执御供应链需求预测

彭文鼎[1] 杨伯文[1]

[1]（北京航空航天大学 计算机学院）

Zhiyu Supply Chain Prediction

Wending Peng[1], Bowen Yang[1]

[1]（*School of Computer Science and Engineering*, *Beihang University*）

Abstract In the e-commerce production chain, in order to enhance the user logistics service experience, the supply chain cooperates to prepare the goods in advance in the local warehouses in various markets around the world, which can effectively reduce the logistics time and greatly enhance the user experience. Unlike the domestic e-commerce logistics situation, oversea goods production and sales areas of e-commerce products are global. Commodity procurement, transportation and customs inspection take much longer time. With the help of artificial intelligence and big data technology, the forward period can be predicted. The supply chain demand, thus optimizing the supply chain solution. In this competition, we made use of the platform to accumulate the most recent one year of commodity data forecast 45 days after 5 weeks per week (week1 ~week5) sales. First through the data Pre-processing, we filtered out the necessary columns and rows as the training data set. Then integrating the external statistical information with feature engineering to integrate the required statistical and non-statistical features. Finally, the algorithm model is designed for different types of features, which is passed into XGBoost, Facebook's Prophet models and statistical rule-based methods. We also tried to simply blend the models to take advantage of all of them. The training diversity achieved a better generalization effect. In the end, we achieved rank 56 in the preliminary round. The results also show that our model can effectively predict the supply chain demand near the difficult festival, which has practical significance.

Keywords supply chain prediction; feature engineering; XGBoost; Prophet; Time series analysis

摘要 在电商产业链中，为提升用户物流服务体验，供应链协同将货品提前准备在全球各个市场的本地仓，可有效降低物流时间，极大提升用户体验. 不同于国内电商物流情况，出海电商的产品生产和销售地区是全球化的，商品的采购，运输，海关质检等，整个商品准备链路需要更长的时间. 而借助人工智能和大数据方面的技术，可以预测远期的供应链需求，从而优化供应链方案. 本次比赛中，我们做的就是运用执御平台积累最近 1 年多的商品数据预测 45 天后 5 周每周（week1~week5）的销量. 首先通过数据预处理，筛选出必要的列和行作为训练的数据集. 然后通过特征工程结合外部信息，整合出需要的统计和非统计特征. 最后针对不同的类型特征，分别设计了算法模型，其中用到的有 XGBoost, Facebook 的 Prophet 和基于统计规则的方法，并且尝试对模型进行简单的融合以利用他们的训练多样性取得更好的泛化效果. 最终我们取得了初赛 56 名的成绩，结果也表明我们的模型可以有效预测出难度较大的节日附近的供应链需求，具有现实意义.

关键词 供应链需求分析；特征工程；XGBoost；Prophet；时间序列分析

浙江执御信息技术有限公司是一家专注出海的跨境电商企业，利用移动互联网创新和大数据应用，助力中国制造升级，将中国和全球的优质品牌、设计、产品输送到“一带一路”沿线国家和地区。在电商产业链中，为提升用户物流服务体验，供应链协同将货品提前准备在全球各个市场的本地仓，可有效降低物

流时间，极大提升用户体验。不同于国内电商物流情况，出海电商的产品生产和销售地区是全球化的，商品的采购，运输，海关质检等，整个商品准备链路需要更长的时间. 而借助人工智能和大数据方面的技术，可以预测远期的供应链需求，从而优化供应链方案. 而我们队参加本次比赛，最终取得了初赛第 56 名的成绩. 本文介绍我们的参赛方法以及参赛过程中的心得体会，主要内容为:

1) 数据探索分析与可视化
2) 特征工程
3) 算法模型与结果
4) 总结与感想

1 赛题分析

本次比赛是由执御公司联合 CCF 举办的.从业务的角度分析，在大数据和人工智能技术快速发展的新时代背景下，运用大数据分析和算法技术，精准预测远期的商品销售，为供应链提供数据基础，将能够为出海企业建立全球化供应链方案提供关键的技术支持.

具体比赛中对原问题做建模问题简化: 考虑商品在制造，国际航运，海关清关，商品入仓的供应链过程，实际的产品准备时长不同，而这里是统一在 45 天内完成，供应链预测目标市场为沙特阿拉伯. 赛题为运用平台积累最近 1 年多的商品数据预测 45 天后 5 周每周（week1~week5）的销量，具体为: 2017 年 3 月 1 日至 2018 年 3 月 16 日数据给定，来预测 2018 年 5 月 1 日，5 月 8 日，5 月 15 日，5 月 22 日，5 月 29 日起 5 周的销量. 但是其实在之后的分析中会发现，要预测的时间段前那中间一个多月的数据缺失，会给实际预测带来较大的困难和麻烦.

另外，为了更好的从业务方面理解比赛，加上我们本身不了解电商 or 供应链方面，我们在开始着手比赛之前，还去查找了更多的关于执御公司和电商供应链方面的背景知识. 经过一番调研发现，最初执御卖的商品主要以女性服装为主，在 2013 年确定的市场方向主要是美国和澳大利亚. 但获得投资之后，执御开始调整策略并转型. 基于对以往销售数据的分析，和对未来市场的判断，执御加大了在中东市场的投入，可以说是 “一带一路” 政策的坚定支持者. 2015 年，销售额上升到 10 亿人民币，是 2014 年的 10 倍. 2016 年，执御对商品做了品类扩充，增加了男性商品，如服装、鞋包、配饰，还有 3C 类目、女性美妆类目. 此外，执御还经常和网红合作. 因此，综合这些信息，我们认为它可以说是中东版的蘑菇街，唯品会，而不是之前简单的认为就是淘宝. 这些信息咋一眼看似乎对于比赛而言没有什么价值，但是后面可以看到，它给了我们从商品的特征工程方面的一些比较重要的启示，比如判断一个款式的 sku_id 在它所述的商品的 goods_id 里是不是所谓的 “爆款商品” 等等，而这些挖掘出来的商品特征则对最后销量的预测起到一定的作用.

比赛评判的 Evaluation: RMS Error (评测的 score = 1/(1+rmse)), 因此做实验的 loss 用 l2 loss 就好了.

2 数据分析与可视化 (EDA)

在本节中，我们主要介绍我们队对数据集的探索分析(EDA)以及可视化的过程. 因为这个比赛的数据量很大, 且有多个表结构比较复杂, 所以就很有必要先对数据清洗和处理, 才方便之后用来的分析. 毕竟 garbage in, garbage out.

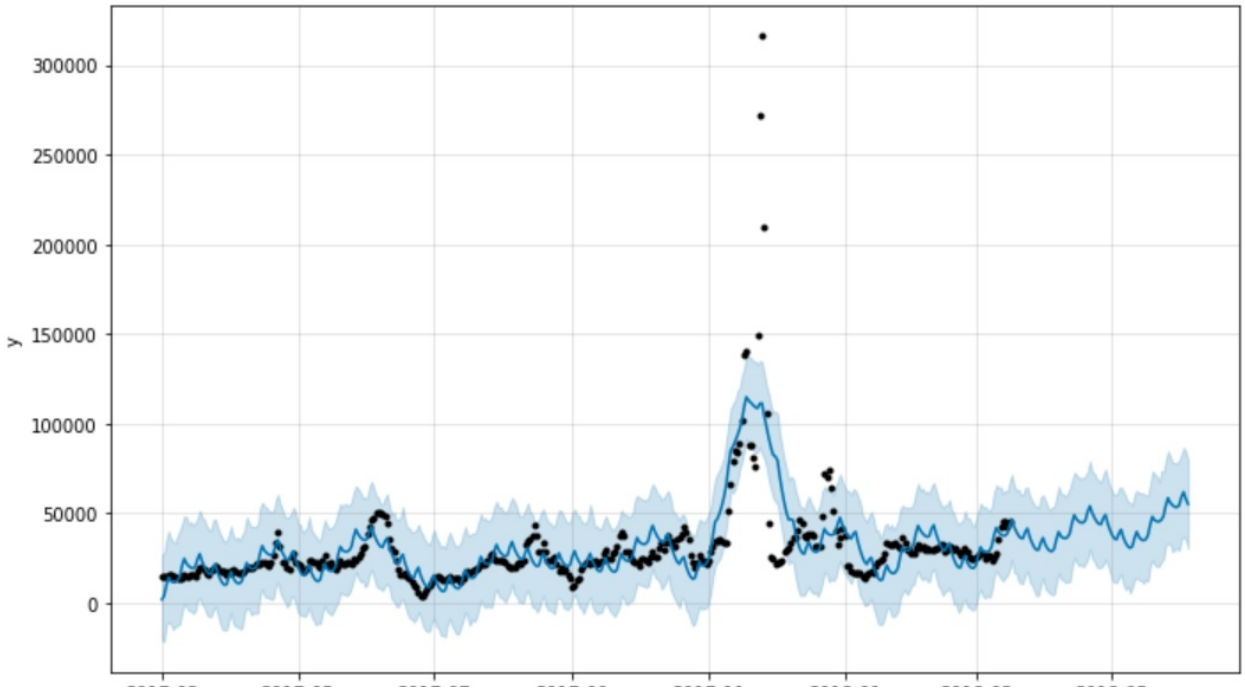
拿到数据, 我们做的第一件事是宏观的角度去理解它, 从而对于数据有高屋建瓴的印象和把握. 而数据清洗总体来说, 包括缺失值处理, 异常值处理和数据的归一化. 但是这些不是都是必须的, 而需要我们通过观察数据的具体情况来决定要做哪些工作和避免哪些无用功.

首先遇到的一个问题是数据集比较大, 有几个表都是几百万和几千万行级别的 .csv 格式数据. 这给我们参加比赛所需要快速迭代更改模型带来了的问题就是很慢, 而即使是把 pandas 的 read_csv()里的 low_memory 改成 False, 可以加快读取的速度, 但是这样做导致的新问题就是电脑会很卡甚至 jupyter notebook 跑着跑着就可能崩. 这里我们发现了一个小的 trick, 就是用硬盘空间换时间, 先第一次读 csv 文件, 之后每次都存成 feather 文件. 顾名思义, feather 是羽毛, 很轻, 是专门设计用来可以每次快速 on-disk 读取的数据格式. 于是虽然这样我们电脑硬盘里多了 5G 左右的 feather 文件, 但是每次读取速度都会快很多, 为每次迭代算法提升了宝贵的时间效率. 类似的在实操中遇到问题->查找解决方案->学到的实用小 trick 还有不少, 这里就不一一列举了.

同样, 各个表的数据之间关系也不是很明确, 这里我们就需要进行大量手工操作, 用各种 pandas 里的类似 SQL 的合并表 (join/merge) 的操作, 从而整合出清晰的且有联系的列(这也是这个比赛给我们的一大收获: 熟练了 pandas 里各种 data manipulation 操作).

而通过对数据的可视化分析, 我们可以看出商品销售的一些特点. 比如历史商品销量统计中，发现 11 月 25 日出现了巨大的波峰，销量总计为 64 万左右. 且前后销量均增大，存在异常，推测为节假日，我们猜测为平台购物节, 类似国内的天猫双十一, 双十二这种超大型活动.

而至于数据清洗, 我们发现表本身很少缺失值, 可能主办方他们设计题目的初衷就是这样, 已经做了很多清洗的工作. 但是如果要按照时间每天排列做成 pandas Series 而不是 Dataframe, 那倒是会有很多缺失, 比如绝大部分天数对应的销量都是 0. 不过就我们需要的数据而言, 这些数据集给我们的是足够(甚至过大)而不是缺失. 因此就没有必要做太多缺失值处理, 异常值处理和数据归一化的工作. 反而我们在之后需要筛选一些来用, 因为太多了以至于都很多列都不知道怎么用…



商品销售波动于时序预测比较, 可以看出 11 月多出现巨大增幅, 在没有节假日信息的模型被视为异常而无法被拟合到

3 特征工程

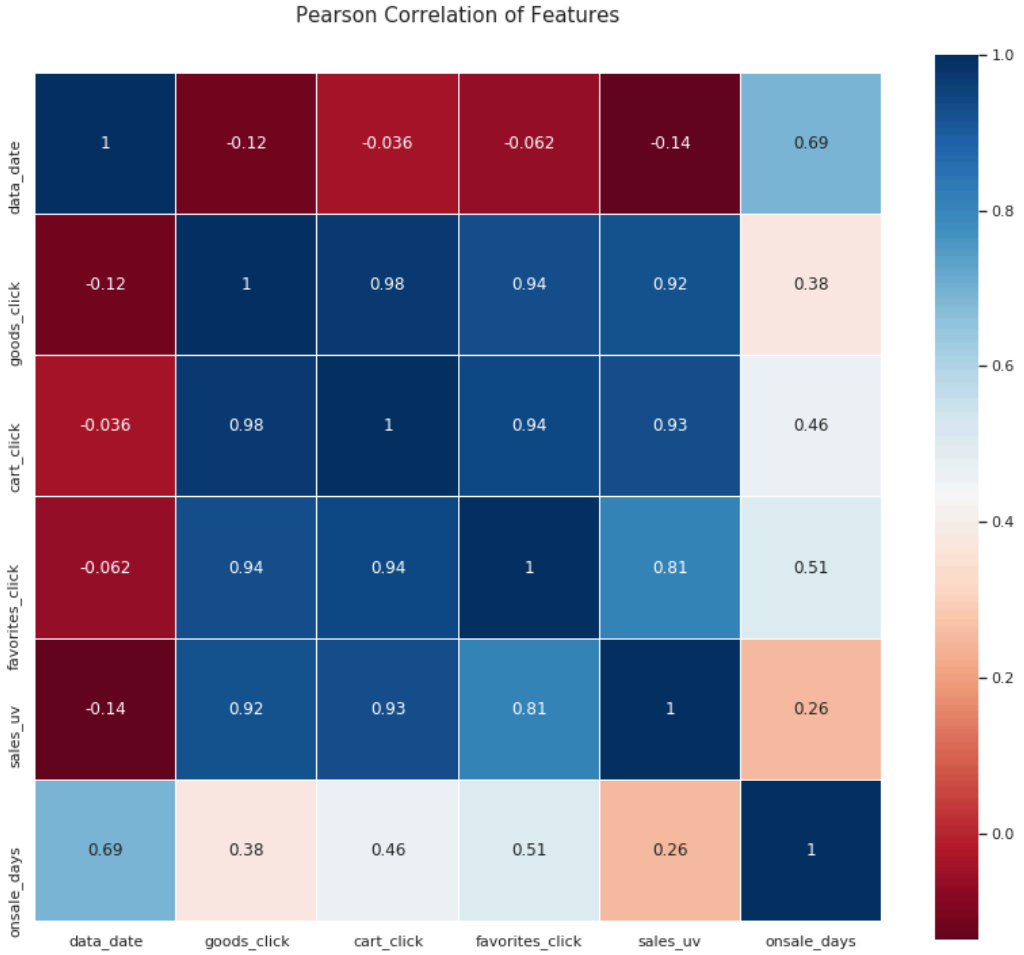
在本节中，我们介绍如何根已经经过预处理的数据来构造模型训练所需要的特征.

首先，特征的目的是给模型使用的. 所以构造前模型的需求是什么， 必须要很清晰. 但由于第一次比赛， 第一次做特征工程尤其是时序模型相关的特征和怎么把时序问题转化为普通的回归问题， 我们其实对于这个问题上的特征工程具体要得到什么特征不是很清晰，因此也走了不少弯路.

首先我们考虑可能可以用到的外部信息. 赛题里已经提示了我们可以考虑宗教和天气， 并且通过搜索在 kaggle 的一个很类似的 Rossmann Store Sales 比赛里看到 Discussion 讨论版有人说可以用 google trend， 这里当然也可以用(比如最先想到的就是直接看执御这个关键词在沙特区域的搜索情况，当然还可以搜索各种商品相关的关键词)， 甚至还有沙特和中国的政治关系变化， 比如有什么外交事件等等...但是这些信息都是要从外部爬取， 由于比赛本来时间就十分有限而我们队伍也只有 2 人， 因此再写爬虫处理额外的数据不太现实就没有做. 所以我们的特征工程主要目标还是找一些比较宏观可以给我们设计算法模型有用的信息， 以及从数据本身挖掘出来一些简单的比如统计学方面的特征.

这里主要介绍和商品本身属性有关而与时间无关的的统计学方面的特征工程. 而对于时序模型，本身就是提取时间相关的特征， 而商品的本身属性特征却用不上， 这个在之后的基于 Prophet 训练的时序模型里具体介绍.

具体特征方法为: 在这里数据集由很多个表构成， 我们把它们合并整合后， 只留下与商品或者特定款式有关的信息. 在对这些按不同时间粒度滑窗，在特定的时间窗口求平均值， 方差， 最小值最大值等等， 这样就得到了基础统计特征， 比如: goods 点击次数最值均值方差，goods 收藏次数最值均值方差等等，这些特征就有几十列了.



商品本身特征，比如商品种类，季节属性.

商品销量特征， 对于长时间段分析， 一个 sku 在商品销量的比例和排名， 可以代表这个商品或者这个款式是不是爆款，还是基本款.

然后， 在之后对于特征重要性通过 XGBoost

feature importance 排序后发现，其实对很多商品而言，最重要的特征还是 marketing，也就是主办方数据集里已经提供的特征，平台商品促销与否. 这也不难理解，符合我们的直观感受.

但是后来发现，这样提取特征和对时间粒度的划分其实没有很好的抓住关键要点，也就是需要预测的和我们构造的相关性. 比如选取数据统计值方面，没有选取到和测试集足够接近的时间段，导致效果也不是很好. 这也是我们反思值得改进的地方.

另外我们查找到了斋月有关的信息，这是一个非常关键的点. 在沙特当地，斋月的时候，商家白天不营业，因此像执御这样的电商商品销量就会有大幅度的增长，而且为了过节会有像我们国内过春节前类似的大采购，为节日做准备. 而 2018 年斋月的时间就是 5 月 16 日-6 月 16 日，可以看出赛题主办方的良苦用心，就是为了让我们预测一个相对比较难的时间—节日前和节日期间—的销量，看看我们的模型是不是足够的鲁棒.

4 基本算法模型与结果

我们主要尝试了 XGBoost，Prophet 和基于统计规则的算法. 最后提交的时候还尝试了融合各个模型. 这里融合就是简单的线性配比(blending). 本节将详细介绍这几种算法模型以及他们的结果.

Model1: XGBoost

梯度提升树(GBDT)已经在实践中证明可以有效地用于分类和回归任务的预测挖掘. 之前人们所选择的提升树算法一直都是 MART（multiple additive regression tree）. 但从 2015 年开始，一种新的且总是获胜的算法浮出了水面：XGBoost. 这种算法重新实现了树提升，并在 Kaggle 和其它数据科学竞赛中屡获佳绩，因此受到了人们的欢迎，被称作大杀器，以至于很多人参加这种数据竞赛熟练到“上来就一把梭”的用 XGBoost. 当然由于我们是第一次参赛，没有什么经验，对它的算法原理也不了解. 抱着要用就得好好用的心态，我们先是深入学习了 GBDT 的算法以及 XGBoost 所做的一些改进的原理(包括调研了一下类似的 GBDT 开源实现库如 lightGBM 和 catboost 等等).

XGBoost 是在 GBDT 的基础上对 boosting 算法进行的改进，内部决策树使用的是 CART 回归树. 回归树的分裂结点对于平方损失函数，拟合的就是残差；对于一般损失函数（梯度下降），拟合的就是残差的近似值，分裂结点划分时枚举所有特征的值，选取划分点.

算法思想就是不断地添加树，不断地进行特征分裂来生长一棵树，每次添加一个树，其实是学习一个新函数，去拟合上次预测的残差。当我们训练完成得到 k 棵树，我们要预测一个样本的分数，其实就是根据这个样本的特征，在每棵树中会落到对应的一个叶子节点，每个叶子节点就对应一个分数，最后只需要将每棵树对应的分数加起来就是该样本的预测值.

$$\mathcal{L}(\phi) = \sum_i l(\hat{y}_i, y_i) + \sum_k \Omega(f_k)$$

$$\text{where } \Omega(f) = \gamma T + \frac{1}{2}\lambda \|w\|^2$$

xgboost 的残差 loss

参 数	说 明
max_depth	树的最大深度，较深的树容易过拟合，较浅的树容易欠拟合。
learning_rate	每个弱学习器的权重缩减系数，也称作步长，取值范围为[0,1]。对于同样的训练集拟合效果，较小的学习率意味着需要更多的弱学习器的迭代次数。
n_estimators	最大的弱学习器的个数。与学习率 learning_rate 一同考虑。
early_stopping_rounds	提前停止训练。给定一个验证集，该模型将开始训练，直到验证得分停止提高为止，防止过拟合。

XGBoost 的主要参数以及相关的说明

用 XGBoost 一大问题是调参. 一开始我们觉得它的调参很玄学，这么多参数怎么调呢？但是经过搜索和学习发现，这其实有固定的套路，比如一个博客[3]介绍的方法比较实用:

1. 先用高的 learning rate(0.1 左右)，用它来确定最优的 number of trees
2. 调 tree-specific 参数 (max_depth, min_child_weight, gamma, subsample, colsample_bytree)，这里也可以分两步，先调深度和子节点权值，再调其他的.
3. 调正则化参数：lambda，alpha

4. 最后在降低 learning rate，找适合的 number of trees

就没有时间去找合适的时间区间段来训练和验证了.

General Approach for Parameter Tuning

We will use an approach similar to that of GBM here. The various steps to be performed are:

1. Choose a relatively **high learning rate**. Generally a learning rate of 0.1 works but somewhere between 0.05 to 0.3 should work for different problems. Determine the **optimum number of trees for this learning rate**. XGBoost has a very useful function called as "cv" which performs cross-validation at each boosting iteration and thus returns the optimum number of trees required.
2. **Tune tree-specific parameters** (max_depth, min_child_weight, gamma, subsample, colsample_bytree) for decided learning rate and number of trees. Note that we can choose different parameters to define a tree and I'll take up an example here.
3. Tune **regularization parameters** (lambda, alpha) for xgboost which can help reduce model complexity and enhance performance.
4. **Lower the learning rate** and decide the optimal parameters .

这样的好处在于可以先快速找到大概的参数，再逐步细调，降低学习率使得拟合和泛化效果更好.

XGBoost 调参方法

而算法的具体流程如下:

1. 将数据集划分为训练集 X_train 和测试集 X_test.
2. 通过网格搜索(GridSearchCV)方法在训练集 X_train 样本上通过交叉验证的方法寻找最优参数集，同时通过验证集来控制是否早停.
3. 对 2 中搜索得到的参数集在全体样本上构造 XGBoost 模型并拟合

参 数	取值范围	适用模型
max_depth	[3, 5]	RF, Xgboost
learning_rate	[0.01, 0.1, 0.3]	Xgboost
n_estimators	[10, 20]	RF, Xgboost
early_stopping_rounds	50	Xgboost

XGBoost 在小训练集上做初步筛选后的参数范围

用 XGBoost 的另一大问题是速度，由于这个比赛的数据集比较大，加上调参需要多次的尝试，而且我们也整合了不少特征，所以模型的迭代速度不是很理想，自然效果也没有预期那么好，最后单模型是 0.085 左右. 分析原因除了上述，还有一个很重要的因素是我们对训练集和测试集的划分上没有很恰当.从时间上划分需要找到合适的趋势匹配要预测的趋势，但是直到比赛后期我们才通过多次提交尝试和时序模型大致分析出了每周的趋势，这时候

Model2 FBprophet 与应用分析

4.1 时间预测简介

时间序列预测一直是预测问题中的难点，人们很难找到一个适用场景丰富的通用模型。这是因为现实中每个预测问题的背景知识，例如数据的产生过程，往往是不同的，即使是同一类问题，影响这些预测值的因素与程度也往往不同，再加上预测问题往往需要大量专业的统计知识，这又给分析人员带来了难度，这些都使得时间序列预测问题变得尤其复杂。传统的时间序列预测方法通常有如下缺陷：

①适用的时序数据过于局限

②缺失值需要填补

③模型缺乏灵活性

④指导作用较弱

2017 年 2 月 24 号 facebook 开源了时间序列预测框架 prophet，目前支持 R 语言和 python 语言，托管在 github 上。prophet 是基于可分解（趋势+季节+节假日）模型的开源库，Prophet 充分的将业务背景知识和统计知识融合起来，它让我们可以用简单直观的参数进行高精度的时间序列预测，并且支持自定义季节和节假日的影响。官方号称“让普通人也能像数据分析师一样得出专业的结论”。

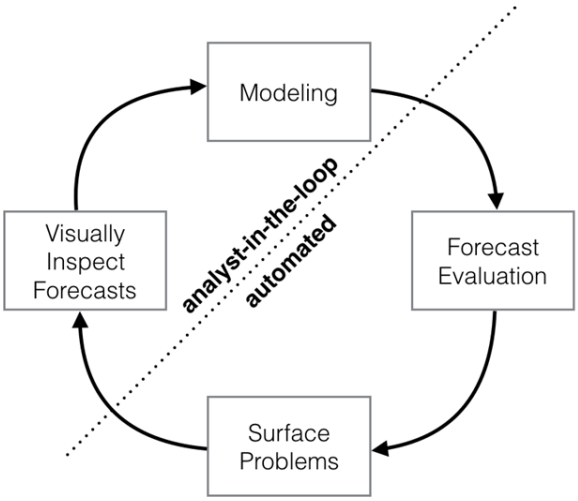
4.2 FBprophet 原理

时间序列模型可分解为三个主要组成部分：趋势，季节性和节假日。它们按如下公式组合：

$$y(t) = g(t) + s(t) + h(t) + \varepsilon_t$$

其中，g(t)表示用于拟合时间序列中的分段线性

增长或逻辑增长等非周期变化；s(t)为周期变化（如：每周/每年的季节性）；h(t)表示非规律性的节假日效应（用户造成）；et 为误差项用来反映未在模型中体现的异常变动。



FBprophet 框架流图

上图是 prophet 的整体框架，整个过程分为四部分：Modeling、Forecast Evaluation、Surface Problems 以及 Visually Inspect Forecasts。从整体上看，这是一个循环结构，而这个结构又可以根据虚线分为分析师操纵部分与自动化部分，因此，整个过程就是分析师与自动化过程相结合的循环体系，也是一种将问题背景知识与统计分析融合起来的过程，这种结合大大的增加了模型的适用范围，提高了模型的准确性。按照上述的四个部分，prophet 的预测过程为：

a.Modeling：建立时间序列模型。分析师根据预测问题的背景选择一个合适的模型。

b.Forecast Evaluation：模型评估。根据模型对历史数据进行仿真，在模型的参数不确定的情况下，我们可以进行多种尝试，并根据对应的仿真效果评估哪种模型更适合。

c.Surface Problems：呈现问题。如果尝试了多种参数后，模型的整体表现依然不理想，这个时候可以将误差较大的潜在原因呈现给分析师。

d.Visually Inspect Forecasts：以可视化的方式反馈整个预测结果。当问题反馈给分析师后，分析师考虑是否进一步调整和构建模型。

4.3 适用场景分析

当预测模型没有按预期运行时，我们希望针对问题来调整模型的参数。调整参数需要对时间序列的工作原理有全面的理解。例如 automated ARIMA 首先输入的参数是差分的最大阶数，自回归分量和移动平均分量。普通分析师不知道如何调整顺序来避免这种表现，这是一种很难掌握积累的专业知识。Prophet 包提供了直观易调的参数，即使是对缺乏模型知识的人来说，也可以据此对各种商业问题做出有意义的预测。

并非所有的预测问题都可以通过同一种策略解决。Prophet 是 Facebook 为所遇到的业务预测任务而优化的，这些任务通常具有以下特点：

①有至少几个月（最好是一年）的每小时、每天或每周观察的历史数据；

②有多种人类规模级别的较强的季节性趋势：每周的一些天和每年的一些时间；

③有事先知道的以不定期的间隔发生的重要节假日（比如国庆节）。

④缺失的历史数据或较大的异常数据的数量在合理范围内；

⑤有历史趋势的变化；

⑥对于数据中蕴含的非线性增长的趋势都有一个自然极限或饱和状态。

通过分析，我们要解决的问题正好符合 Facebook 设计算法的初衷，满足预测任务的基本要求：

①我们有执御公司一年的数据量，而且销售记录精确到每天，所以数据量上不成问题。

②背景为现实生活中的销售问题，销售的商品有的具备季节属性，比如冬季服装与夏季服装的区别；还有周内变量，人们在休息日的购物量一般大于工作日。

③有特定的人造购物节作为重大事件，而且日期已知结果确定，比如双十一或者年中购物节等。

④商品销量都是人的购物行为的结果，所以刨去个别的非理性行为，一般数据波动都是在合理范围内的。

⑤由于执御公司是一个起步仅三年的初创跨境电商公司，所以有明显的扩张与上升趋势。

⑥同理，购物行为近乎理性，而且商品总量有限，所以一定会有一个自然极限的。

综上，我们的问题场景恰好适用于这个算法，所以我们决定用其作为我们的时间序列预测问题的解决方案。

5 基于 FBprophet 的单一商品预测

5.1 以日销售额为单位

我们的问题根据 100K 种商品之前一年多的数据预测未来 35 天的销售量。

因为将问题考虑为时间序列进行分析，所以除了节假日信息、每日销售信息之外的数据就没有太多实际价值了。

第一种想法就是使用每个商品 365 天的销售数据，分别建立 100K 个数据模型，然后对每个数据进行单独预测，得到每类商品未来 35 天的日销售量。

个人感觉这种方式的效果应该不错，因为是对每一类商品都分别建模，所以计算量很大，通过统计学的方式将问题简化，同时进行预测，假如每个数据的浮动合理，结果就会很好。

FBprophet 的另一个有点就是拟合非常快，平均一个模型只需要 4 秒就能训练完成。

```
  0%|          | 1/2000 [00:03<1:51:39,  3.35s/it]INFO:fbprophet.forecaster:Disabling daily se
asonality. Run prophet with daily_seasonality=True to override this.
  0%|          | 2/2000 [00:06<1:51:26,  3.35s/it]INFO:fbprophet.forecaster:Disabling daily se
asonality. Run prophet with daily_seasonality=True to override this.
  0%|          | 3/2000 [00:10<1:51:27,  3.35s/it]INFO:fbprophet.forecaster:Disabling daily se
asonality. Run prophet with daily_seasonality=True to override this.
  0%|          | 4/2000 [00:13<1:51:39,  3.36s/it]INFO:fbprophet.forecaster:Disabling daily se
asonality. Run prophet with daily_seasonality=True to override this.
  0%|          | 5/2000 [00:16<1:51:55,  3.37s/it]INFO:fbprophet.forecaster:Disabling daily se
asonality. Run prophet with daily_seasonality=True to override this.
```

模型训练

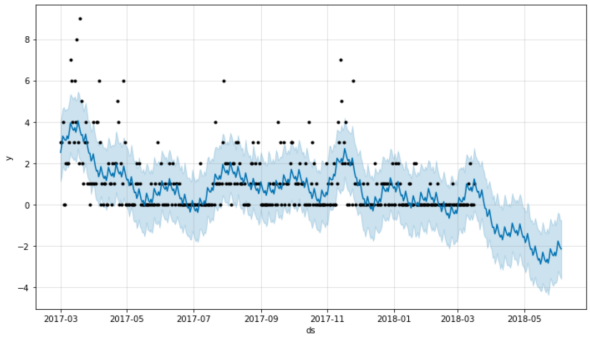
但是十万个模型算起来，也需要近 100 个小时的时间，所以计算力不足是面临的第一个问题。

算力不足就去借吧，感谢 Kaggle 的免费服务器，让我们可以以 50 倍的速度完成全部模型的训练。

多服务器并发

这样一来，加上人工修改模型，大约 4 个小时就可以完成一次训练。晚上心满意足的配置好 50 个服务器的工作，第二天早上起来满心欢喜的收菜，感觉这个问题解决了，可是事与愿违，结果并非如此。

出问题之后我们就对单个模型进行考察，随机挑选了几个商品的模型可视化之后便发现了问题。



低销量商品预测

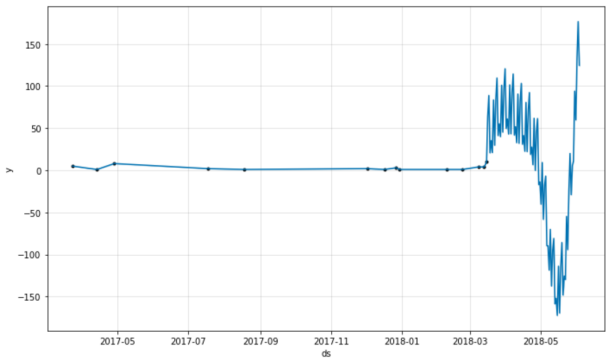
不难发现，这是一款不太受欢迎的商品。全年销售量低迷，平均大概只有全年三分之一到四分之一的日子里有售出，这样就导致商品的销售曲线呈波动幅度巨大的折线图，比如第一天卖出 2 份，第二天卖出 6 份，这就是 200%的增长额。第三天卖出 3 份，就又是 50%的下降。这样的商品难以预测，甚至根据趋势很可能出现负销售量的情形。

我们又分析了一下，100K 类的商品种数可不是一个小数目，生活中常用的商品大约 114 类，每一类就算 100 款商品有稳定的销售量，还有大约 90%的商品如上图所示，全年无人问津，所以如何解决这个问题成了重中之重。

5.2 以周销售额为单位

考虑到最终的预测仅仅要求我们给出每周的销售量，即 5*7=35 天的销售量，所以考虑处理数据将日销售额转化为周销售额。

通过计算 7 天内销售量之和形成周销售量使得数据变得稍微平滑，数据的可视化效果好了。但是同时带来另一个问题，就是数据量上的减少，以周为单位意味着每个模型只有约 50 条数据，而且个别促销周也使得周与周之间的差距变得更大。



周销售额预测

将日销售量预测模型改为周模型，导致数据量大量缺失，结果也不太好。

5.3 Slide Window 处理

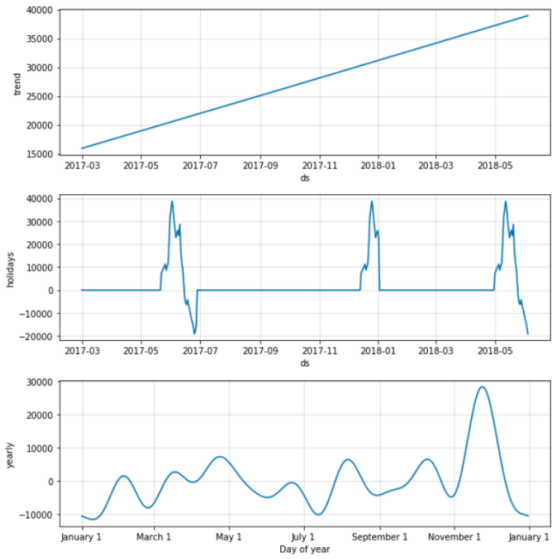
进一步改变思路，考虑图像处理中的 Sliding Window 的思想，将数据以固定区间进行滑动计算。这样让数据变得既平滑而且又保证数据量，效果有了一定的提升。

但是相当于用之前一段时间的平均去衡量下一天的销量，最终曲线过于平滑，数据与数据之间的梯度可能减少过多，导致最终预测效果也不好。

6 基于 FBprophet 和数理统计的整体预测

6.1 总体销售额趋势提取

经过了几次尝试之后我们发现，对每个商品的单独建模的效果并不是很好，因为太多商品数据型不够好，所以绝大多数模型会出现上下波动浮动太大的情况。考虑利用通过整体销量的预测，乘到每个商品单独的销售量基值上，以得到最终的销量。

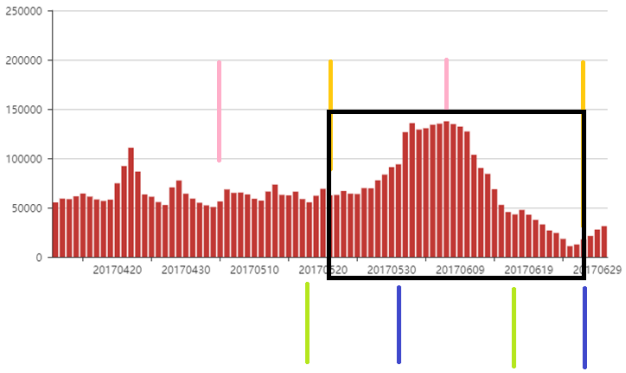


节假日与全年销量变化趋势

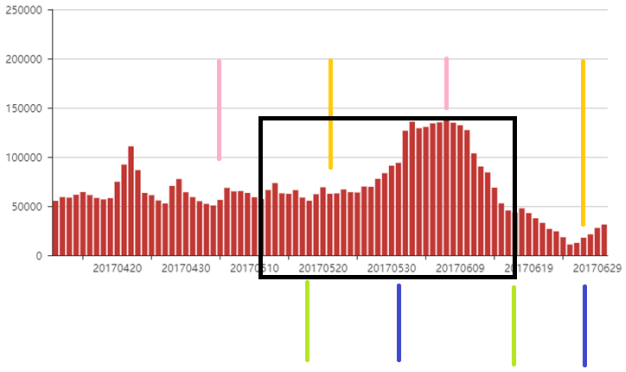
仔细分析，重新审视题目。为什么要预测这 35 天的数据？因为这五周是一个大型购物节，类似于双十一，但是活动力度并没有双十一大。所以，商家为了提前半个月预测之后的销售高峰以提前从国内进货，合情合理。

但是，为什么五月份的时候会有一个这么大型的促销活动呢，而且持续一个多月之久。通过查阅资料发现，这一个月正好在伊斯兰斋月期间，此时商家白天基本不营业，大家时间后移数个小时，正好是电商发力的好时机。所以，将斋月这一根本原因考虑进去同样重要。

下图中，上方双黄线间为 2017 年促销节日期，下方双蓝线间为 2017 年斋月日期，黑色框代表去年同期预测时间内的总销售量。

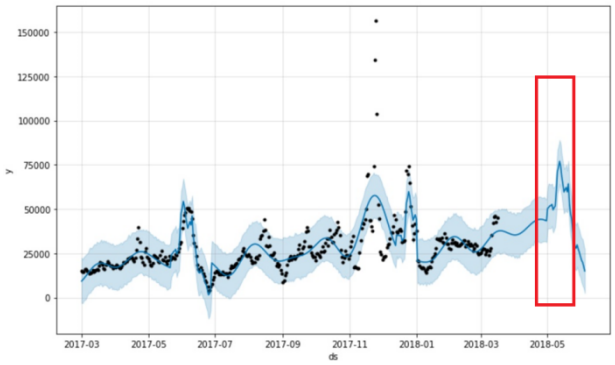


下图中，上方双粉线为 2018 年促销节日期，下方双绿线为 2018 年斋月日期，黑色框代表今年同期应该基于促销节与斋月的时间差做出平移后的预测区间。



这么做的原因是，去年销售节与斋月同时结束，而今年较斋月提前约十天开始，应该是商家发现促销节中大家的购物量最高，同时又是斋月伊始，需要提前通过网购囤积日用品导致。

所以基于此，我们将预测的区间选取提前约十天，得到最终的效果如下，呈现红框内的销售额趋势。



有了总量的趋势，我们考虑使用统计方法，利用最后的一个月的数据均值、权值递降、指数递降等方式作为基准值，分别乘以上面预测得到的趋势，得到最终结果。最后发现，以最近一天为 1，之前 30 天每隔一天乘 0.99 的指数权重的效果最好，成绩为 0.10837166。

7 总结

本文在这项商品销售额预测的比赛中，尝试了基于特征工程的 Xgboost 方法，尽可能多的利用举办方提供的数据，比如点击量、收藏量等可能存在潜在信息的数据，以回归模型的方式进行预测。同时尝试了基于时间序列的 FBprophet，尽可能排除无关信息的干扰，只利用最直接的商品日销售额和节日、促销因素作为预测的基础。

最终发现，由于用户相关的信息量巨大，导致特征工程困难，而且举办方并未提供预测期间相关性高的数据，如点击量等，使得回归模型的效果并不算太好。而由于大多数商品无人问津的原因，基于时间序列的对单一商品逐一预测的方式同样效果不佳。最后，对整体趋势的预测趋势加上数理统计得到的基准值协同预测的效果最好。

参 考 文 献

[1] Hastie, T.; Tibshirani, R.; Friedman, J. H. (2009). "10. Boosting and Additive Trees". The Elements of Statistical Learning (2nd ed.). New York: Springer. pp. 337&ndash, 384. ISBN 0-387-84857-6. Archived from the original on 2009-11-10.

[2] Chen and C. Guestrin. XGBoost: A scalable tree boosting system. InProceedings of the 22nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pages 785{794, San Francisco, CA, 2016.

[3] Complete Guide to Parameter Tuning in XGBoost (with codes in Python), https://www.analyticsvidhya.com/blog/2016/03/complete-guide-parameter-tuning-xgboost-with-codes-python/

[4] Sean J. Taylor , Benjamin Letham , Forecasting at Scale