Stacking 集成学习方法在销售预测中的应用

王 辉 李昌刚
(浙江万里学院信息与智能学院 浙江 宁波 315000)

摘 要 为了提高单一预测模型在销售预测中的性能,提出一种在多机器学习模型融合下基于 Stacking 集成策略的销售预测方法。将数据划分为四个同分布的数据集;基于各数据集训练多个基学习器;以 XGBoost 算法为元学习器构建两层 Stacking 集成学习方法;使用德国 Roseman 超市在 Kaggle 平台上的销售数据对算法进行验证。实验结果表明:在 Stacking 模型中,元学习器利用各基学习器的算法优势提升了模型的预测性能,相比单个模型在测试集上的均方根百分误差,Stacking 模型最高减少了 23.5%,最低减少了 1.8%。

关键词 机器学习 销售预测 Stacking 集成学习 XGBoost

中图分类号 TP3 **文献标志码** A DOI: 10.3969/j.issn.1000-386x.2020.08.016

APPLICATION OF STACKING INTEGRATED LEARNING METHOD IN SALES FORECASTING

Wang Hui Li Changgang
(*School of Information and Intelligence, Zhejiang Wanli University, Ningbo 315000, Zhejiang, China*)

Abstract In order to improve the performance of single prediction model in sales forecasting, we propose a sales forecasting method based on Stacking integration strategy under the fusion of multi-machine learning model. The data was divided into four equally data sets with the same distribution; multiple base learners were trained based on each data set; the two-layer Stacking integrated learning method was constructed with XGBoost algorithm as the meta-learner; we verified the algorithm using the sales data from the German Roseman supermarket on the Kaggle platform. The experimental results show that in the Stacking model, the meta-learner improves the prediction performance of the model by taking advantage of the algorithm advantages of each base learner. The Stacking model reduces the error by 23.5% at the highest and 1.8% at the lowest, compared with the root mean square error of a single model on the test set.

Keywords Machine learning Sales forecasting Stacking integrated learning XGBoost

0 引 言

大数据与机器学习的结合为当今社会带来的巨大的变革,从每天 2 500 万人次足不出户地在饿了么平台上找到自己喜欢的餐厅和食物,到亚马逊在客户服务中运用大数据精准预测出客户的需求来建立高效的物流运转体系,都显现着变革带来的数字化趋势的威力。机器学习技术基于强大的计算平台给各行业,特别是零售行业带来了成本的降低和效益的增长。大数据、云计算、物联网,必然会使传统零售向人工智能互联网时代的新零售过渡,零售业在实体经济中的权重地位也必然会获得大幅增强。

销售预测在零售行业中有着举足轻重的地位,准确的销售预测结果不仅能够让管理者合理安排订货时间和库存,减少安全库存成本和缺货损失,还能够支持高层管理者在指定战略发展目标的可靠性。由于数据存储技术的发展,企业存储了大量的数据来支持企业的运营决策,在一些数据量大,数据结构多样的销售预测场景下,传统的统计学方法[1] 可能无法取得精准的预测结果,因此国内外学者对机器学习技术在销售预测中的应用进行了大量的研究。

收稿日期:2019-06-16。宁波市科技特派员团队项目(2018C80002-10)。王辉,硕士生,主研领域:机器学习,智能信息处理。李昌刚,副教授。

Grasman 等[2] 使用 Bass 模型和销售的数据，为未来销售进行了点估计，并假设噪音的大小与年销售额成正比，利用时间依赖的 Ornstein-Uhlenbeck 过程的形式，给出了误差的置信区间。姜晓红等[3] 以某电商平台数据为例，运用时间序列法 ARIMA 模型预测各种商品在未来一周全国和区域性需求量，并与简单移动平均法预测结果做对比，发现 ARIMA 模型有更高的精准度。Loureiro 等[4] 通过深度学习方法获得的销售预测与决策树、随机森林、支持向量回归、人工神经网络和线性回归获得的销售预测进行比较，发现采用深度学习的模型在预测时尚零售市场的销售方面具有良好的性能。Duan 等[5] 将支持向量机、BP 神经网络与 K-最近邻算法在预测手机销售前景的准确性上进行了比较，发现在小样本的前提下，支持向量机能更好地预测出各类手机的销量。

基于机器学习的销售预测的另一种方式是使用自然语言处理(NLP)工具，使计算机能够识别潜在消费者的语音和电子邮件中的关键词，以预测这些消费者将购买的可能性。Fan 等[6] 为了产品销售预测，开发了一种结合 Bass/Norton 模型和情感分析同时使用历史销售数据和在线评论数据的新方法。利用 Naive Bayes 算法的情感分析方法，从每个在线评论的内容中提取情感指标，并将其整合到 Bass/Norton 模型的仿制系数中，以提高预测精度，并收集了真实的汽车行业数据和相关的在线评论对模型进行验证。

由于机器学习算法在应用的过程中通常需要调整算法中的超参数，因此有些学者对算法进行了优化。黄鸿云等[7] 基于改进的多维灰色模型(Gm(1,N))和神经网络(ANN)来预测销量，其中多维灰色模型对销售数据建模，神经网络对误差进行校正。利用阿里天猫销售数据来评估混合模型的表现，实验结果表明，该模型的预测结果优于其他几种销售预测模型。张文雅等[8] 通过网格搜索优化了支持向量机的超参数，并用汽车销售数据来对优化后的算法进行了验证，发现优化后的算法拥有更好的预测性能。王锦等[9] 利用遗传算法能够全局寻优的特点，将 BP 神经网络各隐层的权值和阈值进行了遗传优化，结果表明，模型的稳定性和收敛速度得到了显著的提高。罗嗣卿等[10] 通过 DBSCAN 算法解决了 K-means 算法对噪声数据敏感的问题，并结合 ARIMA 模型以蓝莓干销售数据验证了改进后的算法的精确性。

虽然通过优化算法的超参数或者结合几种算法能在销售预测问题上取得比未改进的单一模型更高的精准度，但是当销售数据的属性以及异常值很多时，单一模型在样本外的数据中往往不能带来更好的效果。因此，有学者采取组合的方式来将模型结合起来或者通过集成策略来达到更好的预测性能，例如 Timmermann[11] 发现预测的线性组合可能会改善其每个贡献者。常晓花[12] 通过使用 boosting 集成策略下的随机森林算法对医疗器械进行了销售预测，发现采取 boosting 集成策略的随机森林模型比未采取集成策略的预测模型减少了 12% 的误差。不过组合模型的缺陷是各单一模型使用的训练数据集仍然是相同的，采用数值上的线性组合并不能真正带来模型在泛化能力上的提高，各模型不能在算法层面上优势互补，而且组合预测的理论支撑不够，不能让使用者信服，而单一的随机森林算法或者 xgboost[13] 算法属于同质集成，在样本外的泛化能力仍然需要改进。

为解决单模型或者同质集成模型在大样本销售预测中泛化能力不强的特点，考虑使用 Stacking 集成策略[14]，将同质集成算法方法 XGBoost、Randomforest 与其他算法结合，构建两层 Stacking 集成学习模型进行销售预测，并使用德国 Rooseman 超市的销售数据对算法进行了验证。结果表明，Stacing 集成策略能结合不同机器学习算法的优势提升模型的预测性能，相比单个模型，Stacking 模型有着更高的精度和泛化能力。

1 算法理论

1.1 Stacking 集成学习方法

在 Stacking 集成学习方法中，整个历史数据集划分为若干个子数据集，子数据集划分为训练集和验证集，由基学习器拟合训练集中的数据来产生底层模型，并用模型在验证集上产生的预测值作为第二层的输入。这样，高层的学习器能够进一步对模型进行泛化增强，这是 Stacking 方法总能够在测试集上取得很好的预测性能的原因。区别于随机森林中的 Bagging 策略，Stacking 模型利用的是算法层面上的优势结合，因此 Stacking 集成策略可以看作是异质集成。这意味了底层学习器需要保持差异性，否则 Stacking 集成策略依然是变相的 Bagging 策略。基本的两层 Stacking 算法流程如算法 1 所示。

算法 1 Stacking 算法

输入：

训练集：$S_1=\{(x_1,y_1),(x_2,y_2),\cdots,(x_m,y_m)\}$

保留集：$S_{\text{hold-out}}=\{(x_1,y_1),(x_2,y_2),\cdots,(x_l,y_l)\}$

测试集: $S_2=\{(x_1,y_1),(x_2,y_2),\cdots,(x_m,y_m)\}$

基学习器: $\zeta_1,\zeta_2,\cdots,\zeta_l$

元学习器: ζ_{meta}

Step 1　for $i=1,2,\cdots,l$

do $\zeta_i \cdot fit(x_i,y_i)$　　//生成 l 个基学习器

end

Step 2　for $i=1,2,\cdots,l$

do $P_i = \xi_i\, predict\left(\int_{\text{hold-out}}(x_i,y_i)\right)$

　　//生成 l 个保留集上的预测向量

do $H_i=\xi_i\, predict\ S_2$　　//生成 l 个测试集上的预测向量

end

Step 3　$P=\bigcup_{i=1}^{i=L}(p_i,y_i)$　　//生成用于元学习器的训练集

$H=\bigcup_{i=1}^{i=L}H_i$　　//生成用于元学习器的测试集

$\xi_{\text{meta}}\, fit\ P$

Step 4　$\xi_{\text{meta}}\, predict$

输出: 生成测试集上的预测值

由于 Stacking 模型使用基学习器在第一层产生的预测值作为第二层的输入，这意味着基学习器与元学习器所使用的训练数据必须不同，否则数据会被过度学习，导致模型的过拟合。因此在分割原始数据集时，要保证每份数据都有训练集和保留集，用基学习器在保留集上的预测数据作为第二层的训练数据; 在分割时间序列数据时，要保持数据的同分布，不能简单地以时间线来分割数据集。这些工作会让模型拥有更好的性能和泛化能力。

1.2　两层 Stacking 集成模型

集成学习(Ensemble learning) 指的是基于多个算法，通过不同的方式来组成新的学习方法，对于单个预测模型来说，集成学习模型能够获得更加优越的预测性能，模型准确率呈现边际效用递减的趋势。在 Stacking 集成学习模型中，基学习器的差异性越高，模型性能越好，因此在建模之前，不仅要分析各基学习器的预测效果，也要分析基学习器之间的差异。

本文选择随机森林模型、线性回归模型、KNN 模型以及 XGBoost 模型作为第一层的基学习器。其中，随机森林模型和 XGBoost 模型基于决策树模型分别使用 Bagging 和 Boosting 的集成学习方式，在实践应用中取得了较好的效果。KNN 模型因为其理论成熟、训练方式高效等特点也有着广泛的应用，而加入线性回归模型是为了保持各算法之间的差异性，使得模型能够获得更好的预测性能。由于第二层的元学习器既要能够纠正各算法的偏差，也要能够保持较高的泛化能力来防止过拟合，因此选择 XGBoost 算法来作为元学习器。Stacking 模型的总体架构如图 1 所示。

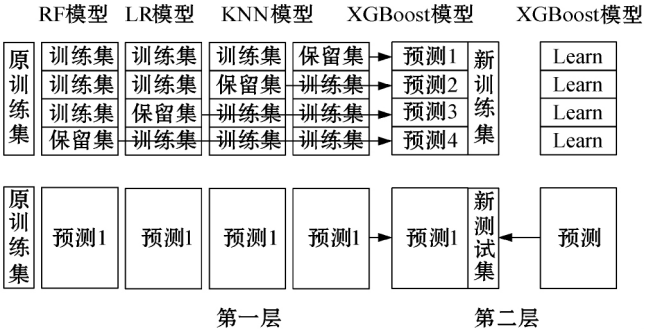


图 1　Stacking 模型框架

对于集成策略来说，集成模型需要从不同的数据空间和数据结构角度来观察预测数据，再依据观察结果以及模型自身的特点来从本质上改善模型的预测性能。因此，需要考察各基学习器在保留集上的预测误差的相关性。本文采用 Pearson 相关系数对各个模型的误差差异度进行计算，以衡量不同基学习器的关联程度，其计算公式为:

$$r_{xy}=\frac{\sum_{i=1}^{m}(x_i-\bar{x})(y-\bar{y})}{\sqrt{\sum_{i=1}^{m}(x_i-\bar{x})^2}\sqrt{\sum_{i=1}^{m}(y_i-\bar{y})^2}} \tag{1}$$

式中: $\bar{x}$ 和 $\bar{y}$ 为各向量中元素的平均值。

为了不让 Stacking 模型过拟合，必须为每个子学习器安排不同的训练集和测试集，这是因为元学习器的训练集是基学习器的输出，如果直接用基学习器的训练集结果来拟合元学习器，会导致学习器对数据的重复学习。因此，本文按照日期将数据分为四块，再将每块中的数据分成四份，从四份中随机抽取一份，按块顺序结合并形成新的四块数据，以保证数据的相同分布; 在新的数据块中将数据分为三份训练和一份验证集，元模型用验证集的来产生第二层的训练集。这样不仅保证了数据不会被重复学习，也保证了各基学习器所使用的数据是同分布的。

另外，在构建模型的输入数据时，需要考察各特征之间的相关性和特征的重要程度。本文先对数据进行了预处理，再通过计算各特征之间的 Pearson 相关系数来挑选出相对独立的特征。由于 Random forest 和 XGBoost 算法可以计算各树的增益情况来评估各特征的评分，所以在完成第一次的模型训练后，查看模型输出的特征评分，从而删除冗余特征，使得模型能够取得更好的性能。

2　实证分析

实验使用了德国 Rossman 日用品超市在 Kaggle 平台上所提供的数据，其中包含了 1 115 家商店在 2013

年 1 月 1 日到 2015 年 7 月 31 日所产生的 1 017 210 条数据以及 1 115 条商店信息数据，销售数据和店铺数据的变量及含义如表 1 和表 2 所示。目标是 2015 年 8 月 1 日到 2015 年 9 月 17 日各商店的销售值。实验在 Google 的 Colab 云平台与 Python 3.7 环境下完成，预测评价指标采用均方根百分误差(Root Mean Square Percentage Error，RMSPE)，其计算公式为：

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

式中：y_i 为真实值；y'_i 为预测值。

表 1 销售数据信息的变量名及其含义

变量名称	含义
Store	每个商店的唯一 ID
DayOfWeek	周几
Date	日期
Sales	一个店铺一天的总销售值
Customers	一个店铺一天中的顾客数量
Open	取值为 0 代表商店关闭，为 1 代表商店开放
Promo	当天是否有促销：0 表示无，1 表示有
StateHoliday	取值为 a 代表公共假日，为 b 代表复活节假期，为 c 代表圣诞
SchoolHoliday	学校是否放假：0 代表不放假，1 代表放假

表 2 店铺数据信息的变量名及其含义

变量名称	含义
Store	每个商店的唯一 ID
StoreType	商店类型：取值为 a、b、c、d
Assortment	品种等级：取值为 a、b、c
CompetitionDistance	距离最近的竞争对手商店的距离/m
CompetitionOpenSinceMonth	最近的竞争对手开店月份
CompetitionOpenSinceYear	最近的竞争对手的开店年份
Promo2	商店是否参与连续促销：0 表示不参与，1 表示参与
Promo2SinceWeek	商店参与连续促销的周时间
Promo2SinceYear	商店参与连续促销的年份
PromoInterval	参加连续促销的月份

2.1 特征相关性及重要性分析

将销售值包含在内，实验数据一共有 18 个特征，涵盖了促销、竞争对手、节假日、商店、商品、顾客等信息。本文将日期分解为年、月、日，将商店类型、商品类型、假日类型采用实数或者独热(one-hot)编码来进行处理。在经过数据预处理后，对 19 个特征进行相关性分析，其结果如图 2 所示。

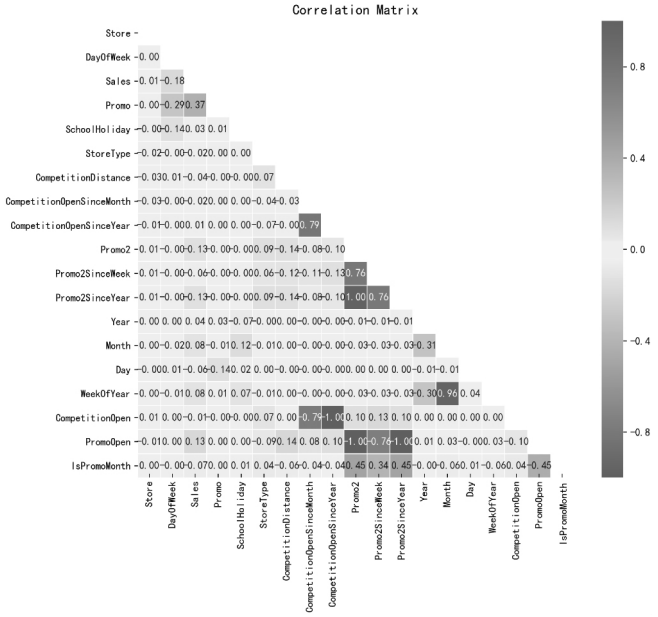
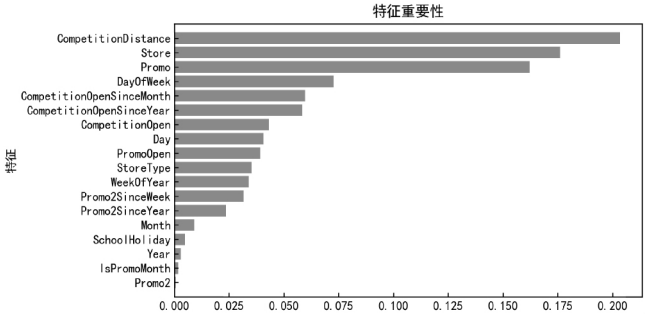
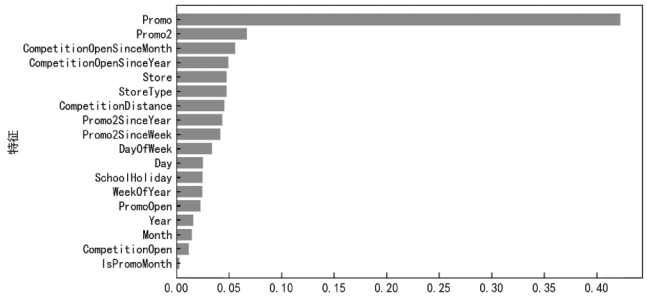


图 2 特征相关性分析

可以看出，Promo2SinceYear 与 Promo2、WeekOfYear 与 Month 的相关度很高。这是因为 Promo2SinceYear 是根据 Promo2 的时间来计算的，而 WeekOfYear 和 Month 是通过日期分解得到的。其他特征之间的相关性都不高，因此可以采用这些特征向量来作为输入。以销售值为目标变量，使用 Random forest 以及 XGBoost 算法进行建模，建模完成后，模型输出的各特征的评分排序结果如图 3 所示。



(a) XGBoost 模型特征重要性排序



(b) Random forest 模型特征重要性排序

图 3 特征重要性分析

可以看出，在 XGBoost 算法中，各特征的差异比较明显，模型能够为每个特征分配更好的权重，而 Random forest 中除了促销特征重要性最高外，其他特征重要性差异不明显。这也是 XGBoost 模型比 Random forest 模型取得更好预测性能的原因。除此之外，两个模型输出的前五名特征中，竞争对手、促销以及商店都囊括在内，这也证明了此次实验中特征选取的有效性。

2.2　基于随机搜索的超参数优化

由于模型中存在一些需要人为调整的超参数，如果采取每个参数都单独调整再观察模型在验证集上的预测性能的方法来进行超参数优化，那么所需要的时间成本太高，而且通常需要有经验丰富的算法工程师来进行这项任务。因此，本文使用一种随机采样交叉验证的方法来进行超参数优化，与网格搜索对比，随机搜索采取搜索各超参数在参数组合空间上的分布，从而能够在选取更优的参数组合的前提下，比网格搜索考虑更少参数组合数量。另外，随机搜索能够在不影响性能的前提下添加参数节点。实验中指定各算法的参数列表，并用 10 折交叉验证后的均方根百分误差(RMSPE)来评估各组合。各模型最终的超参数以及预测误差如表 3 所示。

表 3　单模型随机搜索优化后的超参数选择和误差

模型名称	超参数集	单模型误差
RF	树的个数 128，树的最大深度 20，内部节点的最少样例数 10	0.258
LR	L2 正则化	0.367
KNN	邻居数量 5	0.287
XGBoost	树深度 10，学习率 0.05，数的个数 500，最小叶子节点样本数 2，随机采样的比例 0.08	0.150

对比各模型的均方根百分误差可以发现，XGBoost 模型的误差最低。由于 RMSPE 是在验证集上采集的，证明了 XGBoost 模型有更好的泛化能力，因为 XGBoost 算法将损失函数进行了二阶泰勒展开，使得模型训练更充分。

2.3　模型预测误差相关性分析

对于 Stacking 模型而言，基层不同模型之间的差异性越高，则元学习器能够改善的空间就越大，整个模型的预测性能就越高。因此在选择基学习器后，需要考察各基学习器的预测误差相关性，尽可能选择差异性高的算法。在四份验证集中随机抽取一份数据，让各基学习器在该数据集上做出预测，将预测结果合并在同一数据框内，采用 Pearson 相关系数衡量各基学习器之间的相关性，结果如图 4 所示。

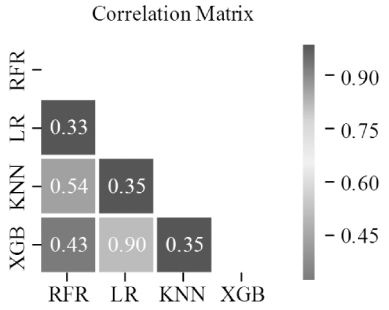


图 4　各模型预测误差相关性分析

可以看到，除了 XGBoost 与 LR 之外，各算法的误差相关性都很低，这是因为在有些异常值上 XGBoost 算法和 LR 算法在某些数据上呈现出了同样的趋势，但是总体的相关性低，这说明 Stacking 模型有待于取得更好的性能。

综上所述，最终选择 RF、LR、KNN、XGBoost 作为 Stacking 集成模型的基学习器来完成实验。

2.4　Stacking 模型预测性能分析

为了验证 Stacking 模型是否能取得比其他模型更好的预测性能: 首先在实验中随机抽取一个商店，对比单一模型和 Stacking 模型在该商店上的预测值与真实值的比较，其结果如图 5 所示。

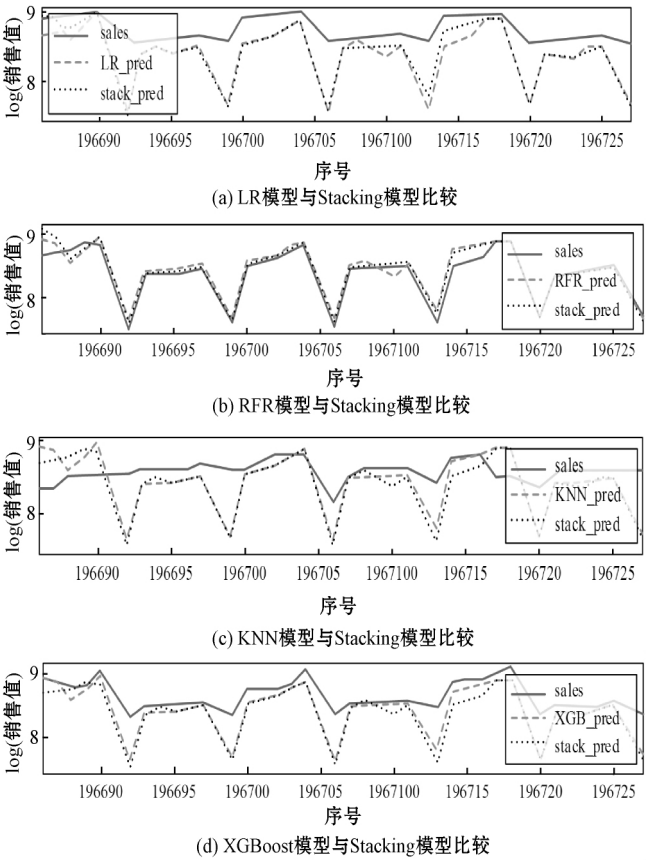


图 5　单模型与 Stacking 模型的预测值与真实值对比

可以看到，Stacking 模型取得了比单一模型更好的预测性能，而且在一些异常值上，Stacking 模型也能够处理得很好，这也证明了 Stacking 模型充分发挥了各算法的优势。通过从算法层面上的结合，改善了各算法预测效果较差的部分并且能够避免算法陷入局部最小点的缺陷。

元模型即第二层 XGBoost 算法的特征重要性如图 6 所示。可以看到，各基学器对在 Stacking 模型中的权重，其中 XGBoost 算法的贡献度最高，而其他学习器的贡献度差异不大，这说明在 Stacking 模型中，XGBoost 算法的性能是整个模型的预测性能得以改善的主要原因，同时也证明了 XGBoost 模型在解决大样本的回归问题时有着较高的鲁棒性和泛化能力。

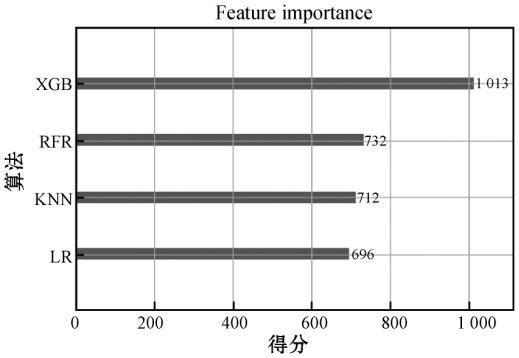


图 6 基学习器对在元模型中的贡献度

为了比较 Stacking 模型与单个模型在整个测试集上的测性能，表 4 展示了以 RMSPE 为衡量标准的各模型的精度。可见，Stacking 模型的 RMSPE 相比单个预测模型中预测性能最好的 XGBoost 模型降低了 1.8%，相比单个预测模型中预测性能最差的 LR 模型降低了 23.5%。对于大型企业来说，每提高 1% 的精度都能降低大量的安全库存或者减少缺货损失。

表 4 Stacking 模型与单一模型精度对比

模型	测试集误差/%
RandomForest	25.8
LinearRegeresion	36.7
KNN	28.7
XGBoost	15.0
Stacking	13.2

3 结 语

本文针对单个模型在大样本的销售预测上不能取得很好的泛化能力的问题，建立了多模型融合下的 Stacking。该模型取得了比单一模型更好的预测性能和泛化能力，在零售企业在库存管理、经营管理、供应链管理中有较高的应用价值。由于 Stacking 模型总的框架比较复杂，总体训练时间较长，未来可以考虑将模型在分布式环境下进行计算。还可以进一步地研究使用 Stacking 集成学习方法来处理实时数据，并根据 Stacking 集成学习方法来研发一套从数据获取到知识获取的数据处理系统，这势必会有更高的应用价值。

参 考 文 献

[1] 王林. 指数平滑法在配件销售预测中的应用[J]. 物流科技，2019，42(3)：45－48.

[2] Grasman J，Marcel K. Forecasting product sales with a stochastic bass model[J]. Journal of Mathematics in Industry，2019，9：2.

[3] 姜晓红，曹慧敏. 基于 ARIMA 模型的电商销售预测及 R 语言实现[J]. 物流科技，2019，42(4)：52－56.

[4] Loureiro A L D，Migueis V L，Da S L F M. Exploring the use of deep neural networks for sales forecasting in fashion retail[J]. Decision Support Systems，2018，114(10)：81－93.

[5] Duan Z K，Liu Y Q，Huang K Y. Mobile phone sales forecast based on support vector machine[J]. Journal of Physics：Conference，2019，1229：12061.

[6] Fan Z P，Che Y J，Chen Z Y. Product sales forecasting using online reviews and historical sales data：A method combining the Bass model and sentiment analysis[J]. Journal of Business Research，2017，74(5)：90－100.

[7] 黄鸿云，刘卫校，丁佐华. 基于多维灰色模型及神经网络的销售预测[J]. 软件学报，2019，30(4)：1031－1044.

[8] 张文雅，范雨强，韩华，等. 基于交叉验证网格寻优支持向量机的产品销售预测[J]. 计算机系统应用，2019，28(5)：1－9.

[9] 王锦，赵德群. 遗传 BP 神经网络在超市大米日销售预测中的应用[J]. 信息与电脑(理论版)，2018，415(21)：47－49.

[10] 罗嗣卿，刘璐. 改进 K-means 算法对大兴安岭蓝莓干销售预测的应用[J]. 黑龙江大学自然科学学报，2017，4(2)：139－144.

[11] Timmermann A G. Forecast combinations[J]. CEPR Discussion Papers，2006，29(5)：135－196.

[12] 常晓花. 基于 Adaboost 的随机森林算法在医疗销售预测中的应用[J]. 计算机系统应用，2018，27(2)：202－206.

[13] 叶倩怡. 基于 XGBoost 的商业销售预测[J]. 南昌大学学报(理科版)，2017，41(3)：275－281.

[14] Wolpert D H. Stacked generalization[J]. Neural Networks，1992，5(2)：241－259.