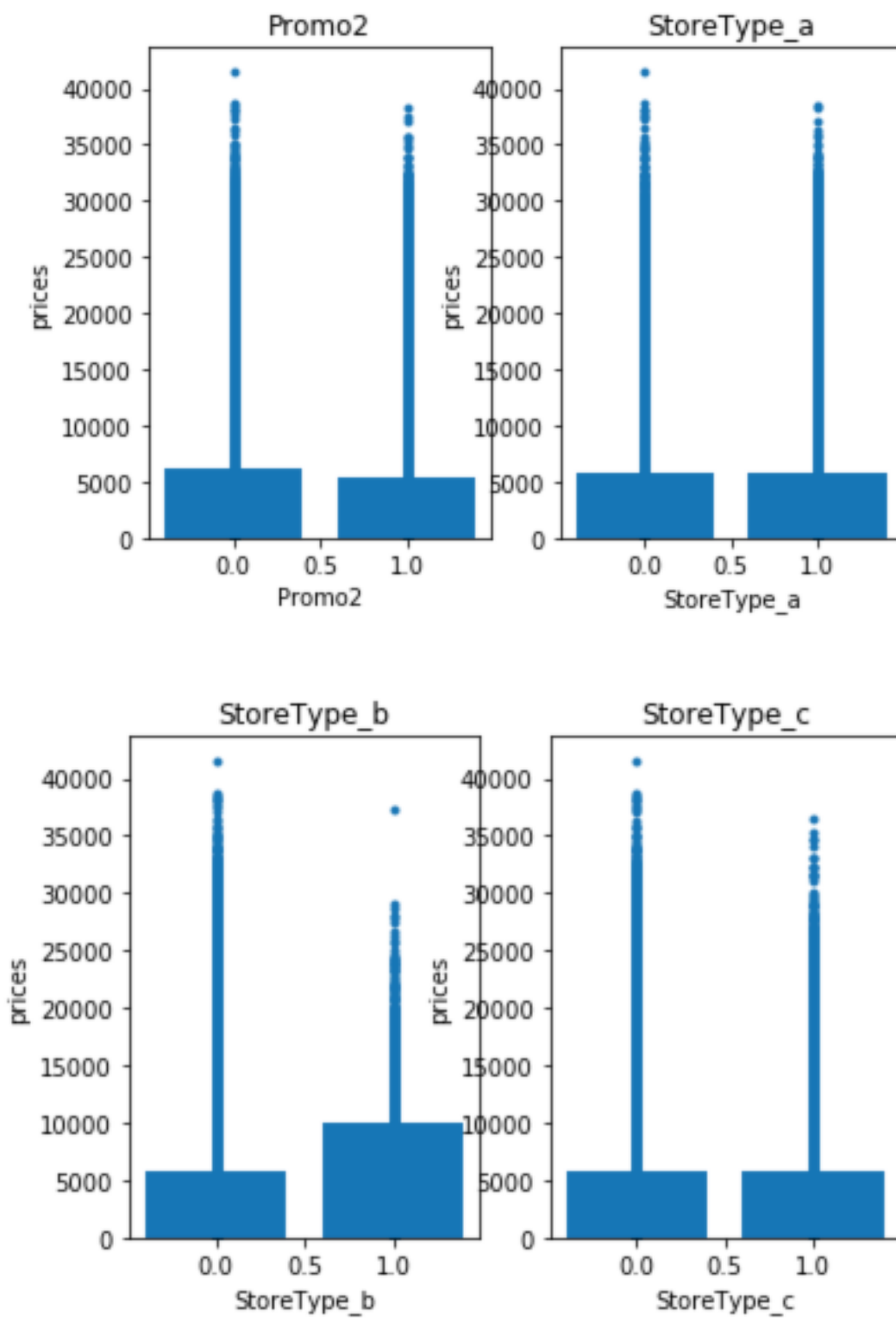
其中 CompetitionOpenSinceMonth, CompetitionOpenSinceYear,  
Promo2SinceYear,Promo2SinceWeek, CompetitionDistance 数据不全。

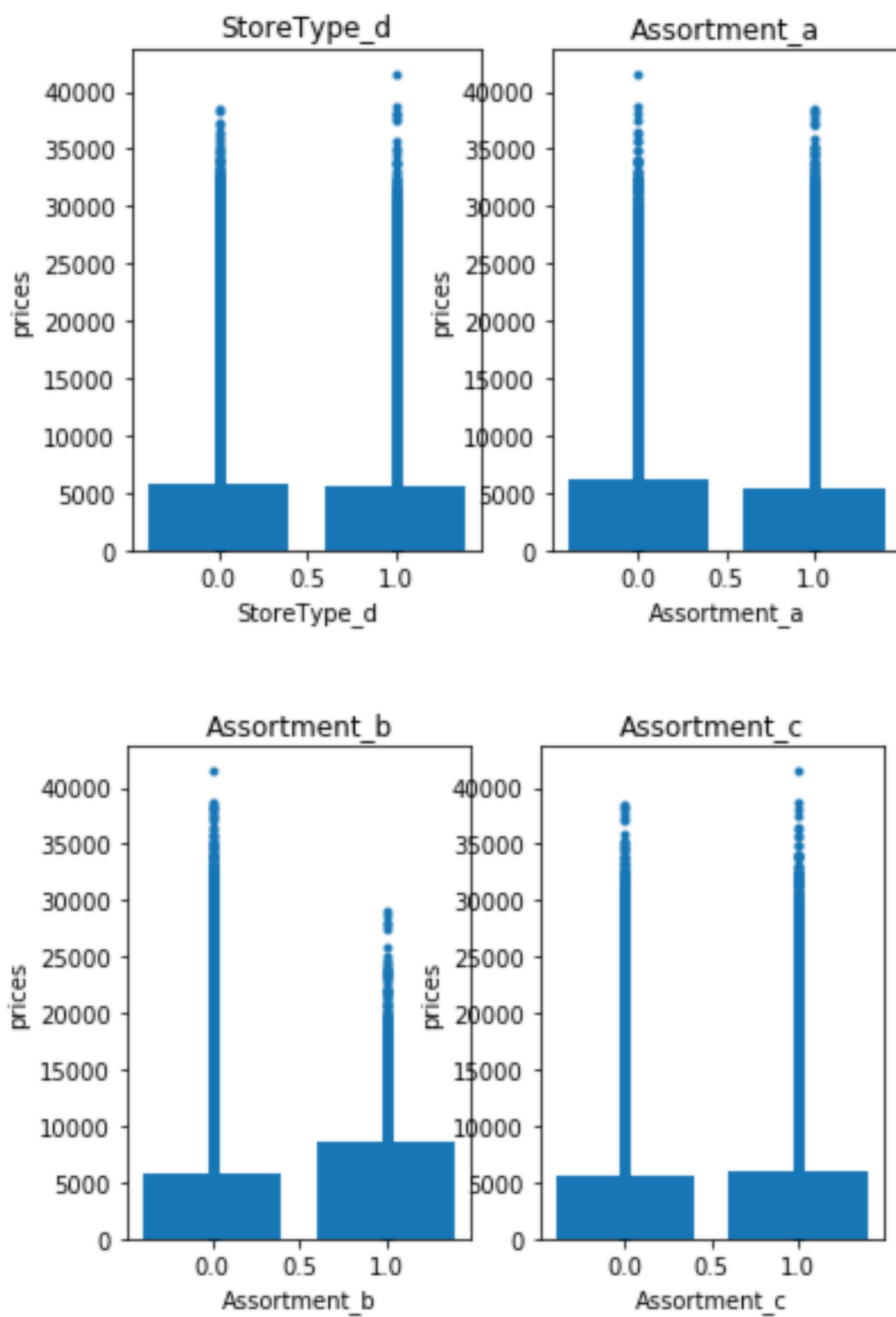
观察各个特征与 prices 的均值与分布情况如下：





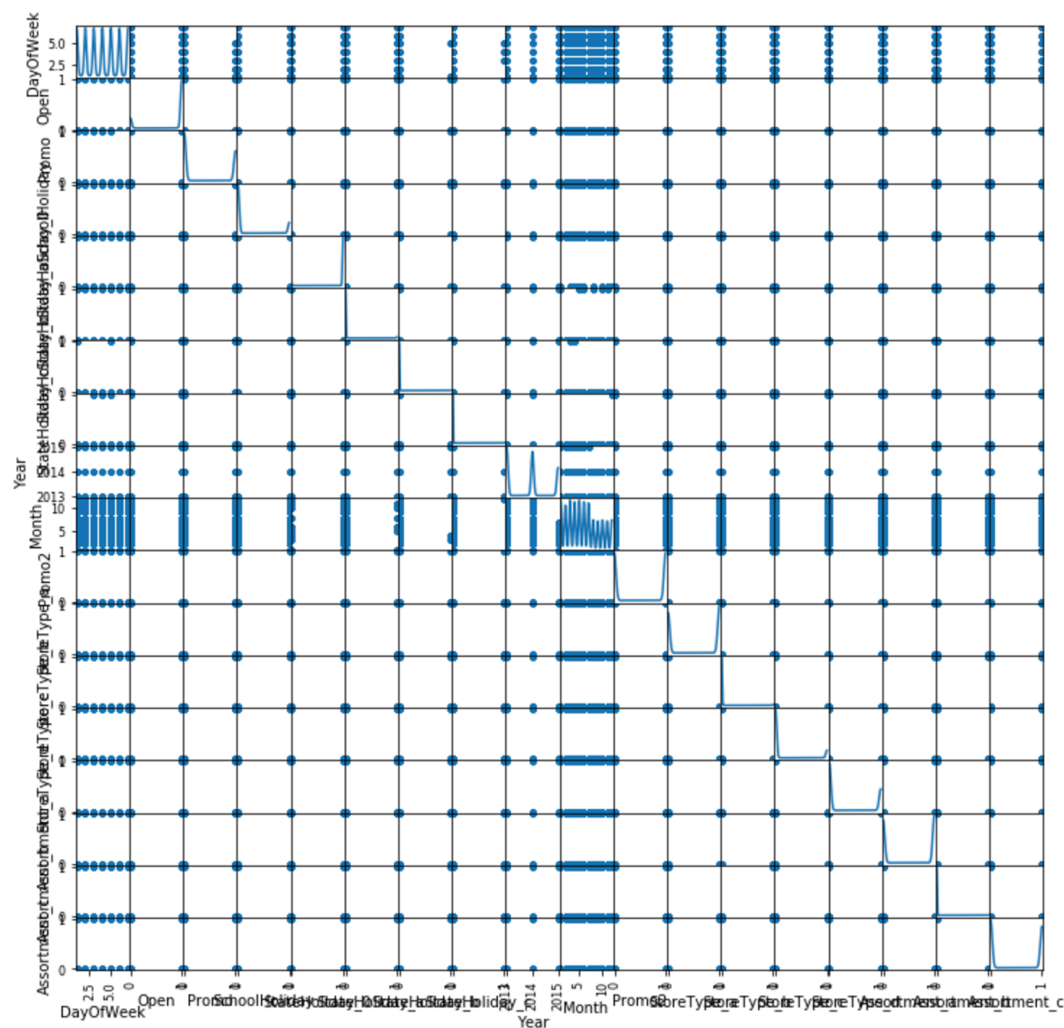






通过均值来看这些特征与结果都有一定的相关性。





通过数据分布可以看出不同特征下都存在一些异常值的分布。我们可以在前面先使用原始数据进行预测，之后使用  $3\sigma$  原则处理异常值，在去除指标的异常值后再次进行预测。查看预测效果。

## 算法和技术

此项目在属于监督学习。

常用的监督学习算法有：回归算法，决策树，神经网络，svm，朴素贝叶斯等算法。

逻辑回归，神经网络，svm，朴素贝叶斯算法多计算用于数据的分类。

数据预测一般是采用线性回归，决策树回归和随机森林这集中算法。

通过上面数据分析查看，可以看出本项目数据是非线性的，所以线性回归算法并不适用。

决策树回归算法与随机森林算法，都有对异常值不敏感的算法。

决策树算法：是一种树形结构，基于特征对目标的分类回归过程。根据某个特征把数据划分到若干子树上，同样再对子树进行递归，直到满足某个条件停止划分，如果不满足一直持续划分。其中 `DecisionTreeRegressor` 主要参数有  
`max_depth`：树的深度。默认是 `None` 不进行限制。

`min_samples_split`：一个内部 Node 的最小采样数量。

`min_samples_leaf`：一个 Leaf Node 的最小采样数量

`max_leaf_nodes`：最大 Leaf nodes 数量

通过这几个参数控制整个树的采样情况，几个参数可能会相互影响。通过几个参数优化控制拟合情况。如减少 `max_depth` 或者加大 `min_samples_split` 和 `min_samples_leaf` 都可以防止过拟合。

随机森林算法：随机森林是一个包含多个决策树的算法，根据特征生成多棵树进行计算。解决决策树泛化能力弱的缺点，这样可以自动忽略没有贡献的特征。随机森林可以计算 `feature importance` 通过 `feature importance` 了解特征重要情况。同样随机森林主要也有上面 4 个优化参数，可以进行优化。

最终采用决策树和随机森林两个算法进行尝试。最终选择效果的好的作为最终模型进行结果进行提交。Xgboost 也是很好的决策树模型，但是目前个人能力有限目前还不能使用，所以并没有这个这个模型。

针对因为两种算法都属于决策树。本项目数据量很大，为了避免过度拟合，主要考虑优化参数 `max_depth`, `min_samples_split`, `min_samples_leaf` 进行优化测试。

## 基准模型

作为预测的基准模型可以使用中值或者均值，这两个值在之前分析中非常接近。分别用这两个值作为结果在 Kaggle 提交测试。

采用所有的 Store 做均值和中值的测试结果分数分别是 0.40815，0.40606。

针对每个 Store 做均值和中值的测试结果分数分别是 0.25278，0.24813。

因为采用 RMPSE 作为评分，分数越少表示错误越少也就是结果更加的准确，对每个 Store 做的中值的评分 0.24813 是最高分。目前检查 test 数据中的 store 在 train 里都存在，所以采用每个 store 做中值的评分 0.24813 作为基准值。

## III. 方法

---

## 数据预处理

Sales 另外和 Date 没有直接关系，所以去除了 Date。另外查看 test.csv 中没有 Customers 数据，所以去除 Customers 数据。

test.csv 中 Open 数据可能是空。

	A	B	C	D	E	F	G	H
	Id	Store	DayOfWeek	Date	Open	Promo	StateHoliday	SchoolHoliday
	480	622	4	2015/9/17		1	0	0
	1336	622	3	2015/9/16		1	0	0
	2192	622	2	2015/9/15		1	0	0
	3048	622	1	2015/9/14		1	0	0
	4760	622	6	2015/9/12		0	0	0
	5616	622	5	2015/9/11		0	0	0
	6472	622	4	2015/9/10		0	0	0
	7328	622	3	2015/9/9		0	0	0
	8184	622	2	2015/9/8		0	0	0
	9040	622	1	2015/9/7		0	0	0
	10752	622	6	2015/9/5		0	0	0

因为 Open 没有准确信息，所以根据 Promo 设置这些数据的 Open，因为 Promo 的时候正常应该是开店。所以 Promo 是 1 就将 Open 设置为 1，Promo 是 0 的时候就将 Open 设置为 0。此部分总共有 11 条数据，而测试数据总量为 41088 条，异常数据量占的比重很小。其实也设置成什么也不会对结果造成大影响，并不用特别在意。

## 执行过程

- 读取 train.csv。
- 整理数据替换非字符指标。
- 生成训练数据。

- 生成多个回归方法进行测试。使用了 RandomForestRegressor、DecisionTreeRegressor。
- 采用搜寻网络进行参数的自动优化。因为两个都是决策树类型的回归方法。主要针对 min\_samples\_split,min\_samples\_leaf 进行优化测试。
- 采用  $3\text{-}\sigma$  原则处理异常值后进行重新训练与网络搜索优化。

## 完善

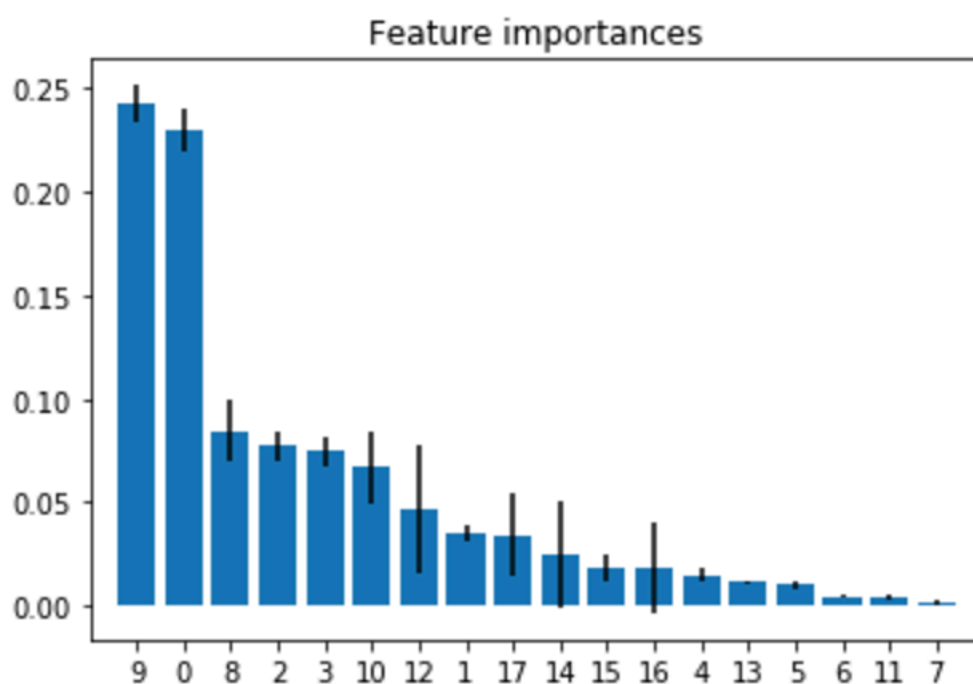
第 1 次训练：拿原始数据测试。训练结果如下。

模型名称	Kaggle 得分	本地 RMSPE 得分
DecisionTreeRegressor	0.47483	0.680670811749
RandomForestRegressor	0.46144	0.69064781293

随机森林 Feature importances:

Feature ranking:

1. feature 9 DayOfWeek (0.242586)
2. feature 0 Open (0.230137)
3. feature 8 Promo (0.084250)
4. feature 2 SchoolHoliday (0.077222)
5. feature 3 StateHoliday\_0 (0.074442)
6. feature 10 StateHoliday\_a (0.067053)
7. feature 12 StateHoliday\_b (0.046904)
8. feature 1 StateHoliday\_c (0.035395)
9. feature 17 Year (0.034141)
10. feature 14 Month (0.024351)
11. feature 15 Promo2 (0.018714)
12. feature 16 StoreType\_a (0.018349)
13. feature 4 StoreType\_b (0.015051)
14. feature 13 StoreType\_c (0.011590)
15. feature 5 StoreType\_d (0.009970)
16. feature 6 Assortment\_a (0.004295)
17. feature 11 Assortment\_b (0.004042)
18. feature 7 Assortment\_c (0.001507)



第2次训练：拿全部数据进行学习训练交叉训练。训练结果如下。

决策树搜索参数是

```
{'min_samples_split':[100,500,1000,2000], 'min_samples_leaf':[100,500,1000,2000], 'random_state':[0]}
```

随机森林分别优化参数是({'n\_jobs':[-

```
1], 'min_samples_split':[1000,2000], 'min_samples_leaf':[1000,2000], 'random_state':[0]})#,'max_depth':[None,1,3,5]
```

GridSearchCV 参数如下：

优化评分为 `scoring_fnc =`

```
make_scorer(performance_metric, greater_is_better=False)
```

交叉测试 `cv=5`。

`performance_metric` 采用 RMPSE 算法。通过这些参数测试不同采样数量与深度的情况下数据的 RMPSE 表现情况自动优化结果。

参数优化结果：

```
DecisionTreeRegressor(criterion='mse', max_depth=None, max_features=None,
```

```
    max_leaf_nodes=None, min_impurity_split=1e-07,
    min_samples_leaf=2000, min_samples_split=100,
    min_weight_fraction_leaf=0.0, presort=False, random_state=0,
    splitter='best')
```

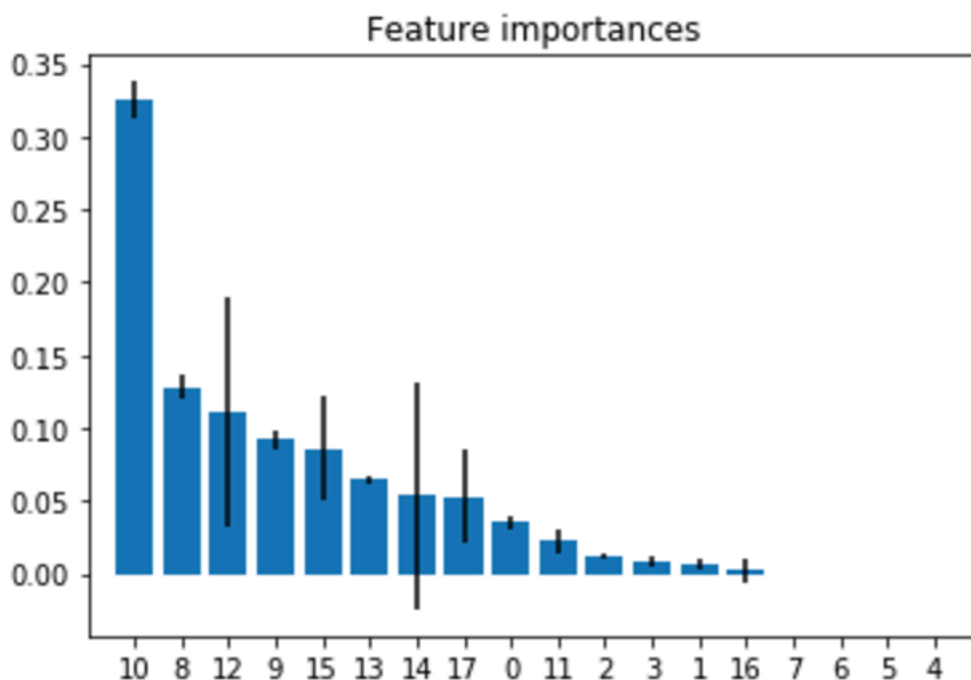
```
RandomForestRegressor(bootstrap=True, criterion='mse', max_depth=None,
    max_features='auto', max_leaf_nodes=None,
    min_impurity_split=1e-07, min_samples_leaf=2000,
    min_samples_split=1000, min_weight_fraction_leaf=0.0,
    n_estimators=10, n_jobs=-1, oob_score=False, random_state=
    0, verbose=0, warm_start=False)
```

模型名称	Kaggle 得分	本地 RMSPE 得分
DecisionTreeRegressor	0.41595	0.668378637568
RandomForestRegressor	0.40901	0.66792283982

随机森林 Feature importances:

Feature ranking:

1. feature 10 DayOfWeek (0.325208)
2. feature 8 Open (0.128111)
3. feature 12 Promo (0.110636)
4. feature 9 SchoolHoliday (0.092240)
5. feature 15 StateHoliday\_0 (0.085691)
6. feature 13 StateHoliday\_a (0.064301)
7. feature 14 StateHoliday\_b (0.053476)
8. feature 17 StateHoliday\_c (0.052877)
9. feature 0 Year (0.035096)
10. feature 11 Month (0.022453)
11. feature 2 Promo2 (0.011994)
12. feature 3 StoreType\_a (0.008468)
13. feature 1 StoreType\_b (0.006737)
14. feature 16 StoreType\_c (0.002711)
15. feature 7 StoreType\_d (0.000000)
16. feature 6 Assortment\_a (0.000000)
17. feature 5 Assortment\_b (0.000000)
18. feature 4 Assortment\_c (0.000000)



第 3 次训练：采用 3- $\sigma$  原则只对 Sales 做异常值后训练结果如下。

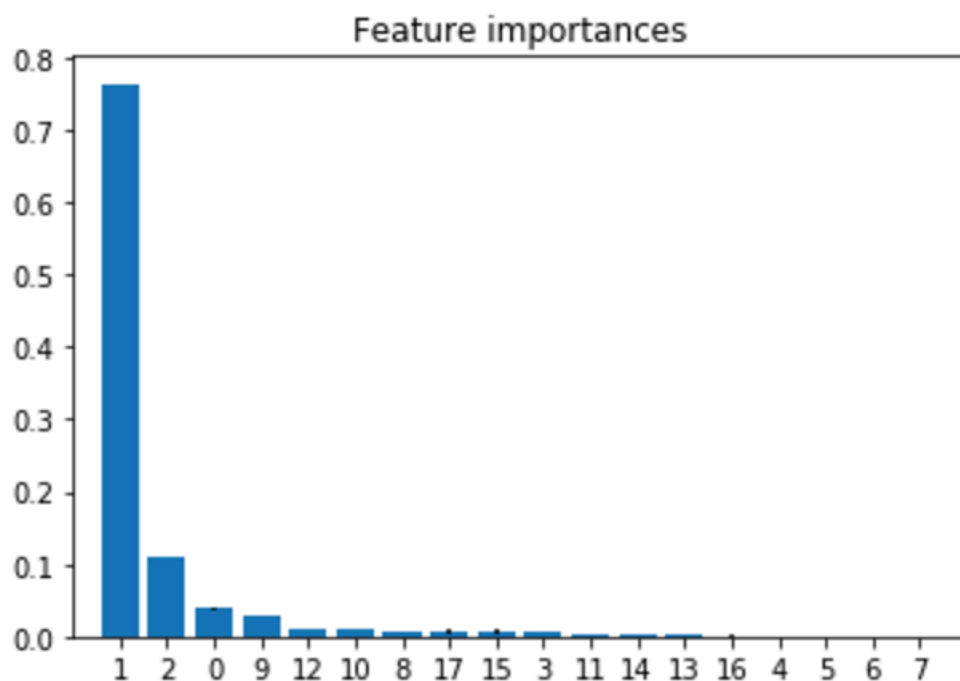


模型名称	Kaggle 得分	本地 RMSPE 得分
DecisionTreeRegressor	0.65122	0.848660136252
RandomForestRegressor	0.70789	0.828673798383

随机森林 Feature importances:

**Feature ranking:**

1. feature 1 DayOfWeek (0.762311)
2. feature 2 Open (0.110023)
3. feature 0 Promo (0.038301)
4. feature 9 SchoolHoliday (0.028667)
5. feature 12 StateHoliday\_0 (0.010569)
6. feature 10 StateHoliday\_a (0.009020)
7. feature 8 StateHoliday\_b (0.007453)
8. feature 17 StateHoliday\_c (0.007069)
9. feature 15 Year (0.006964)
10. feature 3 Month (0.006701)
11. feature 11 Promo2 (0.003873)
12. feature 14 StoreType\_a (0.003814)
13. feature 13 StoreType\_b (0.003580)
14. feature 16 StoreType\_c (0.000959)
15. feature 4 StoreType\_d (0.000492)
16. feature 5 Assortment\_a (0.000161)
17. feature 6 Assortment\_b (0.000022)
18. feature 7 Assortment\_c (0.000021)



第 4 次训练：采用 3- $\sigma$  原则只对 Sales 做处理异常值并做网络搜索优化训练后

结果如下（优化搜索参数同训练 2）：

优化后模型参数是：

```
DecisionTreeRegressor(criterion='mse', max_depth=None, max_features=None,
```

```
    max_leaf_nodes=None, min_impurity_split=1e-07,  
    min_samples_leaf=2000, min_samples_split=100,  
    min_weight_fraction_leaf=0.0, presort=False, random_state=0,  
    splitter='best')
```

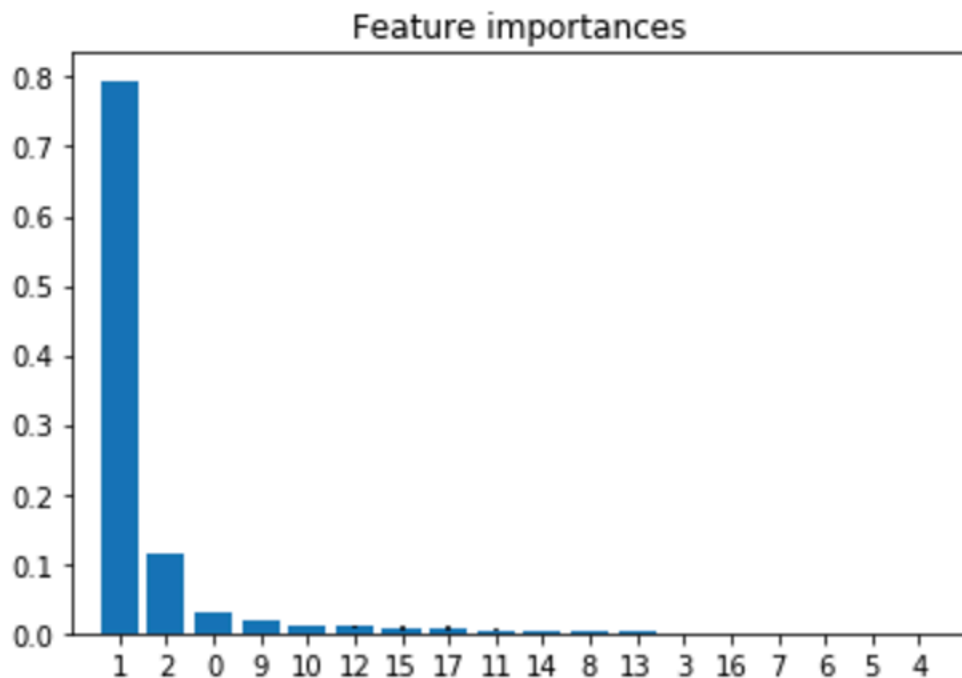
```
RandomForestRegressor(bootstrap=True, criterion='mse', max_depth=None,  
    max_features='auto', max_leaf_nodes=None,  
    min_impurity_split=1e-07, min_samples_leaf=2000,  
    min_samples_split=1000, min_weight_fraction_leaf=0.0,  
    n_estimators=10, n_jobs=-1, oob_score=False, random_state=  
0, verbose=0, warm_start=False)
```

模型名称	Kaggle 得分	本地 RMSPE 得分
DecisionTreeRegressor	0.70747	0.589316895248
RandomForestRegressor	0.69075	0.580364484505

随机森林 Feature importances:

Feature ranking:

1. feature 1 DayOfWeek (0.794296)
2. feature 2 Open (0.114639)
3. feature 0 Promo (0.029915)
4. feature 9 SchoolHoliday (0.017755)
5. feature 10 StateHoliday\_0 (0.009469)
6. feature 12 StateHoliday\_a (0.008975)
7. feature 15 StateHoliday\_b (0.006173)
8. feature 17 StateHoliday\_c (0.005320)
9. feature 11 Year (0.004268)
10. feature 14 Month (0.003351)
11. feature 8 Promo2 (0.003037)
12. feature 13 StoreType\_a (0.002422)
13. feature 3 StoreType\_b (0.000379)
14. feature 16 StoreType\_c (0.000000)
15. feature 7 StoreType\_d (0.000000)
16. feature 6 Assortment\_a (0.000000)
17. feature 5 Assortment\_b (0.000000)
18. feature 4 Assortment\_c (0.000000)



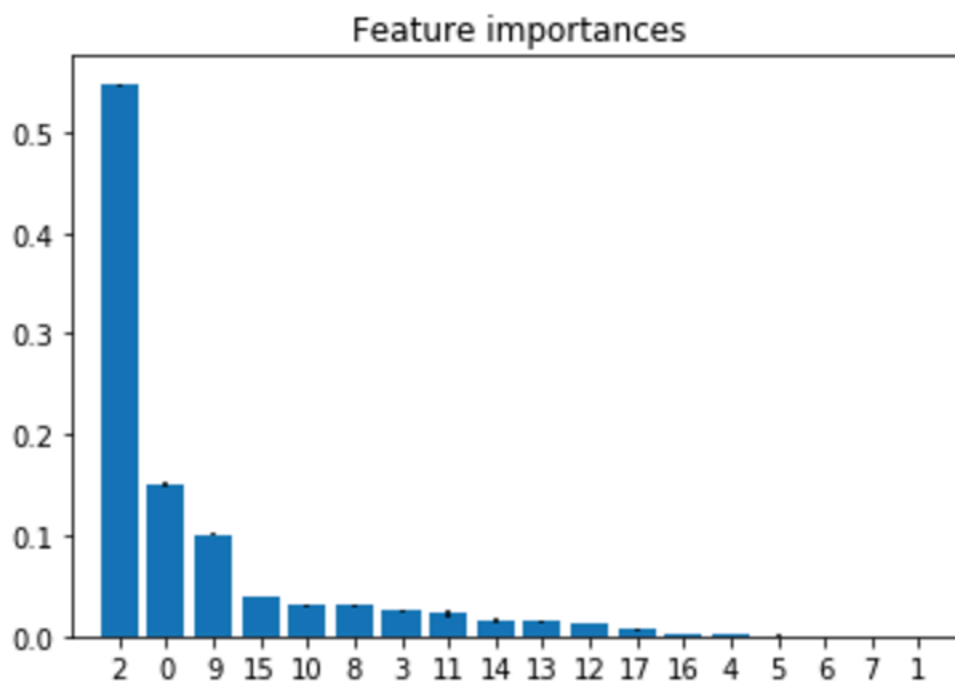
第 5 次训练：根据每个特征对应的 Sales 情况采用 3- $\sigma$  原则处理异常值训练后结果如下。

模型名称	Kaggle 得分	本地 RMSPE 得分
DecisionTreeRegressor	0.57021	0.848660136252
RandomForestRegressor	0.53727	0.828673798383

随机森林 Feature importances:

Feature ranking:

1. feature 2 DayOfWeek (0.546613)
2. feature 0 Open (0.150719)
3. feature 9 Promo (0.100782)
4. feature 15 SchoolHoliday (0.038949)
5. feature 10 StateHoliday\_0 (0.030371)
6. feature 8 StateHoliday\_a (0.030231)
7. feature 3 StateHoliday\_b (0.024896)
8. feature 11 StateHoliday\_c (0.022826)
9. feature 14 Year (0.015867)
10. feature 13 Month (0.014380)
11. feature 12 Promo2 (0.013341)
12. feature 17 StoreType\_a (0.006279)
13. feature 16 StoreType\_b (0.002237)
14. feature 4 StoreType\_c (0.001620)
15. feature 5 StoreType\_d (0.000644)
16. feature 6 Assortment\_a (0.000177)
17. feature 7 Assortment\_b (0.000070)
18. feature 1 Assortment\_c (0.000000)



第 6 次训练：根据每个特征对应的 Sales 情况采用 3- $\sigma$  原则处理异常值后并做

网络搜索优化训练后结果如下（优化搜索参数同训练 2）：

优化后参数：

```
DecisionTreeRegressor(criterion='mse', max_depth=None, max_features=None,
```

```
    max_leaf_nodes=None, min_impurity_split=1e-07,  
    min_samples_leaf=100, min_samples_split=100,  
    min_weight_fraction_leaf=0.0, presort=False, random_state=0,  
    splitter='best')
```

```
RandomForestRegressor(bootstrap=True, criterion='mse', max_depth=None,  
    max_features='auto', max_leaf_nodes=None,  
    min_impurity_split=1e-07, min_samples_leaf=1000,  
    min_samples_split=1000, min_weight_fraction_leaf=0.0,  
    n_estimators=10, n_jobs=-1, oob_score=False, random_state=  
0,
```

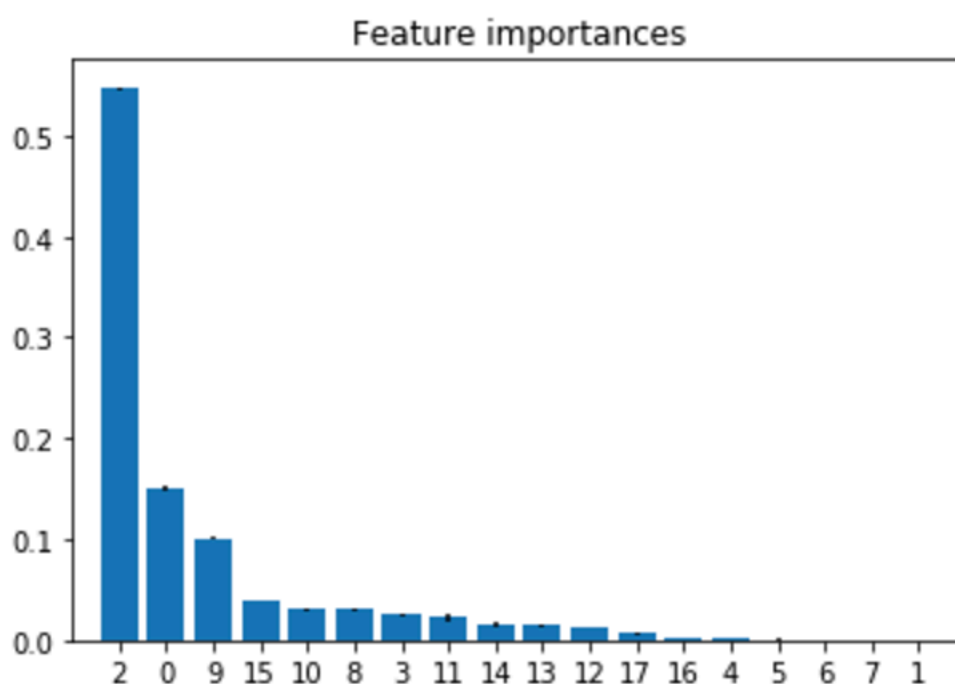
```
    verbose=0, warm_start=False)
```

模型名称	Kaggle 得分	本地 RMSPE 得分
DecisionTreeRegressor	0.52070	0.589316895248
RandomForestRegressor	0.54563	0.580364484505

随机森林 Feature importances:

Feature ranking:

1. feature 2 DayOfWeek (0.546613)
2. feature 0 Open (0.150719)
3. feature 9 Promo (0.100782)
4. feature 15 SchoolHoliday (0.038949)
5. feature 10 StateHoliday\_0 (0.030371)
6. feature 8 StateHoliday\_a (0.030231)
7. feature 3 StateHoliday\_b (0.024896)
8. feature 11 StateHoliday\_c (0.022826)
9. feature 14 Year (0.015867)
10. feature 13 Month (0.014380)
11. feature 12 Promo2 (0.013341)
12. feature 17 StoreType\_a (0.006279)
13. feature 16 StoreType\_b (0.002237)
14. feature 4 StoreType\_c (0.001620)
15. feature 5 StoreType\_d (0.000644)
16. feature 6 Assortment\_a (0.000177)
17. feature 7 Assortment\_b (0.000070)
18. feature 1 Assortment\_c (0.000000)





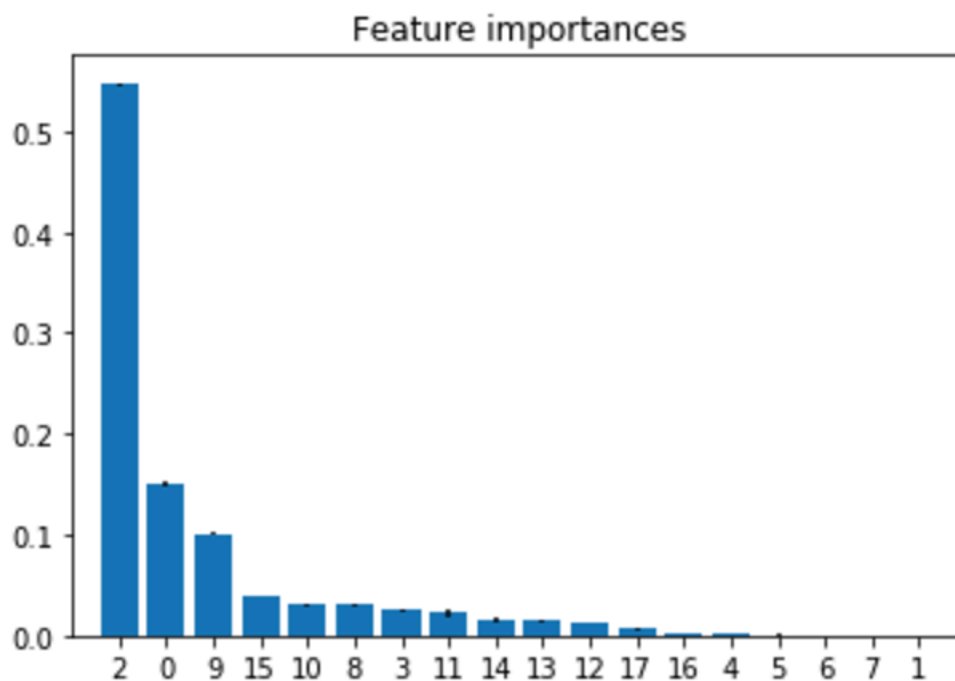
第 7 次训练：对 3-6 次训练可以看出去除异常值对实际预测没有提升。所以本次训练采用针对每个 store 去除异常值进行训练测试。因为训练时间比较长，先使用训练 6 的参数进行测试。

模型名称	Kaggle 得分	本地 RMSPE 得分
DecisionTreeRegressor	0.63615	0.587687144939
RandomForestRegressor	0.63592	0.578719567577

随机森林 Feature importances:

**Feature ranking:**

1. feature 2 DayOfWeek (0.546613)
2. feature 0 Open (0.150719)
3. feature 9 Promo (0.100782)
4. feature 15 SchoolHoliday (0.038949)
5. feature 10 StateHoliday\_0 (0.030371)
6. feature 8 StateHoliday\_a (0.030231)
7. feature 3 StateHoliday\_b (0.024896)
8. feature 11 StateHoliday\_c (0.022826)
9. feature 14 Year (0.015867)
10. feature 13 Month (0.014380)
11. feature 12 Promo2 (0.013341)
12. feature 17 StoreType\_a (0.006279)
13. feature 16 StoreType\_b (0.002237)
14. feature 4 StoreType\_c (0.001620)
15. feature 5 StoreType\_d (0.000644)
16. feature 6 Assortment\_a (0.000177)
17. feature 7 Assortment\_b (0.000070)
18. feature 1 Assortment\_c (0.000000)



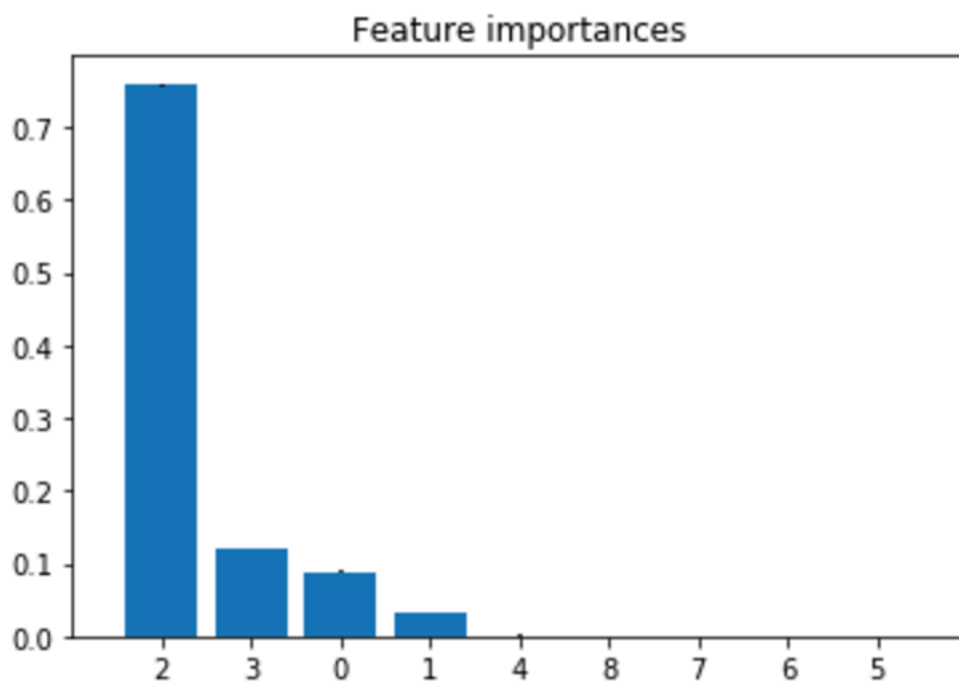
第 8 次训练：考虑是特征使用不对，所以尝试去除 Store 的扩展信息进行训练预测。因为训练时间比较长，先使用训练 6 的参数进行测试。。

模型名称	Kaggle 得分	本地 RMSPE 得分
DecisionTreeRegressor	0.21477	0.192174584985
RandomForestRegressor	0.42731	0.330593810724

随机森林 Feature importances:

**Feature ranking:**

1. feature 2 Store (0.758368)
2. feature 3 DayOfWeek (0.119900)
3. feature 0 Open (0.089095)
4. feature 1 Promo (0.031633)
5. feature 4 SchoolHoliday (0.001003)
6. feature 8 StateHoliday\_0 (0.000000)
7. feature 7 StateHoliday\_a (0.000000)
8. feature 6 StateHoliday\_b (0.000000)
9. feature 5 StateHoliday\_c (0.000000)



## IV. 结果

采用了 RandomForestRegressor、DecisionTreeRegressor。两种模型进行预测。两个模型最终的最好成绩如下：

模型名称	Kaggle 得分
DecisionTreeRegressor	0.21477
RandomForestRegressor	0.42731

从预测结果来看决策树回归效果最好，所以最终选择决策树模型。

尝试去除异常数据进行预测。第一次是只根据 Sales 进行去除。第二次是针对每个特征的值进行去除，效果都有一定的提升，但是总体效果一般。之后尝试去除 Store 的扩展数据，效果有了显著提升，超过基准值达成目标。

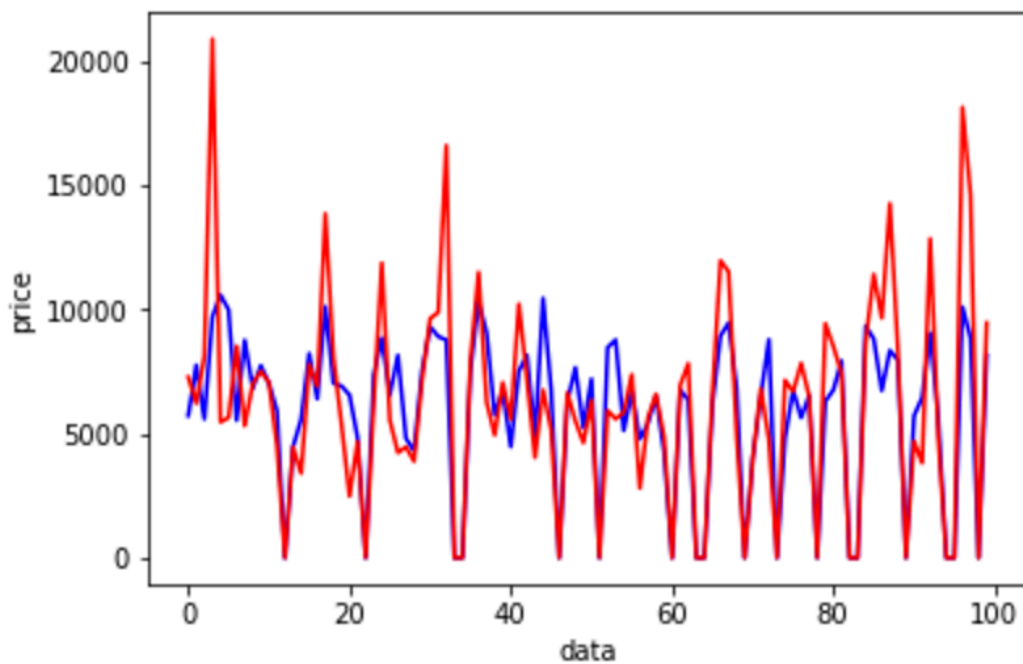
## 合理性分析

从结果来看决策树模型效果为 0.21477，比基准值 0.24813 有一定提升。所以认为结果合理，可以实际应用。

## V. 项目结论

---

### 结果可视化



抽取部分数据查看预测值与实际值相对比较接近。预测效果比较好。不过在异常值时不够准确。

## 对项目的思考

随机森林是一种特殊的决策树回归。

在这个项目中决策树比随机森林效果好。

决策树回归效率比随机森林高。以第 5 次训练来看 `DecisionTreeRegressor` 使用训练数量:805700 用时:1.9 秒。`RandomForestRegressor` 使用训练数量:805700 用时:14.5 秒。`DecisionTreeRegressor` 效率远远高于 `RandomForestRegressor`。

所以项目最终选择选择决策树比较好。

另外通过项目测试，在选择数据应该去除异常值，这样预测相对会更加的准确。

## 需要作出的改进

通过整个项目考虑一些通用方法。在特征不多并且可以方便验证的情况，可以编写程序自动优化选用特征。这样应该可以获取比较好的优化效果与效率。另外可以考虑先 `store` 进行分类，分类后再进行预测，这样可能准确性会更高。另外，针对各种指标结合实际意义进行整理生成新的指标。另外使用网络搜索时可以结合实际情况使用自己的程序。这样会可能优化效果会更好。