

# 多种算法融合的产品销售预测模型应用<sup>①</sup>



张雷东<sup>1,2</sup>, 王嵩<sup>2</sup>, 李冬梅<sup>2</sup>, 朱湘宁<sup>2</sup>, 焦艳菲<sup>3</sup>

<sup>1</sup>(中国科学院大学 计算机控制与工程学院, 北京 100049)

<sup>2</sup>(中国科学院 沈阳计算技术研究所, 沈阳 110168)

<sup>3</sup>(沈阳高精数控智能技术股份有限公司, 沈阳 110168)

通讯作者: 张雷东, E-mail: ucaszld@163.com

**摘要:** 销量预测一直是一个热点研究的课题, 对于各个企业有着重要的意义. 近年来, 随着深度学习的崛起, 用于销量预测的模型越来越多, 而单一模型的预测性能往往不够理想, 所以出现了越来越多的组合模型. 本文利用 Stacking 策略将 XGBoost、支持向量回归 (Support Vector Regression, SVR)、GRU 神经网络作为基础模型, 然后将 LightGBM 作为最终的预测模型, 并且融合了新的特征. 集中了几种模型的优势, 大大提高了模型的预测性能, 更加接近真实的销量数据, 为回归预测提供一种新的预测方法.

**关键词:** 销量预测; Stacking 算法; 集成学习; 特征工程; 梯度提升树

引用格式: 张雷东, 王嵩, 李冬梅, 朱湘宁, 焦艳菲. 多种算法融合的产品销售预测模型应用. 计算机系统应用, 2020, 29(9): 244-248. <http://www.c-s-a.org.cn/1003-3254/7550.html>

## Application of Product Sales Forecast Model Based on Multiple Algorithm Fusion

ZHANG Lei-Dong<sup>1,2</sup>, WANG Song<sup>2</sup>, LI Dong-Mei<sup>2</sup>, ZHU Xiang-Ning<sup>2</sup>, JIAO Yan-Fei<sup>3</sup>

<sup>1</sup>(School of Computer and Control Engineering, University of Chinese Academy of Sciences, Beijing 100049, China)

<sup>2</sup>(Shenyang Institute of Computing Technology, Chinese Academy of Sciences, Shenyang 110168, China)

<sup>3</sup>(Shenyang Golding NC Technology Co. Ltd., Shenyang 110168, China)

**Abstract:** Sales forecasting has always been a hot research topic and has great significance for all enterprises. In recent years, with the rise of deep learning, there are more and more models for sales forecasting, and the performance of single models is often not ideal. Therefore, there are more and more combinatorial models. In this study, we use Stacking strategy to support XGBoost, Support Vector Regression (SVR), GRU neural network as the basic model, then lightGBM as the final prediction model, with new features are merged. The advantages of several models are condensed, which greatly improves the prediction performance of the model, good enough to be more close to the real sales data, and provide a new prediction method for regression prediction.

**Key words:** sales forecast; stacking algorithm; ensemble learning; feature engineering; gradient boosting tree

## 1 引言

随着经济全球化的逐渐加深, 企业之间的竞争日益激烈. 并且当下随着电子商业的飞速发展, 每天都有上万笔交易在互联网上完成, 这样就产生了海量的数据. 企业如果能利用这些数据, 从数据中挖掘出用户购

买的一些规律, 就能更好的在这个市场中占据有利的位置. 销量预测一直是一个重点研究的问题, 一个准确的销量预测可以为企业控制成本, 提前规划市场和部署战略. 这个问题解决的好就能为企业的未来规划提供重要决策的理论支持.

① 收稿时间: 2019-12-16; 修改时间: 2020-01-14; 采用时间: 2020-03-11; csa 在线出版时间: 2020-09-04

目前为止,销量预测的研究方法有自回归移动平均法(ARMA)、线性回归、灰色系统理论等.基于时间序列的预测方法也被越来越多的学者采用,郑琰等基于时间序列的商品预测模型研究<sup>[1]</sup>、顾涵等支持向量归等在高频时间序列中的研究中<sup>[2]</sup>以及Fan ZP等在预测产品销售时<sup>[3]</sup>,采用这样方法都得了良好的效果.随着深度学习的崛起,又有了长短记忆网络(LSTM)这样优秀的预测算法,葛娜等利用 Prophet-LSTM 组合模型对销售量进行预测研究<sup>[4]</sup>.除此之外,张彤等利用 LSTM 的该并模型 GRU 结合 LMD 形成了一种新的预测方法<sup>[5]</sup>.往往将传统的线性模型和深度学习算法相结合,通常比单一算法表现优秀,或者结合以上某两种算法.例如冯晨等基于 XGBoost 和 LSTM 加权组合模型在销售预测的应用<sup>[6]</sup>,表示不同模型的组合,很大的提高了模型的泛化能力.

集成学习就是这样的一种策略,集中几种模型的优势,大大提高模型的性能. Stacking 模型堆叠就是一种集成学习的思想,集成几种模型的优势.盛杰等利用

Stacking 算法对恶意软件进行检测<sup>[7]</sup>,包志强等将传统的 Stacking 算法进行了改进,提出了一种熵权法的 Stacking 算法<sup>[8]</sup>,明显提高了算法的预测性能.

## 2 预测算法核心内容

### 2.1 预测算法框架

整个预测算法框架如图 1 所示.

(1) 将整个数据集分为训练集和测试集.

(2) 对所有的特征进行 Pearson 相关系数计算,筛选掉冗余的特征,选择相关性的特征,作为模型训练的特征向量.

(3) 将 SVR、GRU、XGBoost 作为 Stacking 的基础模型,利用特征向量进行训练.

(4) 将历史信息、均值、价格变化、评价分析和 Stacking 第一层的输出以及原始的特征进行特征融合,形成最终预测模型的输入.

(5) 将 lightGBM 作为 Stacking 算法的次级预测模型,进行训练并且在测试集上进行预测,得到结果.

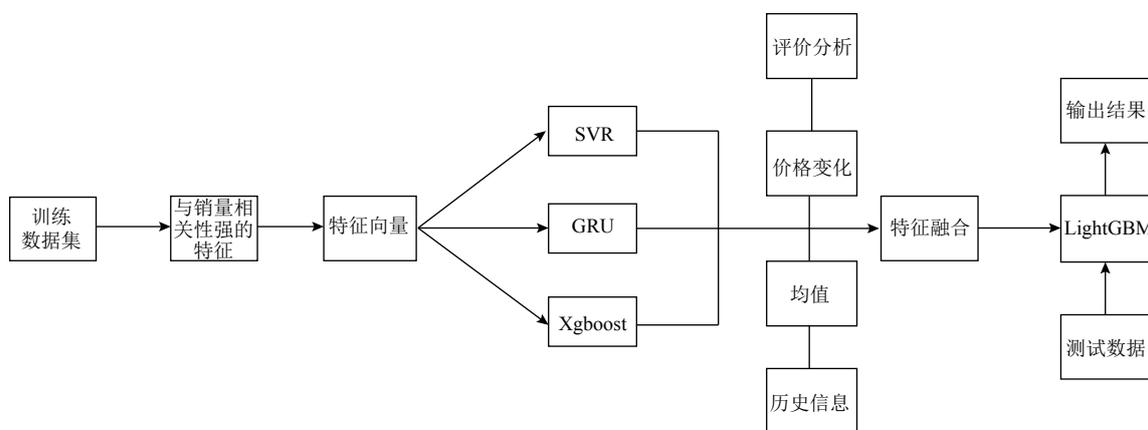


图 1 预测算法框架图

### 2.2 Stacking 算法

Stacking 算法简单理解就是几个简单的模型,一般采用将它们进行 K 折交叉验证输出预测结果,然后将每个模型输出的预测结果合并为新的特征,并使用新的模型加以训练.

Stacking 通常分为很多层,每一层的输出作为下一层的输入.但是伴随着层数的增多,模型的复杂度也在增加,训练起来也会慢很多.所以通常采用两层结构.在第一层时,将 K-1 份数据作为训练集进行训练,将剩

下的一份用来预测.这样 K 轮下来以后,将 K 轮的预测结果和原始数据进行拼接就形成了新的训练集.然后,第二层的模型利用新的数据集进行训练.最终预测,输出结果.如图 2 所示.

在本文中,将 SVR、GRU、XGBoost 作为第一层的预测模型,lightGBM 作为第二层的预测模型.具体训练过程如下:

(1) 将处理好的训练集平均分为 5 份数据量大小相等的的数据  $D_i (i \in 1, 2, 3, 4, 5)$ ;

- (2) 在每一轮训练中, 对于每一训练模型, 例如 SVR, 在 4 份数据上进行训练, 剩下一份用来预测得到  $P_i$ ;
- (3) 将第一层预测模型的输出  $P_i$  和原始的特征进行拼接, 得到最终的训练数据集  $D$ ;
- (4) 利用最终模型 lightGBM 在训练集  $D$  上进行训练;
- (5) 最后在测试集上进行预测, 得到最终的预测结果 Prediction.

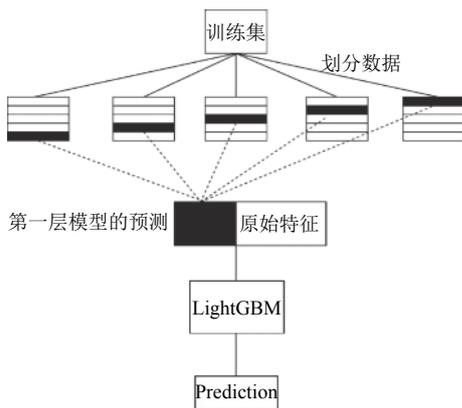


图2 Stacking 算法结构图

### 2.3 LightGBM 算法

LightGBM 是 Boosting 集合模型中的新进成员, 由微软提供, 它和 XGBoost 一样是对 GBDT 的高效实现. LightGBM 通过使用 Gradient-based One-Side Sampling (GOSS) 和 Exclusive Feature Bundling (EFB) 两种方法来解决特征选择的问题, 进而加速训练速度. 伪代码如算法 1 及算法 2.

算法1. GOSS算法伪代码

输入:  $I$ : training data,  $d$ : iterations,  $a$ : sampling ratio of large gradient data,  $b$ : sampling ratio of small gradient data,  $loss$ : loss function,  $L$ : weak learner

```

models ← {}, fact ←  $\frac{1-a}{b}$ 
topN ←  $a \cdot \text{len}(I)$ , randN ←  $b \cdot \text{len}(I)$ 
For  $i=1$  to  $d$  do
    preds ← models.Predict( $I$ )
     $g \leftarrow loss(I, preds)$ ,  $w \leftarrow \{1, 1, \dots\}$ 
    sorted ← GetSortedIndices(abs( $g$ ))
    topSet ← sorted[1: topN]
    randSet ← RanomPick(sorted[topN: len( $I$ )], randN)
    usedSet ← topSet+randSet
     $W[\text{randSet}] \ast = fact$  ▷ Assign weight fact to the
    Small gradient data
    newModel ←  $L(I[\text{usedSet}], -g[\text{usedSet}], w[\text{usedSet}])$ 
    Models.append(newModel)
    
```

算法2. EFB算法伪代码

```

输入:  $F$ : feature,  $K$ : max conflict count
输出: bundles
Construct graph G
searchOrder ← G.sortByDegree()
Bundles ← {}, bundlesConflict ← {}
for  $i$  in searchOrder do
    needNew ← True
    for  $j = 1$  to len(bundles) do
        cnt ← ConflictCnt(bundles[j],  $F[i]$ )
        if  $cnt + \text{bundlesConflict}[i] \leq K$  then
            bundles[j].add( $F[i]$ ), needNew ← false
            break
    if needNew then
        Add  $F[i]$  as a new bundle to bundles
    
```

### 2.4 数据预处理

- (1) 对于缺失值和由于促销引起的销量数据激增, 采用这个月销量的均值代替.
- (2) 将标签特征利用 one-hot 编码转化为 0 或 1 的特征.

### 2.5 特征工程

特征工程的结果对模型对着模型效果改善有着重要的作用. 特征工程通常用来减少冗余的特征, 提高模型训练的速度以及准确率. 经常用的特征选择的方法有 Pearson 相关系数、互信息和最大信息系数、皮尔森卡方检验、距离相关系数等.

#### 2.5.1 特征选择

在这里选用 Pearson 相关系数, 这是比较简单使用的特征选择方法. 这种方法能够反映特征和预测值之间的线性相关关系, 结果值范围在  $[-1, 1]$ ,  $-1$  表示该特征和预测值完全是负相关,  $0$  表示没有线性相关的关系,  $1$  则表示该特征值和预测值正相关. 其计算公式如下:

$$\rho_{(X,Y)} = \frac{\text{cov}(X, Y)}{\sigma_X \sigma_Y} \tag{1}$$

在式 (1) 中  $\text{cov}(X, Y)$  表示两个变量的协方差, 其计算公式如下:

$$\text{cov}(X, Y) = \sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y}) \tag{2}$$

在式 (1) 中  $\sigma_X$  表示变量  $X$  的标准差, 计算公式如下:

$$\sigma_X = \sqrt{\sum_{i=1}^n (X_i - \bar{X})^2} \tag{3}$$

### 2.5.2 特征构造

(1) 按照一级类型、二级类型统计该类型产品月销量的均值。

(2) 提取当前月的前第一个月、前第二个月、前第三个月、前第四个月、前年该月的销量数据。

(3) 加入商品的价格变化。

(4) 商品的评价分析: 对评论数据进行分词, 然后构造情感词典, 计算每条评论的情感值, 最后累加每天的评论情感值。

## 3 实验与结果分析

本实验数据来源于一个氨纶制造的企业. 数据主要包括从2014年1月到2018年1月每一天的历史销量数据. 数据属性主要有: 产品id、产品类型、产品等级、产品用途、产品价格、重量、日期等。

### 3.1 评价指标

(1) *RMSE* 均方根误差

均方根误差是预测值与真实值偏差的平方与观测次数  $n$  比值的平方根. 计算公式如下:

$$RMSE(X, h) = \sqrt{\frac{1}{n} \sum_{i=1}^n (h(x_i) - y_i)^2} \quad (4)$$

(2) *MAE* 平均绝对误差

*MAE* 是真实值与预测值的差值的平方然后求和平均. 计算公式如下:

$$MAE = \frac{1}{n} \sum_{i=1}^n (h(x_i) - y_i)^2 \quad (5)$$

(3) *MAPE* 平均绝对百分比误差

$$MAPE = \frac{100\%}{n} \sum_{i=1}^n \left| \frac{h(x_i) - y_i}{y_i} \right| \quad (6)$$

### 3.2 特征选择

利用 Pearson 相关系数, 筛选掉一些相关性弱的特征. 最终选择了产品id、产品等级、价格、产品类型、产品用途、颜色、材质、是否有促销活动、所属的年月等属性。

### 3.3 实验结果

#### 3.3.1 模型参数设置

利用 XGBoost 和 LightGBM 单模型在训练集训练模型时, 使用 Skit-learn 机器学习库中的 GridSearchCV

方法, 对模型的参数使用交叉验证的方式去探索最优的参数. 核心参数如表1和表2所示。

表1 LightGBM 参数表

参数	值
boosting_type	gbdt
objective	binary
metric	mae
n_estimators	700
num_leaves	10
max_depth	4
min_data_in_leaf	0.7
learning_rate	0.01
max_bin	30
feature_fraction	0.8
bagging_fraction	0.7
bagging_freq	30
min_split_gain	0.1
reg_lambda	0.1
reg_alpha	0.01

表2 XGBoost 参数表

参数	值
n_estimators	550
min_child_weight	5
max_depth	4
gamma	0.1
subsample	0.7
colsample_bytree	0.7
learning_rate	0.1
reg_lambda	1
reg_alpha	1

SVR 模型涉及的核心参数主要有3个: 选用的核函数、惩罚因子以及核系数. 通过 Grid Search 实验验证最佳的参数见表3。

表3 SVR 参数表

参数	值
kernel	rbf
C	2.0
gamma	1.0

#### 3.3.2 预测结果分析

在训练集上训练单模型以及组合模型, 最终在测试集上进行预测. 将预测结果绘制成折线图, 得到的结果图对比如下。

为了更加具体的比较出各个模型预测性能之间的差异, 采取 *RMSE*、*MAE* 以及 *MAPE* 3个评判指标进行对比. 具体的结果对比见表4。

表4 各模型预测性能对比

指标	RMSE	MAE	MAPE
SVR	86.7551	86.322	0.094
GRU	48.530	48.161	0.0517
XGBoost	69.256	68.516	0.075
LightGBM	59.869	59.677	0.0613
Stacking	35.582	35.484	0.036

从图3中可以看出SVR预测结果最差, LightGBM要比XGBoost好一些. 组合模型和SVR、XGBoost、LightGBM以及GRU相比, 性能都有所提升. Stacking组合模型的性能最优. 因此, 基于Stacking集成策略的组合模型确实能提升算法的预测性能. 在做回归问题时可以使用Stacking方式集中几个模型的优势, 可以为实验准确度的提升带来一定的帮助.

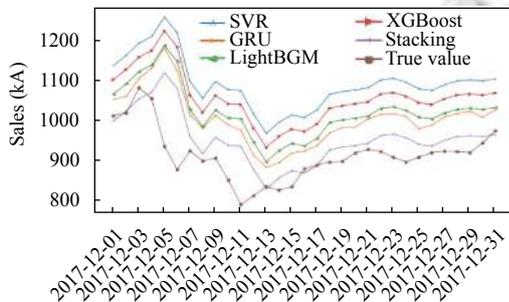


图3 各个模型最终预测效果图

在加入历史信息、均值、价格变化以及评价分析这些新的特征以后, 和没有进行特征融合的算法预测结果进行对比. 将预测结果绘制成图, 如图4所示.

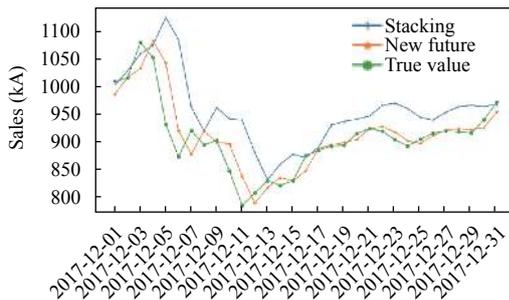


图4 特征融合之后预测结果对比图

按照评价标准进行计算后, 具体结果见表5.

表5 模型融合后的预测性能对比

指标	RMSE	MAE	MAPE
Stacking	35.582	35.484	0.036
new future	22.123	21.064	0.020

从预测效果图上可以看出, 在加入新的特征之后, 算法预测性能有明显提高. 从3个评价指标的数值上

也可以看出, 特征融合之后的算法预测的MAPE提升到了2%. 也证明这些新的特征确实发挥了作用.

#### 4 总结

本文使用Stacking策略结合了LightGBM、GRU、SVR、XGBoost几种经典的解决回归问题的算法提出了一种多种算法融合的产品销售预测模型, 并且结合数据做了实验. 实验证明这种多种算法融合的产品销售预测模型的预测结果要比这些单一模型的预测结果更加接近真实数据. 并且在引入了历史信息、均值、价格变化以及消费者的评价分析等这些新的特征后, 算法的预测性能又有了进一步的提升.

最终实验表明, 通过集成学习能够有效提高氨纶产品销量的预测性能. 为回归问题的提出一种新的思考方式和解决办法. 同时, Stacking策略也使得整个组合模型的复杂度变高, 训练速度变得缓慢, 这也是进一步要尝试改进的地方. 同时未来工作中, 将进一步尝试结合不同新的模型来做回归类问题的研究.

#### 参考文献

- 1 郑琰, 黄兴, 肖玉杰. 基于时间序列的商品需求预测模型研究. 重庆理工大学学报(自然科学), 2019, 33(9): 217-222.
- 2 顾涵. 支持向量回归在高频金融时间序列中的应用研究 [硕士学位论文]. 南昌: 江西财经大学, 2019.
- 3 Fan ZP, Che YJ, Chen ZY. Product sales forecasting using online reviews and historical sales data: A method combining the Bass model and sentiment analysis. Journal of Business Research, 2017, 74: 90-100. [doi: 10.1016/j.jbusres.2017.01.010]
- 4 葛娜, 孙连英, 石晓达, 等. Prophet-LSTM组合模型的销售量预测研究. 计算机科学, 2019, 46(S1): 446-451.
- 5 张彤, 徐晓钟, 王晓霞, 等. 基于改进LMD与GRU网络的短期燃气负荷预测. 计算机系统应用, 2018, 28(6): 29-37. [doi: 10.15888/j.cnki.csa.006938]
- 6 冯晨, 陈志德. 基于XGBoost和LSTM加权组合模型在销售预测的应用. 计算机系统应用, 2019, 28(10): 226-232. [doi: 10.15888/j.cnki.csa.007091]
- 7 盛杰, 刘岳, 尹成语. 基于多特征和Stacking算法的Android恶意软件检测方法. 计算机系统应用, 2018, 27(2): 197-201. [doi: 10.15888/j.cnki.csa.006183]
- 8 包志强, 胡啸天, 赵媛媛, 等. 基于熵权法的Stacking算法. 计算机工程与设计, 2019, 40(10): 2885-2890.