小超市供销存管理优化

数据科学导论课程实验报告

王珺[PB14011076] **汪飞[PB14011014]**
cnwj@mail.ustc.edu.cn wf314159@mail.ustc.edu.cn
中国科学技术大学计算机学院

摘要

本文所介绍的实验采用了 SARIMA 与 XGBoost 两种模型对某连锁超市的日常经营数据进行了学习与预测，还使用集成学习的方法将不同参数下的两种模型进行整合，从而加强了预测的准确性。实验的数据集包括该超市内 14 个大类、87 个中类在 243 天的销售数据，采用均方根误差（RMSE）作为算法的性能标准，最终取得了可以接受的结果。

实验背景与问题描述

在大数据行业的推动下，无人超市变的越来越火爆，供销存管理也显得格外重要，精准的预测店铺的商品销量，能让超市及时上货，也可以提升用户的体验，预防因为缺货而导致销量降低的风险。

本次实验为 2017 年 CCF 大数据与计算智能大赛的试题之一[1]，使用了某地市级小连锁超市日常经营的真实数据，利用学习算法在商品促销影响其它商品销量的噪音下，预测出商品大中类的日销量，为超市的供销存管理，提供智能化技术的支撑，为该连锁超市转型无人超市设计合理的算法模型。

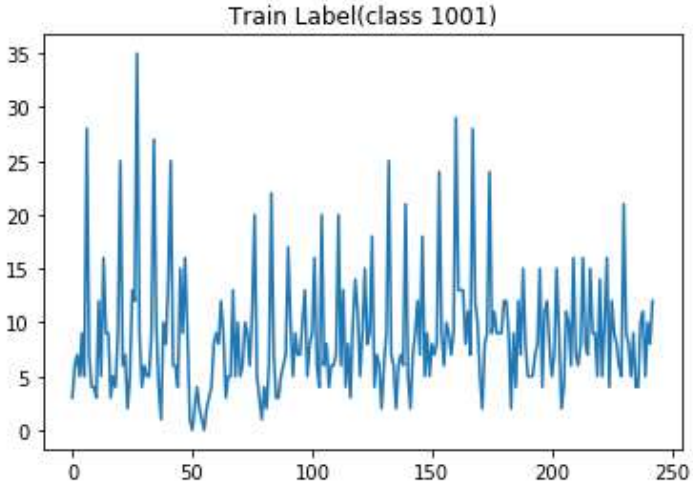
实验的原始数据为 csv 格式的销售记录，共 90843 条，大小为 9.17MB，时间跨度 243 天，每条记录的格式如下图示：

custid	大类编码	大类名称	中类编码	中类名称	小类编码	小类名称	销售日期	销售月份	商品编码	规格型号	商品类型	单位	销售数量	销售金额	商品单价	是否促销	销售周
0	12	蔬果	1201	蔬菜	120109	其它蔬菜	20150101	201501	DW-120106		生鲜	个	8	4	2	否	1
1	20	粮油	2014	酱菜类	201401	榨菜	20150101	201501	DW-201406	60g	一般商品	袋	6	3	0.5	否	1
2	15	日配	1505	冷藏加味	150502	冷藏加味	20150101	201501	DW-150502	150g	一般商品	袋	1	2.4	2.4	否	1
3	15	日配	1503	冷藏料理	150305	冷藏面食	20150101	201501	DW-150305	500g	一般商品	袋	1	6.5	8.3	否	1
4	15	日配	1505	冷藏乳品	150502	冷藏加味	20150101	201501	DW-150502	100g*8	一般商品	袋	1	11.9	11.9	否	1
5	30	洗化	3018	卫生巾	301802	夜用卫生	20150101	201501	DW-301802	10片	一般商品	包	1	8.9	8.9	否	1
6	12	蔬果	1201	蔬菜	120104	花果	20150101	201501	DW-120104	散称	生鲜	千克	0.964	8.07	5.6	否	1
7	20	粮油	2001	袋装速食	200101	牛肉口味	20150101	201501	DW-200101	120g	一般商品	袋	1	2.5	3	否	1
8	13	熟食	1308	现制中式	130803	现制烙类	20150101	201501	DW-130803	个	生鲜	个	2	2	1	否	1
9	22	休闲	2203	膨化点心	220302	袋装薯片	20150101	201501	DW-220302	45g	一般商品	袋	1	4	4	否	1
10	22	休闲	2201	饼干	220111	趣味/休闲	20150101	201501	DW-220111	60g	一般商品	盒	1	6.5	6.7	否	1
11	12	蔬果	1201	蔬菜	120104	花果	20150101	201501	DW-120104	散称	生鲜	千克	0.784	1.55	1.6	否	1
11	12	蔬果	1201	蔬菜	120104	花果	20150101	201501	DW-120104	散称	生鲜	千克	0.401	2.3	9.6	否	1
12	15	日配	1521	蛋类	152101	新鲜蛋品	20150101	201501	DW-152101	散称	一般商品	千克	0.744	5.9	6.78	否	1
13	13	熟食	1301	凉拌熟食	130101	凉拌素食	20150101	201501	DW-130101	散称	联营商品	kg	0.282	6.8	20	否	1
14	20	粮油	2011	液体调料	201111	料酒	20150101	201501	DW-201111	500mL	一般商品	瓶	1	5.5	5.5	否	1
15	10	肉禽	1004	鸡产品	100404	调味鸡肉	20150101	201501	DW-100404	散称	生鲜	Kg	0.64	10.11	19.6	否	1
16	31	家居	3119	卫浴用品	311902	浴球和浴	20150101	201501	DW-311902	202	一般商品	只	1	3	3	否	1
7	34	针织	3412	毯子	341206	双人电热	20150101	201501	DW-341206	150*120	一般商品	条	1	79	90	是	1
17	22	休闲	2201	饼干	220110	简装/压缩	20150101	201501	DW-220110	散称	一般商品	KG	0.198	2.7	19.8	是	1
17	22	休闲	2206	即食熟制	220607	豆干类	20150101	201501	DW-220607	散称	一般商品	千克	0.228	11.4	59	否	1
18	20	粮油	2013	调味酱	201302	辣酱	20150101	201501	DW-201302	280g	一般商品	瓶	1	7.5	7.9	否	1
19	15	日配	1518	常温乳品	151805	利乐枕纯	20150101	201501	DW-151805	240ml	一般商品	袋	16	33.9	2.7	是	1
19	20	粮油	2011	液体调料	201102	生抽酱油	20150101	201501	DW-201102	500ml	一般商品	瓶	1	6.9	7.9	否	1
19	30	洗化	3008	洗护发用	300801	洗发水	20150101	201501	DW-300801	200ml	一般商品	瓶	1	9.5	9.5	否	1
2	12	蔬果	1201	蔬菜	120104	花果	20150101	201501	DW-120104	散称	生鲜	千克	0.708	3.8	2.58	否	1
2	12	蔬果	1201	蔬菜	120104	花果	20150101	201501	DW-120104	散称	生鲜	千克	0.636	1.5	1.8	否	1
20	20	粮油	2007	南北干货	200701	木耳	20150101	201501	DW-200701	散称	一般商品	kg	0.132	8.28	89.8	是	1
2	12	蔬果	1203	水果	120303	梨类	20150101	201501	DW-120303	散称	生鲜	KG	1.922	7.64	3.18	否	1

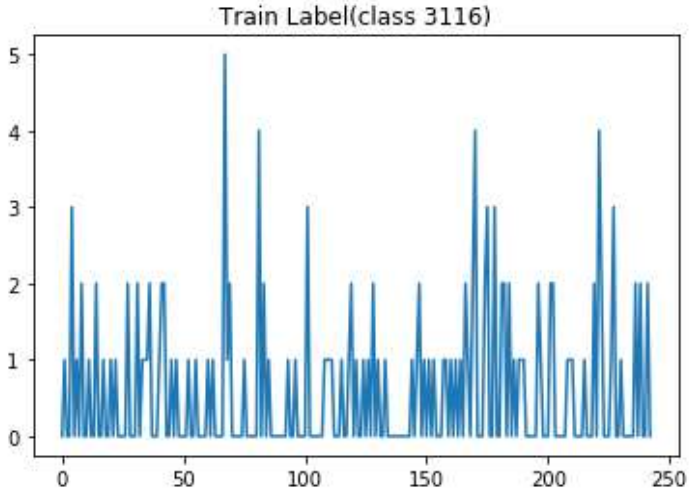
实验的结果为在紧随数据集的 59 天中，对数据集内 14 个大类、87 个中类的销量预测。原始数据中的每条记录（无论其销售数量、销售金额的数值）作为 1 个单位销量。实验结果的格式亦为 csv 文件，形如下图所示：

bianma	riqi	xiaoliang
1001	20150901	9
1001	20150902	16
1001	20150903	9
1001	20150904	8
1001	20150905	8
1001	20150906	9
1001	20150907	9
1001	20150908	10
1001	20150909	17

需要注意的是，对于不同的大类、中类，其日销量有较大的差异，如中类 1001（猪肉）日销量如下图示：



中类 3116（保鲜用品）日销量如下图示：



因此，尽管所有数据均来自于同一超市，如果仅使用单一的预测方法预测所有类别的数据，将难以有好的结果。

小组分工

在本次实验中，王珺主要负责了基于 SARIMA 模型、XGBoost 模型的预测器以及为各个类别自动进行模型选择并利用集成学习结合所有模型做出预测；汪飞主要负责了基于随机森林模型、KNN 模型的预测器并测试了基于 stacking 的集成学习。

鉴于汪飞未选修此课程，他不需要撰写报告。故本报告及报告中所述的所有结果的贡献者均为王珺。

相关工作

基于已有数据对未来销量、价格或市值进行预测是机器学习的热门工作之一，目前这方面已有大量的研究。基于线性回归、神经网络、支持向量回归与决策树等模型进行预测的文献可见[2]-[11]，文献[12]是对利用机器学习方法进行序列预测的一个总结。

在本次实验中用于预测的 XGBoost 模型是一个基于集成学习的模型，其全称是 eXtreme Gradient Boosting，是 Gradient Boosting Machine 的一个实现，开发者为华盛顿大学的博士研究生陈天奇[13]。XGBoost 针对传统的 GBDT 算法做了很多细节改进，对损失函数、正则化、切分点查找算法、稀疏感知算法、并行化算法等方面进行了调整，从而使其能够更好地利用并行计算加快速度，同时也提高了精度。XGBoost 模型的原理可见文献[13]，其使用了一系列的简单模型（如浅层的决策树或线性模型）来逐步逼近复杂的目标函数。

XGBoost 目前已有封装后的 Python 包可用，相关文档见文献[14]。

金融与统计领域也有名为时间序列分析的类似工作。与一般的机器学习模型相比，时间序列分析并不考虑输入的“特征”，即其将要预测的值简化为了一个随时间变化的序列，训练数据为此前一段时间中序列的值，预测的结果是序列在未来一段时间中的值。

时间序列分析的方法大体有简单（加权）平均法、简单（加权）移动平均法、指数平滑法、季节趋势预测法、市场寿命周期预测法等。经常使用的模型包括逐步自回归（StepAR）模型、Winters Method—Additive 模型、ARIMA 模型、Winters Method—Muhiplicative 模型和 GARCH（ARCH）模型等[15]。其中 ARIMA 模型全称为自回归积分滑动平均模型（Autoregressive Integrated Moving Average Model,简记 ARIMA），是由博克思和詹金斯于 1970 年代初提出的一个时间序列预测方法[16]。ARIMA 模型可用于对差分后平稳的时间序列进行建模，其建模公式为

$$\left(1-\sum_{i=1}^{p}\phi_i L^i\right)(1-L)^d X_t = \left(1+\sum_{i=1}^{q}\theta_i L^i\right)\varepsilon_t$$

其中 d 为差分阶数，（p、q）分别是 AR 模型、MA 模型的阶数（后文中记为 ar、ma）。差分阶数 d 需要多次尝试并对差分后的时间序列进行平稳性检测来得到；（ar、ma）可以根据拟合模型的效果来确定，也可以通过自相关系数、偏相关系数或模型的信息量（如 AIC、BIC）来确定。

SARIMA 模型是 ARIMA 模型的拓展，其可以建模具有周期性的时间序列。在 ARIMA 模型

的基础上，SARIMA 模型多了一组参数（sar、sd、sar、s），其中前 3 个参数的作用与 ARIMA 中的参数（ar、d、ma）类似，只是差分的时间间隔变成了 s。

在 python 中，statsmodels 包实现了 SARIMA 模型，相关文档见文献[17]。

实验方法

本次实验分为预处理和学习预测两个阶段，均使用 python 程序实现。

在预处理阶段，需要对销售记录进行统计以得到各中类、大类每天的销量（标签）以及其他有价值的特征。考虑到对于测试集除了日期相关的信息外几乎没有任何其他可用信息，故预处理过程中提取的特征以日期相关的特征为主，预处理后数据的格式如下：

中类特征																
唯一性数据			时间类				促销类		历史信息				全局类			标签
中类编号	总日期	月份	日期	星期	节假日前	节假日	是否有促销活动	大类中其他促销的中类数量	前一天销量	两天前销量	三天前销量	前一周均销量	当天总销量	当天总营业额		当天销量
大类特征																
唯一性数据			时间类				促销类		历史信息				全局类			标签
大类编号	总日期	月份	日期	星期	节假日前	节假日	是否有促销活动		前一天销量	两天前销量	三天前销量	前一周均销量	当天总销量	当天总营业额		当天销量

其中时间类、促销类、历史信息、全局类中的条目可以视为特征，标签为当天的销量。

此外，由于数据集中有几天的销售记录缺失，在预处理的过程中还要填充缺失数据。填充的方式为先设置这几天的销量为 0 并训练一个低阶的 SARIMA 模型，然后利用上述模型预测这几天的销量作为这几天的真实销量。

预处理后的数据集如下图示：

1001	1	1	1	4	0	1	0	0	0	0	0	0	313	2765.8	3
1001	2	1	2	5	0	1	0	0	3	0	0	0	338	3253.5	6
1001	3	1	3	6	0	1	0	0	6	3	0	1	283	2400	7
1001	4	1	4	0	0	1	0	0	7	6	3	2	343	2521.6	5
1001	5	1	5	1	0	0	0	0	5	7	6	3	381	3026.9	9
1001	6	1	6	2	0	0	0	0	9	5	7	4	408	3971.4	5
1001	7	1	7	3	0	0	0	0	5	9	5	5	506	5036.5	28
1001	8	1	8	4	0	0	0	0	28	5	9	9	348	3020.9	7
1001	9	1	9	5	1	0	0	0	7	28	5	9	351	2778.3	4
1001	10	1	10	6	0	1	0	0	4	7	28	9	372	3400.4	4
1001	11	1	11	0	0	1	0	0	4	4	7	8	347	3027.7	3
1001	12	1	12	1	0	0	0	0	3	4	4	8	346	3602.9	12
1001	13	1	13	2	0	0	0	0	12	3	4	9	372	2734.6	5
1001	14	1	14	3	0	0	0	0	5	12	3	9	433	4843.7	16
1001	15	1	15	4	0	0	0	0	16	5	12	7	365	2747.53	9
1001	16	1	16	5	1	0	0	0	9	16	5	7	399	3266.1	9
1001	17	1	17	6	0	1	0	0	9	9	16	8	347	2963.5	3
1001	18	1	18	0	0	1	0	0	3	9	9	8	411	3583.8	5
1001	19	1	19	1	0	0	0	0	5	3	9	8	523	5109.34	4
1001	20	1	20	2	0	0	0	0	4	5	3	7	411	3398.1	9
1001	21	1	21	3	0	0	0	0	9	4	5	7	591	5462.7	25
1001	22	1	22	4	0	0	0	0	25	9	4	9	406	3187.54	6
1001	23	1	23	5	1	0	0	0	6	25	9	8	380	4228.8	7
1001	24	1	24	6	0	1	0	0	7	6	25	8	330	3586.2	2
1001	25	1	25	0	0	1	0	0	2	7	6	8	379	3600	5
1001	26	1	26	1	0	0	0	0	5	2	7	8	400	3768.6	13
1001	27	1	27	2	0	0	0	0	13	5	2	9	401	3860.63	12
1001	28	1	28	3	0	0	0	0	12	13	5	10	685	6867	35
1001	29	1	29	4	0	0	0	0	35	12	13	11	359	2962.93	9
1001	30	1	30	5	1	0	0	0	9	35	12	11	305	2781	4

在学习预测阶段，实验中分别采用了 SARIMA 预测器和 XGBoost 预测器对**每个类别的数据分别进行了学习与预测**，然后使用集成学习的方法将不同参数下的上述预测器进行集成，从而提高预测的性能。

SARIMA 预测器可以调用 statsmodels.tsa.statespace.sarimax 包实现，输入的训练数据为数据集上连续若干天的销量（标签），输出预测数据为紧随输入之后若干天的销量。SARIMA 预测器首先利用 log 函数对输入数据进行平滑处理以消除趋势，然后拟合 SARIMA 模型。SARIMA 模型的确定需要参数(ar,d,ma)和(sar,sd,sma,s)，易于发现对于所有的类别有(sar,sd,sma,s)=(1,1,0,7)时模型拟合的效果最好，d 的最优取值也总为 1，但是对于参数 ar,ma 不同类别的最优选择不尽相同。因此，SARIMA 预测器首先将训练数据划分为训练集、验证集两部分并在训练集上分别训练(ar,ma)为(1, 1), (0, 1), (1, 2), (2, 0), (2, 1), (2, 2)的模型以在验证集上测试准确度，然后选择准确度最高的模型在全部训练数据上进行训练并预测未来销量进行输出。

在实验中可以发现，当训练数据较少时，如仅使用测试准确度作为标准选择模型参数，可能会产生严重的过拟合。这种情况在训练数据增加时可以得到显著缓解，实验最终采用的集成学习机制也减轻了过拟合的误差。如欲仅采用单个 SARIMA 模型在小数据集上进行预测，则在参数选择时可以综合考虑测试准确度与 AIC，但是需要注意的是仅考虑 AIC 的效果比仅考虑测试准确度更差。

XGBoost 预测器可以调用 xgboost 包实现，其输入的训练数据包括矩阵格式的训练样本特征，向量形式的训练样本标签（销量）和矩阵形式的测试样本特征，输出为测试样本的标签。由于 XGBoost 预测器最初仅作为 SARIMA 预测器效果不佳时的替补，其在十余个 SARIMA 预测误差较大的中类上进行了特征选择与参数调优，最终确定的输入特征为：

中类特征					
日期	星期	节假日前	节假日	是否有促销活动	大类中其他促销的中类数量
大类特征					
日期	星期	节假日前	节假日	是否有促销活动	

模型参数为（除指定参数外其他参数均采用默认参数）：

{"objective":"reg:linear", "max_depth":1, "gamma":2}

在集成学习的过程中，首先从 243 天的数据集中确定以下 3 个学习问题

序号	训练集	测试集
1	前 150 天数据	第 151 至 178 天数据
2	前 180 天数据	第 181 至 208 天数据
3	前 210 天数据	第 211 至 238 天数据

然后对每个学习问题分别训练全部的预测器并记录下在每个问题上表现最好的预测器。最后使用全部数据集训练记录下来的 3 个预测器并将其输出的平均值取整后作为预测值输出。需要注意的是，虽然本次实验中只采用了 SARIMA 预测器与 XGBoost 预测器两个预测器，但是在 SARIMA 预测器中包含了参数选择的过程（选择合适的 ar、ma 参数）且对于不同的学习问题 SARIMA 预测器选择的参数未必相同。因此，集成学习中预测器间的差异性是可以保证的。

实验结果

本次实验中全部的代码、数据和提交的结果（代码的作者为王珺、汪飞两人，其中汪飞的贡献在本文中未予体现）已经在 GitHub 上开源，网址为：

https://github.com/IngramWang/BDCI2017

在 2017 年 CCF 大数据与计算智能大赛中，本小组的成绩为

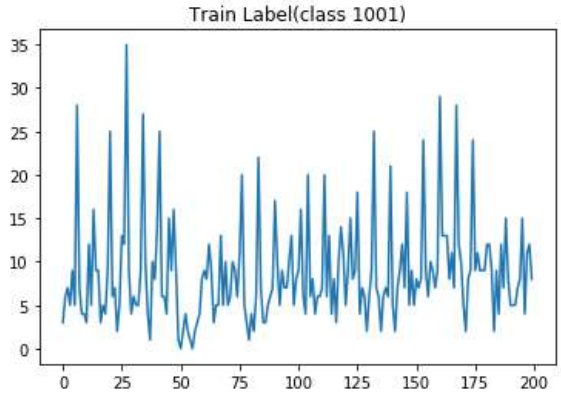
队伍编号：42990 队伍名称：喃喃		
榜单	得分*	名次
初赛榜 A 榜	0.17856	18
初赛榜 B 榜	0.19340	14
复赛榜 A 榜	0.19133	47
复赛榜 B 榜	0.11275	57

* 得分的计算公式为$\frac{1}{\mathrm{RMSE}+1}$，其中RMSE为均方根误差，$\mathrm{RMSE}=\sqrt{\frac{\sum_{i=1}^{n}(x_{obs,i}-x_{pred,i})^2}{\mathrm{n}}}$.

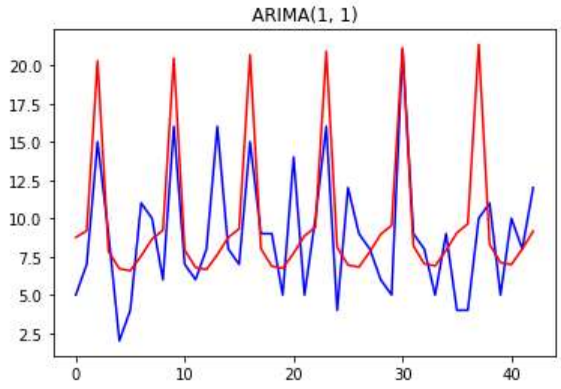
鉴于比赛的测试集并未公布，下面在比赛的训练数据上测试 SARIMA 预测器与 XGBoost 预测器的性能（由于集成学习结果的时间范围已在代码中写定，这里难以进行展示）。限于篇幅长度，这里只讨论中类 1001（猪肉）、中类 1201（蔬菜）、中类 1504（冷藏奶油芝士）和大类 20（粮油）几个类别，它们分别代表了销量一般的中类、销量较多的中类、销量较少的中类及大类的情况。

在以下的测试中，训练数据为前 200 天的结果，测试数据为后 43 的结果。由于 SARIMA 模型要求训练数据连续，这里无法采用交叉验证。

对于中类 1001 有训练标签如下：

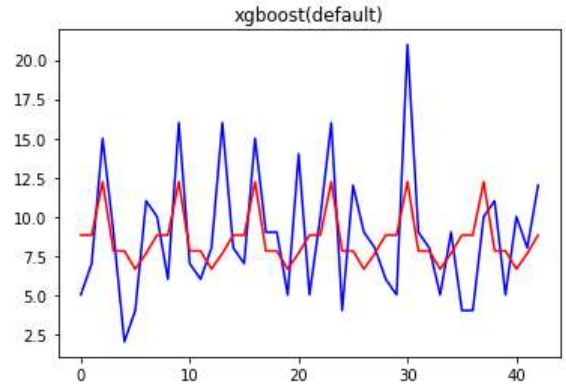


SARIMA 分类器的预测结果为（红色为预测值，蓝色为实际值，下同）：



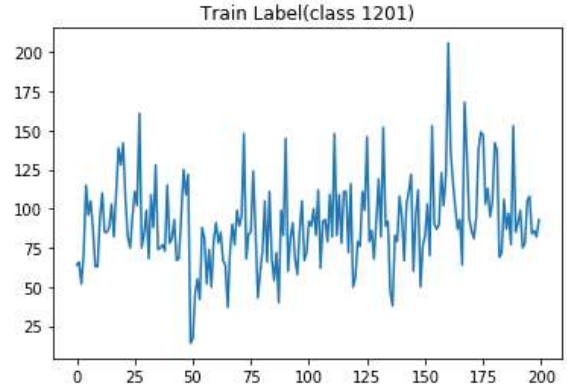
RMSE = 3.7914535309520607

XGBoost 分类器的预测结果为：

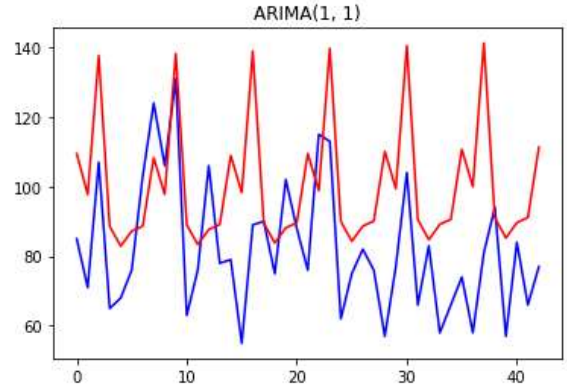


RMSE = 3.4090451851947168

对于中类 1201 有训练标签如下：

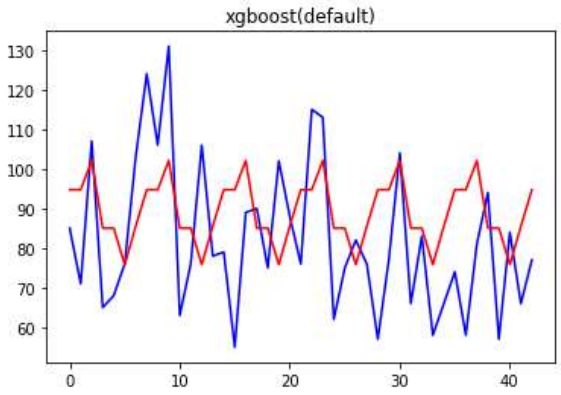


SARIMA 分类器的预测结果为：



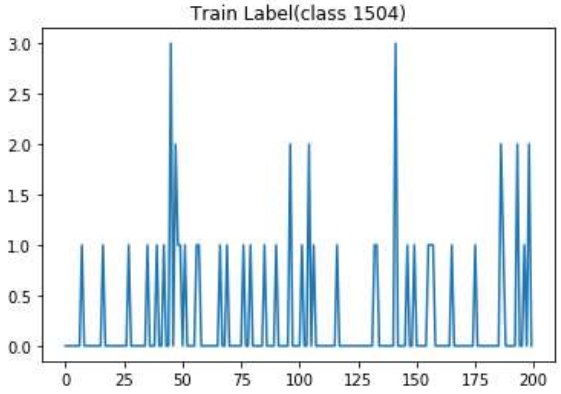
RMSE = 26.359961044416508

XGBoost 分类器的预测结果为：

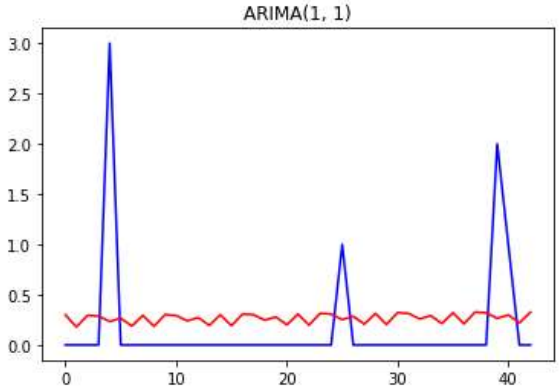


RMSE = 19.376705844334452

对于中类 1504 有训练标签如下：

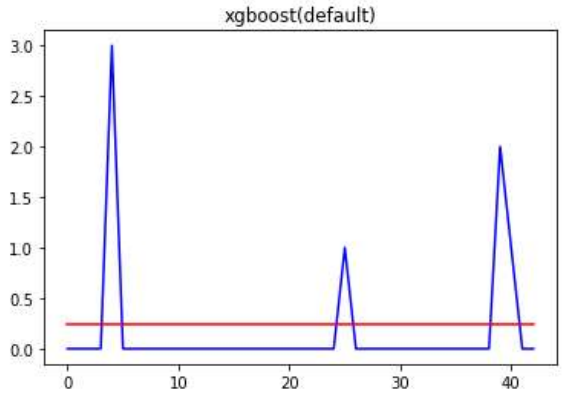


SARIMA 分类器的预测结果为：



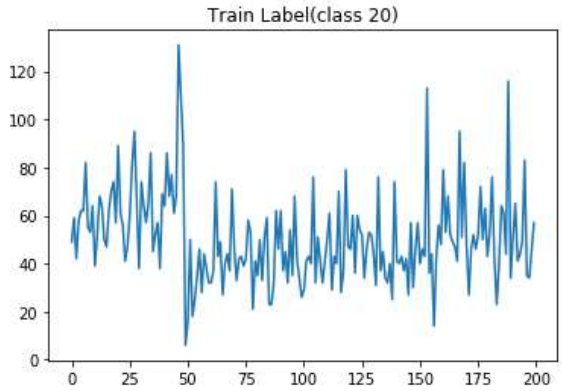
RMSE = 0.5820394564588943

XGBoost 分类器的预测结果为：

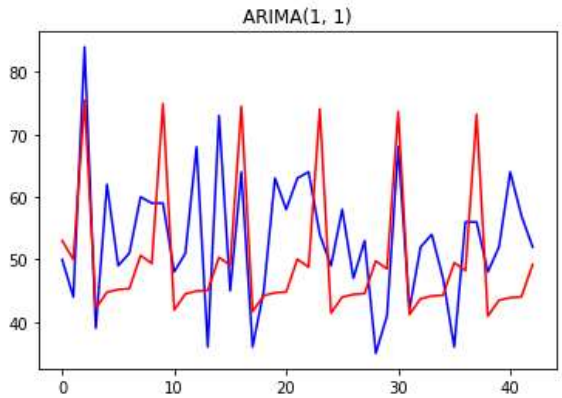


RMSE = 0.573526983572075

对于大类 20 有训练标签如下：

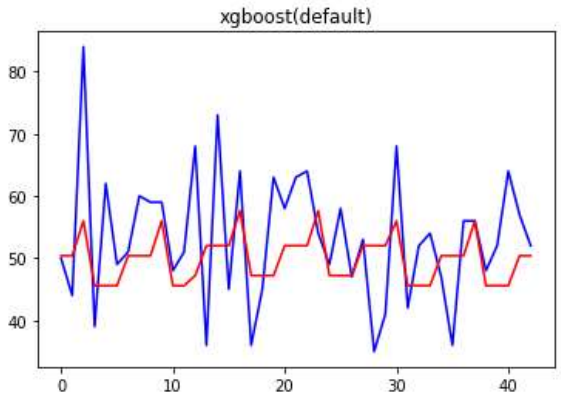


SARIMA 分类器的预测结果为：



RMSE = 11.453227305690119

XGBoost 分类器的预测结果为：



RMSE = 10.516797816856501

分析与总结

从实验结果中可以看出，由于销售数据具有较大的随机性，对于销量较大的类别（如中类1201），两种预测器均很难做出准确的预测。对于销量一般的类别（如中类 1001、大类 20），两种预测器的输出基本上都是一个以 7 为周期的序列，其准确度是可以接受的。对于销量较少的类别（如中类 1504），尽管单个预测器的结果貌似不尽人意，但是在使用集成学习的方法平均并取整后效果将大大改善。

从上述图表中也可以看出，对于相同类别的预测，SARIMA 预测器的输出往往变化范围更大而 XGBoost 预测器的输出往往波动较小。这是 XGBoost 预测器在调参的过程中仅使用了 SARIMA 预测器预测效果较差或无法收敛的类别（一般销量较少或销量变化随机性较大）的结果。从参数中也可以看出，"max_depth":1 决定了 XGBoost 的基学习器仅仅是最简单的决策树桩，这在最大限度地避免了过拟合的同时也限制了结果变化的幅度。

对于这次实验，进一步提高预测器性能的关键是要设法从数据中抽取出更有意义的特征。事实上，在实验的过程中小组成员也尝试了包括 LSTM 在内的深度学习模型。这些模型有更为复杂的结构，具备更强的表达能力，却无法得到更精确的结果。究其原因，数据量太少是显而易见的：每个中类的 243 条数据与神经网络动辄成千上万的数据相比可谓只是零头；数据的维数太小也是不可忽略的：过少的数据维数使得模型难以区分各个样本，这使得输出基本上都是一个以 7 为周期的序列。然而，考虑到我们对于超市在待预测时期内的活动几乎一无所知，想要抽取出在训练集中有代表性、在测试集中易于得到的特征是十分艰难的。

至于比赛，在初赛与复赛中，本小组的名次变化较大。这一方面是因为复赛过程中小组成员事务繁多难以投入足够的时间；另一方面也是因为初赛与复赛的数据集有较大的不同：不少在初赛中需要预测的类别在复赛中不再要求预测了。考虑到这些类别往往是销量较少的类别，删去这些类别将使得 XGBoost 预测器的效果大打折扣。

参考文献

[1] 小超市供销存管理优化 http://www.datafountain.cn/#/competitions/274/intro

[2] 嵌入遗传算法的神经网络技术在煤炭销量预测研究中的应用 林红 东北大学硕士生论文

[3] 神经网络在卷烟销量预测及营销策划中的应用 关雷 内蒙古大学硕士生论文

[4] 基于 SVR 与半监督学习的时间序列预测 周若愚 西安电子科技大学硕士生论文
[5] 金融时间序列预测的 SVR 建模及参数优化研究 焦帅 东华理工大学硕士生论文
[6] 基于 ARMA 和 BP_AdaBoost 的组合销售预测模型研究 闫博，周在金等 计算机与现代化，2015 年 02 期
[7] 基于决策树分类算法的高职学生就业分析与预测 孙晓璇 云南大学硕士生论文
[8] 基于决策树 CART 算法的镍金属价格预测研究 孙建召 世界有色金属，2016 年 18 期
[9] G. Peter Zhang. Time series forecasting using a hybrid ARIMA and neural network model. Neurocomputing, Volume 50, January 2003
[10] E. Michael Azoff. Neural Network Time Series Forecasting of Financial Markets. John Wiley & Sons, Inc. New York, NY, USA ©1994
[11] L.J. Cao, F.E.H. Tay. Support vector machine with adaptive parameters in financial time series forecasting. IEEE Transactions on Neural Networks (Volume: 14, Issue: 6, Nov. 2003)
[12] Nesreen K. Ahmed, Amir F. Atiya, Neamat El Gayar, Hisham El-Shishiny. An Empirical Comparison of Machine Learning Models for Time Series Forecasting. Econometric Reviews Volume 29, 2010
[13] Tianqi Chen, Carlos Guestrin. XGBoost: A Scalable Tree Boosting System. KDD 2016
[14] XGBoost Documents. https://xgboost.readthedocs.io/
[15] 时间序列预测法. http://wiki.mbalib.com/wiki/%E6%97%B6%E9%97%B4%E5%BA%8F%E5%88%97%E9%A2%84%E6%B5%8B%E6%B3%95
[16] G. E. P. Box and G. M. Jenkins. Time Series Analysis: Forecasting and Control. San Francisco：Holden Day, 1976
[17]statsmodels.tsa.statespace.sarimax.SARIMAX. http://www.statsmodels.org/dev/generated/statsmodels.tsa.statespace.sarimax.SARIMAX.html