大数据管理课程论文

本科生姓名 梁 智 鹏

学 科 专 业 管理科学与工程

研 究 方 向 Google Store需求量预测——Kaggle TOP 3%解决方案

指 导 教 师 张宏斌

中山大学岭南学院

2018年11月18日

目　　录

第一章　摘　　要 **3**

第二章　绪论 **1**

2.1 论文研究背景与意义 1
2.2 论文主要内容 2

第三章　Google Store需求量预测 **4**

3.1 数据情况 4
3.2 数据探索 6
3.2.1 目标变量 6
3.2.2 Device 6
3.2.3 Geography 8
3.2.4 Traffic 11
3.2.5 Vistor 12
3.2.6 异常情况 13
3.3 数据清洗 14
3.3.1 常数列剔除 14
3.3.2 异常值处理 15
3.3.3 缺失值探索 17
3.3.4 填充数值变量 17
3.3.5 填充类型变量 20
3.4 数据规约 22
3.5 基础模型 24
3.5.1 决策树回归模型 24
3.5.2 随机森林模型 24
3.5.3 LightGBM 26
3.5.4 神经网络 31
3.6 特征工程 32
3.6.1 时序特征 32
3.6.2 行为特征 34
3.6.3 类别性特征 35
3.7 模型构建 35
3.7.1 会话级别 36
3.7.2 用户级别 36

3.7.3 Stacking 37
3.8 结论 39

致谢 **42**

参考文献表 **43**

第一章 摘 要

用户每次访问线上商店都会带来大量的痕迹信息，对于商家而言，如果能够通过这些信息对用户进行精准画像，特别是预测他在当次访问预期会产生的消费量，就可以对每一位用户进行更加精准的营销，提高单次访问的消费量甚至提高访问次数和忠诚度。本文选取谷歌商城G Store中2016到2017一部分顾客访问的记录，包括顾客的设备信息、地理信息、访问时行为特征等，预测用户在一段时间内产生的有效消费量。为了便于模型预测，此处对消费量金额进行加一后取自然对数的变换。在特征构造层面，通过多阶段特征构造过程，即第一阶段直接从原始特征入手，从时序性特征、地理性特征、设备型特征等多维度进行有针对性的特征构建，通过加入“未来特征”、类别特征交互项，以及时区转换，实现异常值检测和处理、数据规约、特征规约，确保了模型的泛化能力和提高了训练速度。在初步模型构造层面，通过引入决策树、随机森林、xgboost、神经网络以及LightGBM等新型框架，分析比较它们在需求预测方面的优劣，并通过网格搜索展示了超参数的调优过程，并通过参数调优得到了显著的模型提升，为之后更为复杂的多模型调参工作提供帮助。本文最终提出两阶段需求预测模型，即在访问层次的需求预测模型之上，构造新的特征并作为第二阶段的模型特征输入，最后再次使用Stacking集成学习方法进一步捕捉如此大量特征之中丰富的信息，并使用简单的第二层模型确保模型的泛化能力，预测效果比起一阶段模型有了显著的提升。

关键词：需求预测; **集成学习**; **数据挖掘**; XGboost; **神经网络**; **网格搜索**

第二章　绪论

§2.1　论文研究背景与意义

在商业中普遍存在着80/20法则：即20%的用户贡献80%的利润。这反应在数据当中就会出现严重的数据不均衡问题，即难以捕捉到真正能够产生利润的客户，因此这对于营销团队来说，难以决定正确的营销对象，营销的效率就无法提升，这无疑是不利于企业在日渐激烈的商业竞争中取得优势的。更为重要的是，在供应链当中，需求的预测一直都是核心问题之一，因为这会影响到供应链的性能以及相应的设计机制。著名的“牛鞭效应”,就是揭示了需求的不确定性对于供应链中整体性能的巨大影响效应。即供应链上的一种需求变异放大现象，发生在信息流从最终客户端向原始供应商端传递时，无法有效地实现信息的共享，使得信息扭曲而逐级放大，导致了需求信息出现越来越大的波动，此信息扭曲的放大作用在图形上很像很一根甩起牛鞭。为了减少对于需求不确定性造成的不良影响，需求预测一直是供应链当中的核心话题之一,例如Sheu等人(2005)就提出了多层需求供应链反应模型来减少牛鞭效应，Fiala等人(2005)也总结了信息共享降低需求不确定性的方法。

随着大数据时代的到来，越来越多的企业开始有意识地收集数据并且拥有大量的数据。除了数据的量呈现飞速增长外，数据的维度也在快速增大。一方面为供应链管理提供了大量的数据支持，使得科学的决策方法特别是基于数据的方法得到了长足的发展。另外不少基于新特征新维度数据的方法也被证明是能提高管理效率和精度，例如Taylor等人(2003)就在预测电力需求的时候引入天气特征并通过实验结果表明其有效性。但另一方面，大量特征之间的关系还难以探明，尽管业界一直在探索有效的需求预测模型和方法，但是由于数据规模大、特征关系复杂，尤其是不同维度之间的特征的交互效应的存在，学界一直没有一套系统的分析方法。

而近年来随着人工智能方法的进步，数据挖掘作为一个应用机器学习、深度学习在数据分析的领域也取得了极大的进步，相比于传统的回归模型，神经网络、树模型等非线性的模型被广泛应用到数据挖掘当中，并且取得了突破性的进展。例如Tso等人(2007)就尝试运用回归模型、决策树模型和神经网络分别的电力消费量进行预测并进行了全面的模型比较。而Zhang等人(2005)也运用BP神经网络预测城市水资源需求预测。国内也已经开始了相关的研究，例如后锐等人(2005) 年提出了基于多层神经网络的区域物流需求方法。但是对此讨论目前还停留在较少的水平。而对于更为关键的特征构造方法的讨论，根据笔者的了解则更少。

因此出于提升供应链效率的角度出发，笔者在本文中尝试通过对网上购物的真实数据集进行探索分析，通过用户访问的特征包括行为特征、地理信息、设备信息等进行深入分析，通过特征工程构建出大量更具有业务意义的特征，并提出两阶段需求预测模型来提高需求精度，希望能够有助于提高供应链管理水平。

§2.2 论文主要内容

用户每次访问线上商店都会带来大量的痕迹信息，对于商家而言，如果能够通过这些信息对用户进行精准画像，特别是预测他在当次访问预期会产生的消费量，就可以对每一位用户进行更加精准的营销，提高单次访问的消费量甚至提高访问次数和忠诚度。因此本文主要研究通过用户的多维度特征，包括地理信息、设备信息、访问行为特征等信息构建消费量预测模型。

本文选取谷歌商城G Store 中2016年8月到2017年8月的顾客访问的记录，包括顾客的设备信息、地理信息、访问时行为特征等，预测用户在一段时间内产生的有效消费量。为了便于模型预测，此处对消费量金额进行加一后取自然对数的变换。在特征构造层面，通过多阶段特征构造过程，即第一阶段直接从原始特征入手，从时序性特征、地理性特征、设备型特征等多维度进行有针对性的特征构建，通过加入“未来特征”、类别特征交互项，以及时区转换，实现异常值检测和处理、数据规约、特征规约，确保了模型的泛化能力和提高了训练速度。

另外在第二阶段的初步模型构造层面，通过引入决策树、随机森林、xgboost、神经网络以及LightGBM等新型框架，分析比较它们在需求预测方面的优劣，并通过网格搜索展示了超参数的调优过程，使模型得到了显著的提升效果，为之后更为复杂的多模型调参工作提供帮助。本文最终提出两阶段需求预测模型，即在访问层次的需求预测模型之上，构造新的特征并作为第二阶段的模型特征输入，最后再次使用Stacking集成学习方法进一步捕捉360个特征之中丰富的信息，并使用简单的第二层模型确保模型的泛化能力，预测效果比起一阶段模型有了显著的提升。

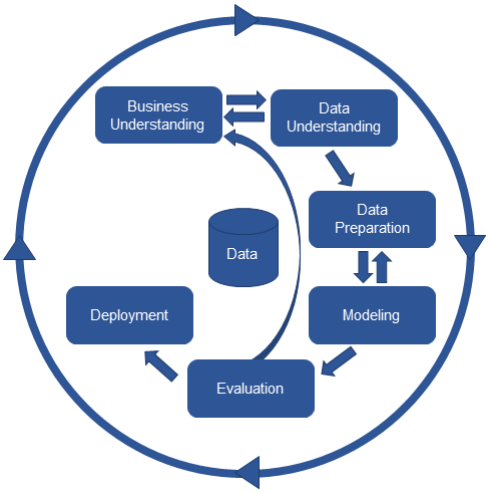

图 2.1 数据挖掘流程图

第一部分本文首先介绍问题定义以及数据基本情况，包括数据文件情况，数据的字段名及其含义等。

第二部分则是介绍数据预处理和基础特征工程的具体步骤，包括数据探索、数据清洗、数据规约以及特征工程。

第三部分则是进行两阶段模型训练和预测，并根据模型给出更加精确的特征重要性分析，并给出相应的管理启示。

第四部分则是总结与展望，总结本文的研究成果并提出研究不足，最后提出下一阶段的研究方向。

第三章　Google Store需求量预测

本次分析遵循数据挖掘基本流程，即定义问题，数据探索、数据清洗、基础模型预测，模型调优和整合并且最终展示结果。本文尝试通过顾客访问网上商场的行为特征（例如点击数）和地理位置、设备情况等对用户在一段时间内最终产生的购买量进行预测。

§3.1　数据情况

本数据为Kaggle平台上Google Analytics Customer Revenue Prediction比赛所用数据。Google Analytics Customer Revenue Prediction为Kaggle为2018年9 月10日新上线的一项数据挖掘比赛，一共持续两个月，北美时间11 月30 日00：00:00停止竞赛，因此在本文写作过程中比赛仍未停止。由比赛所提供的数据一共分为三个部分：训练数据train.csv，测试数据test.csv以及提交的样板sample_submission.csv。三个数据的结构以及大小如表：

数据集名称	数据条目数	字段数	文件大小
train.csv	903653	12	1.47G
test.csv	804684	12	1.31G
sample_submission.csv	617000	2	16M

表 3.1　文件情况

训练数据和测试数据中的12个特征分别为：channelGrouping,date,device,fullVistorId, geoNetwork,sessionId,socialEngagementType,totals,traffic Source,visitId。由于device,geoNetwork,totals三个字段是以字典的形式压缩了大量的字段数据，因此我们将其转换后得到真正的字段数据。转换后字段数有53个，对于从device,geoNetwork,totals中解压出来的字段，我们在其前加入解压原字段前缀。详细名称以及其具体含义如表3.2：

字段名	含义	补充说明
fullVisitorId	每个用户的唯一ID	
channelGrouping	访问商店的渠道(e.g.Organic Search)	
date	访问商店的日期	
sessionId	此次访问的唯一ID	
visitId	表示访问的唯一id	为了获得完整的ID，需要整合fullVistorId和visitId
visitNumber	本次为该用户的第几次访问	
visitStartTime	访问的开始时间	使用时间戳形式
device.browser	访问的浏览器	
device.deviceCategory	访问的设备类型	
device.isMobile	是否用移动端访问	
device.operatingSystem	操作系统	
geoNetwork.continent	居住的大陆	
geoNetwork.country	居住的国家	
geoNetwork.metro	网络所在的城市	缺失值用'not available in demo dataset'
geoNetwork.networkDomain	网络端	
geoNetwork.region	地区	缺失值用'not available in demo dataset'
geoNetwork.subContinent	子大陆	
totals.bounces	页面总跳出次数	取值为1和null。
totals.hits	会话包含的总匹配数	
totals.newVisits	是否首次访问该页面	取值为1和null
totals.pageviews	会话包含的网页浏览总数	
totals.transactionRevenue	交易收益（目标变量）	
trafficSource.adContent	以字符串形式表示的广告内容	
trafficSource.adwordsClickInfo.adNetworkType	看到广告所采用的渠道	取值为nan, 'Google Search', 'Search partners'三种
trafficSource.adwordsClickInfo.gclId	每次点击广告的唯一id	
trafficSource.adwordsClickInfo.isVideoAd	是否为视频广告	取值只有两个类型：nan, False
trafficSource.adwordsClickInfo.page	展示广告的页面在搜索结果中的页码	取值为nan等
trafficSource.adwordsClickInfo.slot	广告的展示位置	取值为[nan, 'Top', 'RHS']
trafficSource.campaign	指引到广告的源头	
trafficSource.campaignCode	campaign的编号	
trafficSource.isTrueDirect	表示用户在浏览器中是否输入了您网站网址的名称	取值为[nan, True]
trafficSource.keyword	在google store中搜过所用的关键词	缺失值用'(not provided)'表示
trafficSource.medium	媒体	取值为'organic', 'referral'等
trafficSource.referralPath	人们是通过什么地方来到这个商店的页面的	链接地址
trafficSource.source	资源	

表 3.2 完整字段名称

§ 3.2 数据探索

§ 3.2.1 目标变量

在开始一切数据探索之前，对于目标变量的探索是至关重要的。笔者主要关注浏览量与购买量的时间分布。从图3.1中，count代表数据集中出现的次数，而Non Zero Count则是代表数据集中非零消费量在该时间点上的频数。此处数据为训练集数据，时间跨度为2016 年8 月到2017 年8 月。从图中可以看出，在2016 年10 月到2016 年12 月浏览量具有一个明显的攀升又回落的过程，但是对于购买量来讲，其攀升过程则是发生在2016 年12 月到2017 年1 月之间，两者不同步，与我们常识认为的“浏览量会带来购买量”的观念不符合，也就意味着在两者之间的转换机制上存在着另外很重要的因素但没有被考虑进来。

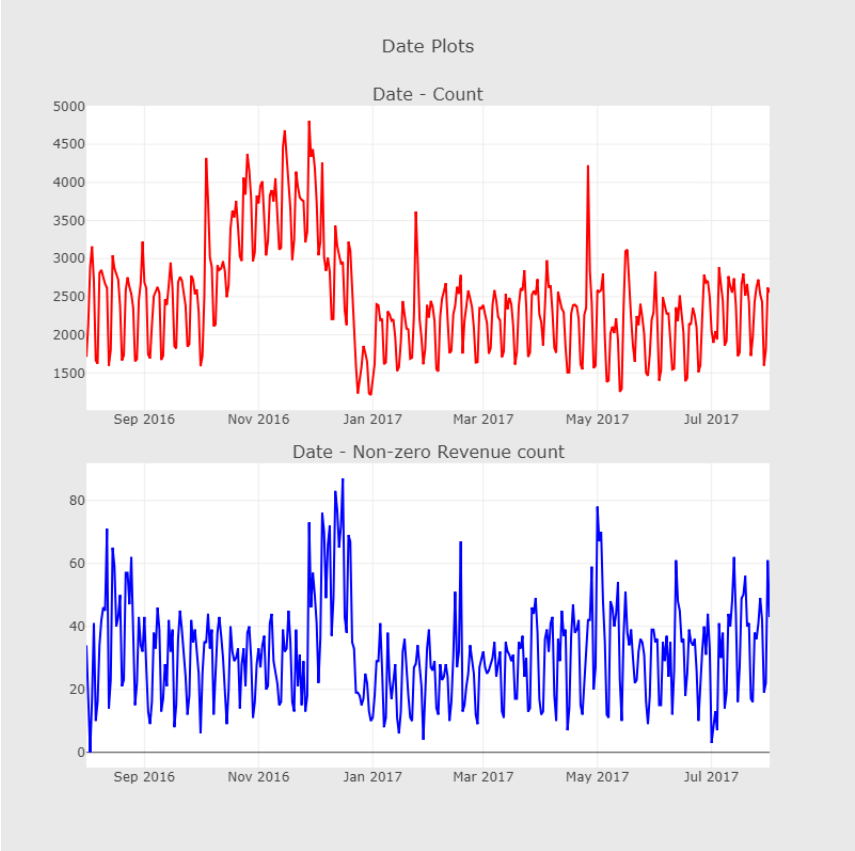

图 3.1 消费量的时间分布

§ 3.2.2 Device

考虑到在运营管理当中，用户的设备信息可能会隐含用户的个人收入水平信息，比如著名的如苹果用户浏览机票获得的机票价格一般会高于使用windows系统的客户。因此为了判断使用设备种类的不同对于判断用户需求信息是否有帮助，笔者分别观察了使用不同设备的人群的平均消费均值分布情况。但是由于消费需求严重不均衡，因此笔者另外再构造出

一个新的指标——购买指标，即如果消费大于0则为1，否则为0来判断顾客的消费意愿。结果见图3.2。Count仅仅表示所有顾客中使用该种设备的人数，Non-zero Revenue Count则是表示消费大于0 的顾客(下称有价值的用户)中使用这种设备的数量。Mean Revenue则是将统计使用该种设备的顾客的历史消费均值综合得到，利用三个维度下用户的分布情况可以判断消费非0的用户的使用设备特性，从而为后续的特征工程以及模型构建提供准备。从图中可以看出，在使用浏览器的种类上，在有价值的用户中使用safari的比重低于平均水平，firefox和其他浏览器也都出现类似现象，Chrome的用户使用两幅图中的最高比重，因此可以解释为使用谷歌浏览器的用户对于谷歌的产品具有一定的信任度和依赖度，因此也更容易在谷歌商店中发生购物行为。但是在消费均值上分布情况就大相径庭，使用firefox消费均值远超使用其他浏览器的用户，因此判断有可能是异常值的情况。

而在设备的种类上面，有价值的用户使用电脑端的比重依然超过平均水平，同时在消费水平均值分布上也保持第一，因此使用设备的顾客可以被认为更加有购买的意愿。

而观察设备系统可以发现，有价值的用户分布较于一般情况有显著差异。在有价值的用户中使用苹果系统（Macintosh）的占据了超过一半的比重。这也与常识相符——苹果产品一般价格较高，苹果产品的拥有者也一般为消费能力较强的人，另外使用Chrome OS系统和Linux系统在有价值的用户中也值得注意。

另外注意到not set的用户在三个维度下都没有表现出显著的购买意愿。Not Set用户是属于缺失值情况，即缺失值也能为预测带来重要信息。

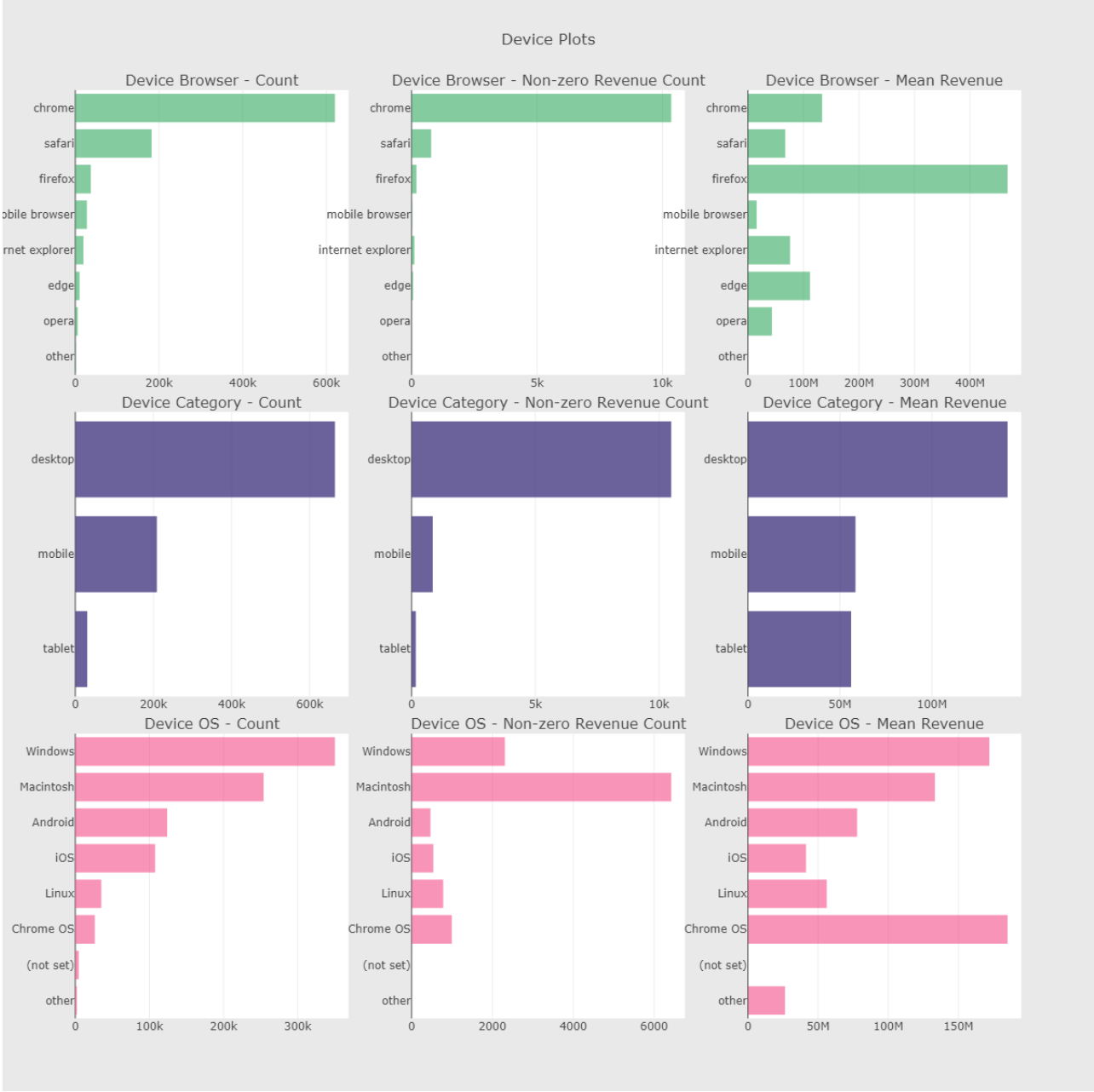

图 3.2 设备与浏览量和购买量的情况

§ 3.2.3 Geography

地理分布同样也是用户画像的重要分析维度。一般认为，居住在发达国家或地区的顾客具有较高的经济水平和较强的消费意愿也容易在浏览商店时发生交易行为。同样从上述的三个维度进行分析，我们发现谷歌商城在美国的浏览量和购买量都是最多的，在亚洲和欧洲也都有很高的浏览量，但是并不能转换为购买量。分析购买平均水平可以发现一个有趣的现象：非洲的人均购买金额远超所有大洲。这貌似与常识违背，但是考虑到谷歌商城在

非洲的浏览量不高，也就意味着知名度不高的情况下，能够主动登陆谷歌商城的人群在购买力、消费意愿上都与其他人群有着明显区别，这种区别比起在经济水平普遍较高的美洲、欧洲等都要大得多，因此凡是登陆谷歌商城的非洲用户都是带着强烈的购买意愿，也就导致了最高的消费平均水平。

再从亚大陆视角仔细分析需求量的来源，北美也就主要是美国、加拿大由于谷歌的知名度较高，因此有较高的浏览量和购买量。但是从平均消费金额数来看，西非和东非远超其他地区，原因也与上述分析类似。

但是从网络前缀则不能看出有价值的信息。因此此处分析略过。

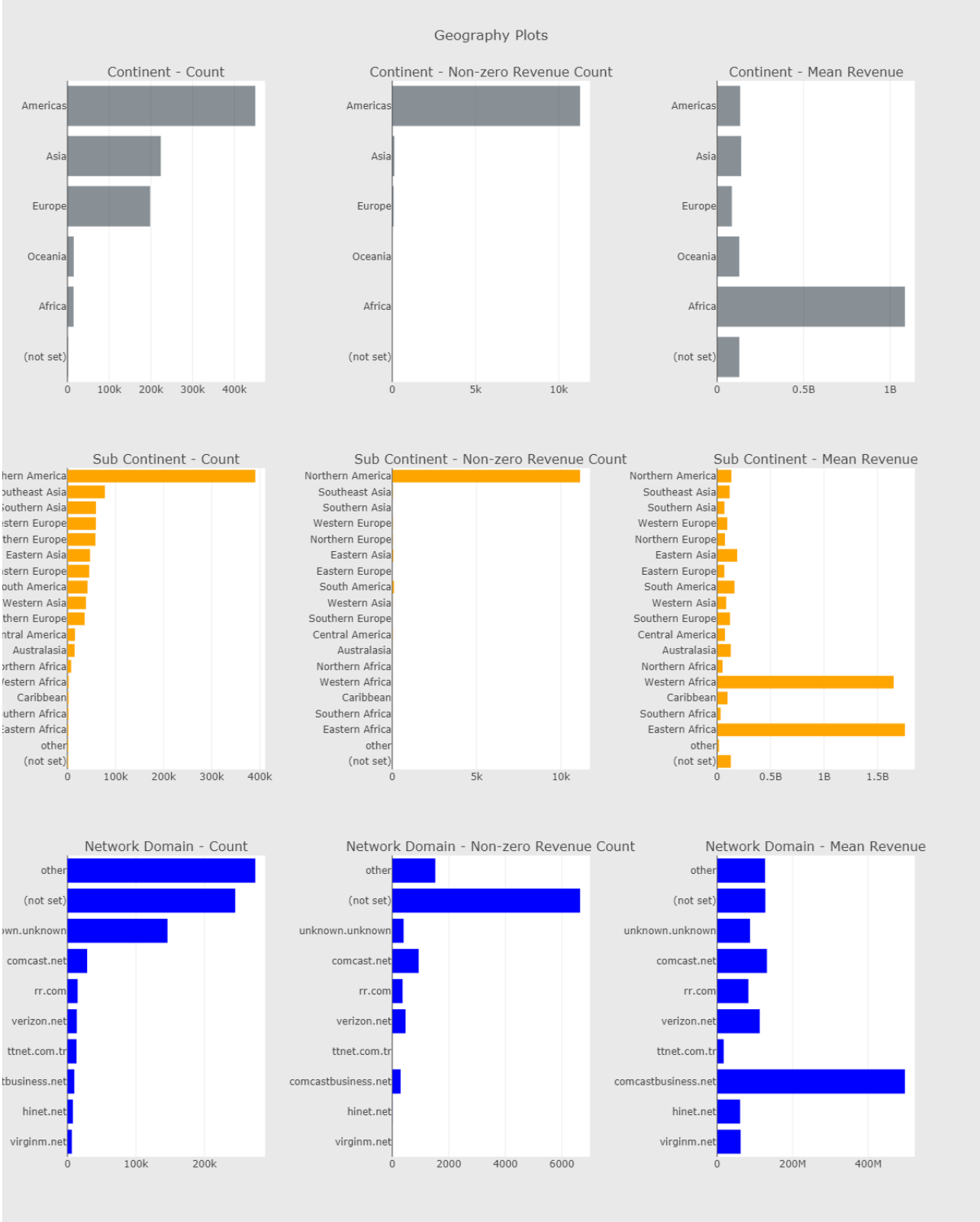

图 3.3 地理位置与浏览量和购买量的情况

§ 3.2.4 Traffic

从网络的流入来源可以发现，从谷歌搜索直接到谷歌商城的用户数是最多的，其他就是youtube，意味着谷歌商城在youtube上面的广告收到了一定效果。但是从转换到购买量的情况来看，始终是从谷歌搜索谷歌商城以及直接输入谷歌商城地址的用户能够带来实际收益，后者能够直接跳转到谷歌商场，说明其对谷歌商场已经具有相当高的熟悉度了。这一点也从平均消费金额上可以看出，直接跳转的用户达到了第二高的水平。而第一高平均消费金额的用户则是从dfa跳转而来的，但是由于笔者未能理清dfa的真实含义，因此此处无法进一步分析。另外从平均消费金额上可以看出，facebook也为谷歌商城的购买量提供了有力的支持。

从Traffic Source Medium中可以看出，直接找到谷歌商城的用户(organic)是占据最大的浏览量以及第二大的购买量。但是从平均购买金额来看则是cpm的用户最高。

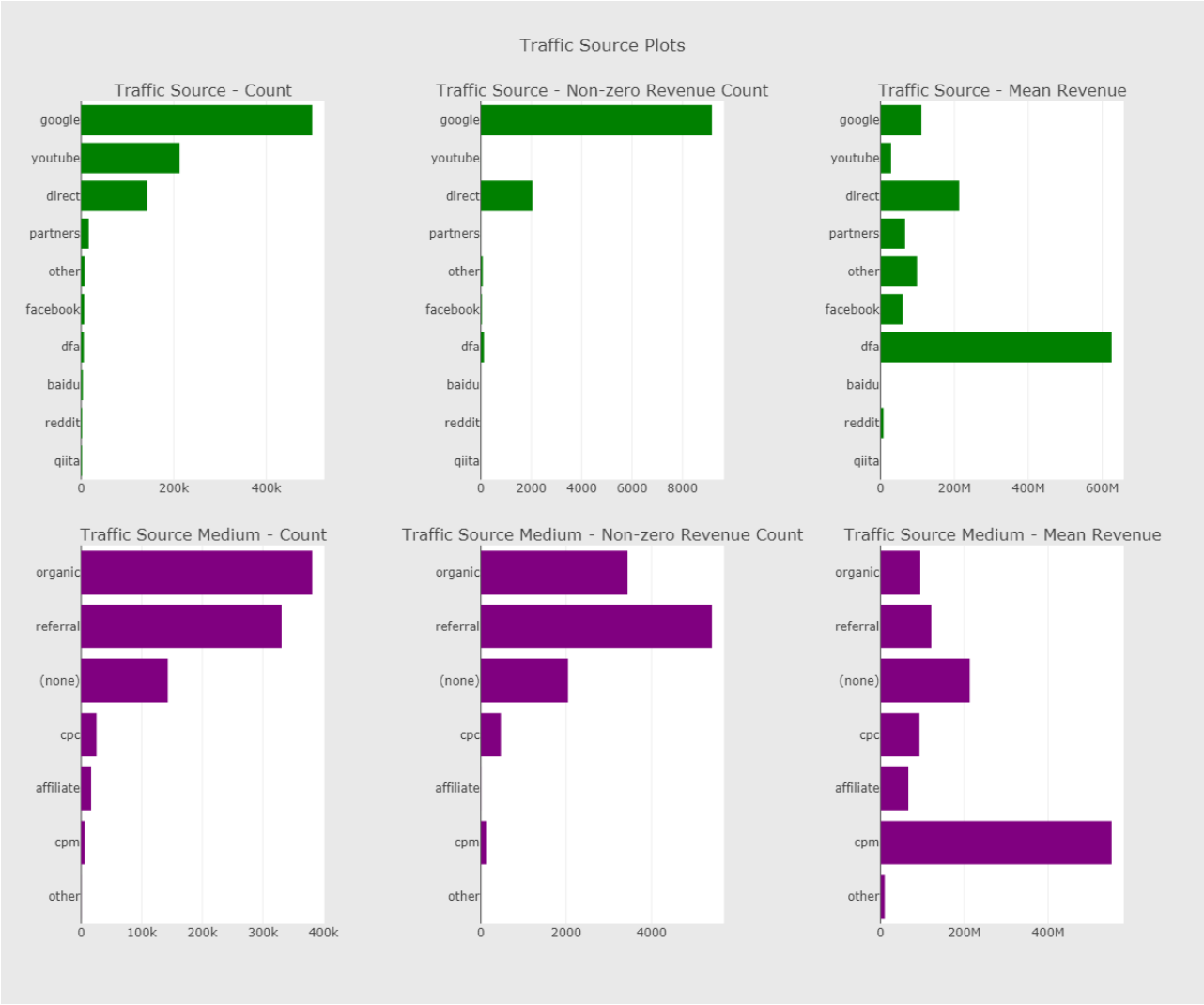

图 3.4 网络与浏览量和购买量的情况

§ 3.2.5 Vistor

除了之前的分析角度，直接从顾客当次访问的行为进行分析也是一种十分重要的分析角度。直观来想，当一名顾客在一个页面逗留的时间越长，他对于该商品的兴趣也一般会越高，购买行为也更加容易发生。同时一次访问浏览多个页面的用户也意味着具有更高的消费倾向。因此笔者统计了不同浏览页面数下的人数、有价值顾客数以及平均购买需求。从图中可以看出，浏览页面数低于10的和点击数少于10的用户占据绝大多数，但是从有价值的用户来看，反而是超过10的用户贡献了非零消费量，这也印证了商业当中著名的帕累托法则——20%的用户贡献了80% 的销售量。同时可以看出，有价值的用户也集中在浏览20个页面附近，并且呈现长尾效应。而消费均值图中可以看出，浏览页面数越多的用户平均消费金额也一般越大，这也符合我们常识。Pageviews和Hits这两个描述顾客浏览行为的指标对于判断用户消费量具有重要作用，之后从模型中笔者会进一步深入分析。

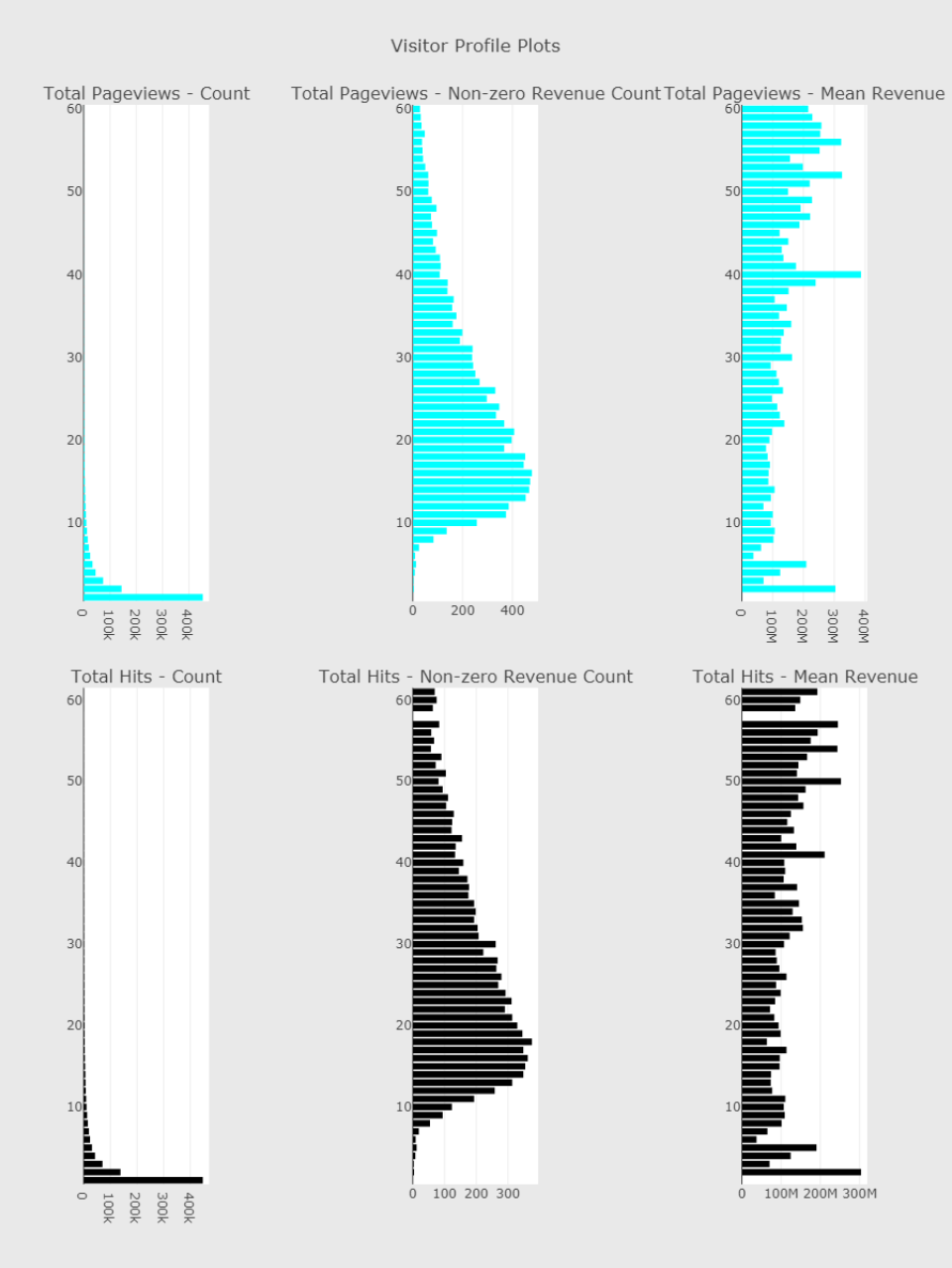

图 3.5 行为特征与浏览量和购买量的情况

§3.2.6 异常情况

除了对于数据集特征单独分析之外，特征之间存在着重复的含义也值得我们研究，特别是异常的情况，有时候能够为预测精度的提升带来帮助。从之前的数据分析中我们可以看出，visitStartTime和visitID应该是一致的，因为visitID就是以访问的开始时间来作为唯一编号。但是从数据中我们发现，训练集中存在1711条记录是两个不匹配的，可见图3.6。

另外训练集中一共有715119位用户，而测试集中一共有617242位用户，两者相同的用户

为7679。而数据集中也出现了相同记录的情况，一共有1796条。都需要进行继续清洗，笔者直接删除重复的记录情况以确保模型的稳定性。

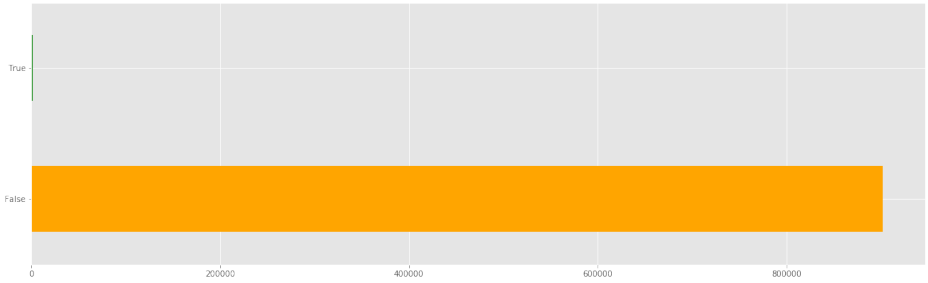

图 3.6 ID异常情况

§ 3.3 数据清洗

§ 3.3.1 常数列剔除

首先我们需要去除所有常数列——即该字段在训练集上取值（包括缺失也认为是一种值）只有一种。常数列无法提供任何有助于提升预测能力的信息。经过统计后发现有不少常数列，详细字段参考表3:

字段名	
socialEngagementType	Not Socially Engaged
device.browserSize	not available in demo dataset
device.browserVersion	not available in demo dataset
device.flashVersion	'not available in demo dataset
device.language	not available in demo dataset
device.mobileDeviceBranding	not available in demo dataset
device.mobileDeviceInfo	not available in demo dataset
device.mobileDeviceMarketingName	not available in demo dataset
device.mobileDeviceModel	not available in demo dataset
device.mobileInputSelector	not available in demo dataset
device.operatingSystemVersion	not available in demo dataset
device.screenColors	not available in demo dataset
device.screenResolution	not available in demo dataset
geoNetwork.cityId	not available in demo dataset
geoNetwork.latitude	not available in demo dataset
geoNetwork.longitude	not available in demo dataset
geoNetwork.networkLocation	not available in demo dataset
totals.visits	1
trafficSource.adwordsClickInfo.criteriaParameters	not available in demo dataset

表 3.3 常数字段名

同时我们发现，训练集中的totals.transactionRevenue和trafficSource.campaignCode这两个字段在测试集中没有。

§ 3.3.2 异常值处理

§ 3.3.2.1 目标特征异常值处理

在分析各特征对于目标变量的影响之前，我们有必要对目标变量的分布进行探究。3.7

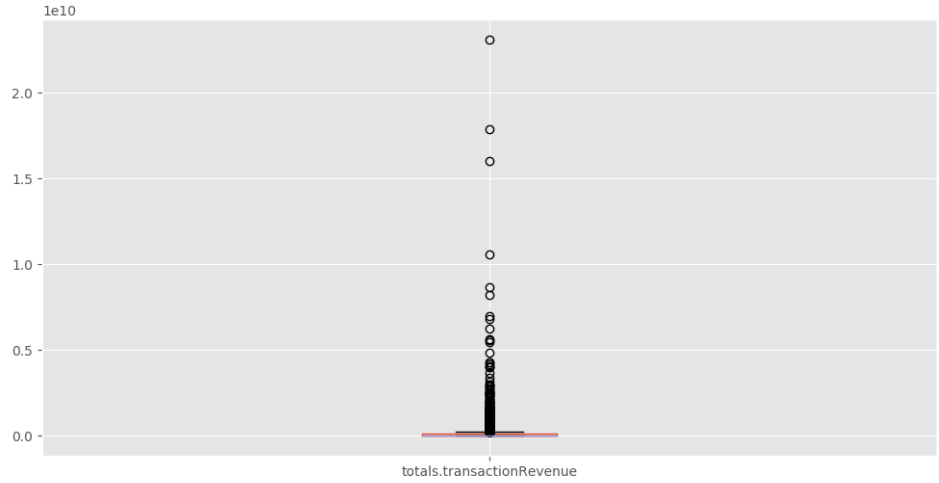

图 3.7 目标变量的分布

从箱线图中可以看出，目标变量存在异常值情况，即单次访问google store的购买量竟然出现超过10^10美元的消费量，显然是不合理的。为了去除异常值的影响，我们采用数据分析方法中常用的盖帽法：即将超过99分位数*4的所有数人为设置为99分位数*4。

去除异常值之后的数据我们画出类似基尼系数图的图像3.8，尽管商业中存在80/20法则，即20%的用户贡献了80% 的收入，但是从数据中我们发现实际上真实数据低于20%。

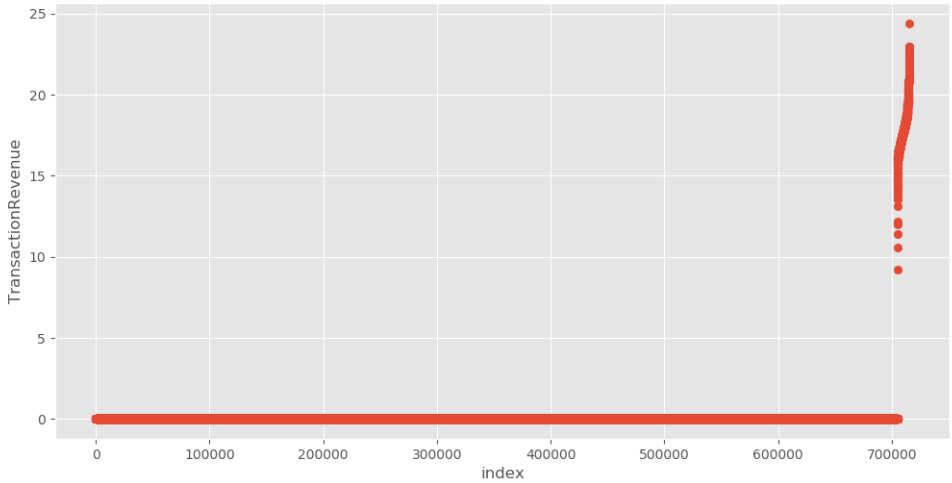

图 3.8 目标特征分布

而经过统计分析我们可以得知，非0行仅占12.7%，而贡献为非0的用户也仅占14.0%。因

此数据当中存在着严重的不均衡情况。需要进一步处理。

§3.3.3 缺失值探索

数据中的缺失值包括两种。一种是本身即是na值，能够自动被程序识别为缺失情况，一般存在于数值型特征中。另外一种即在初步探索中已经初步介绍的'not available in the demo dataset'，一般存在类型特征中。下面我们对数据中的缺失情况进行探索。首先对显式缺失值情况进行探索——将所有存在缺失的字段以及它在训练集和测试集中的缺失率，情况如表3.4

columns	train_missing_ratio	test_missing_ratio
trafficSource.campaignCode	0.999999	NaN
trafficSource.adContent	0.987887	0.933153
totals.transactionRevenue	0.987257	NaN
trafficSource.adwordsClickInfo.adNetworkType	0.976252	0.933124
trafficSource.adwordsClickInfo.isVideoAd	0.976252	0.933124
trafficSource.adwordsClickInfo.page	0.976252	0.933124
trafficSource.adwordsClickInfo.slot	0.976252	0.933124
trafficSource.adwordsClickInfo.gclId	0.976140	0.933064
trafficSource.isTrueDirect	0.696781	0.676254
trafficSource.referralPath	0.633774	0.707558
trafficSource.keyword	0.556551	0.485945
totals.bounces	0.501324	0.476878
totals.newVisits	0.221980	0.248935
totals.pageviews	0.000111	0.000173

表 3.4 缺失值字段名

在测试集中不存在的两个特征在训练集中也严重缺失，因此直接舍弃这些特征。而trafficSource.adContent特征缺失率在训练集中高达98.7%。但是缺失程度高的特征不代表不能带来信息，甚至很有可能出现该特征的确实与否就与最终标记存在强关联关系。因此为了正确处理缺失值，我们对其余缺失特征进行了更加深入的探索，具体情况如表3.5

§3.3.4 填充数值变量

而针对数值型特征的缺失值填补，我们通过箱线图进一步查看了它们的情况。

字段名	非零样本数	取值数	众数	众数频率
trafficSource.adwordsClickInfo.adNetworkType	75274	3	Content	0.560924
trafficSource.adwordsClickInfo.isVideoAd	75274	1	False	1
trafficSource.adwordsClickInfo.slot	75274	3	RHS	0.567925
trafficSource.adwordsClickInfo.gclId	75423	59008	CN_Wh……Q	0.000981
trafficSource.isTrueDirect	534518	1	True	1
trafficSource.referralPath	566264	3196	/	0.24422
trafficSource.keyword	814376	5392	(not provided)	0.87215

表 3.5 缺失值字段情况（类别特征）

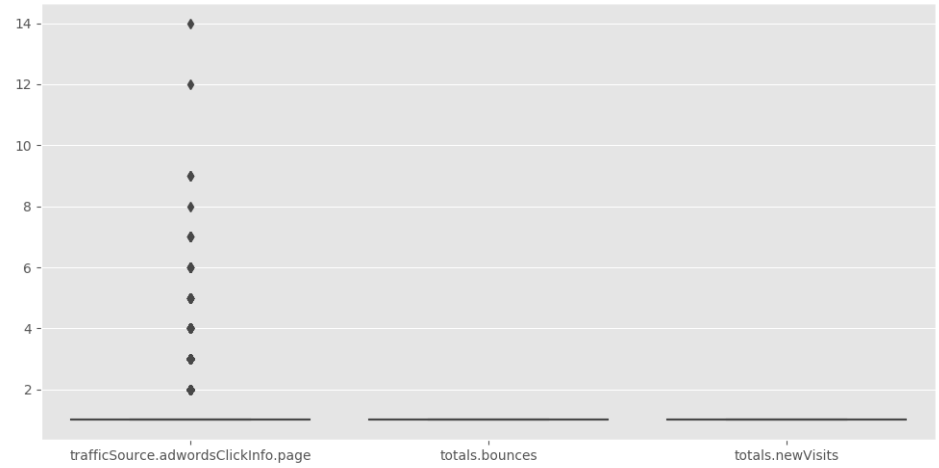

图 3.9 数值型缺失变量箱线图

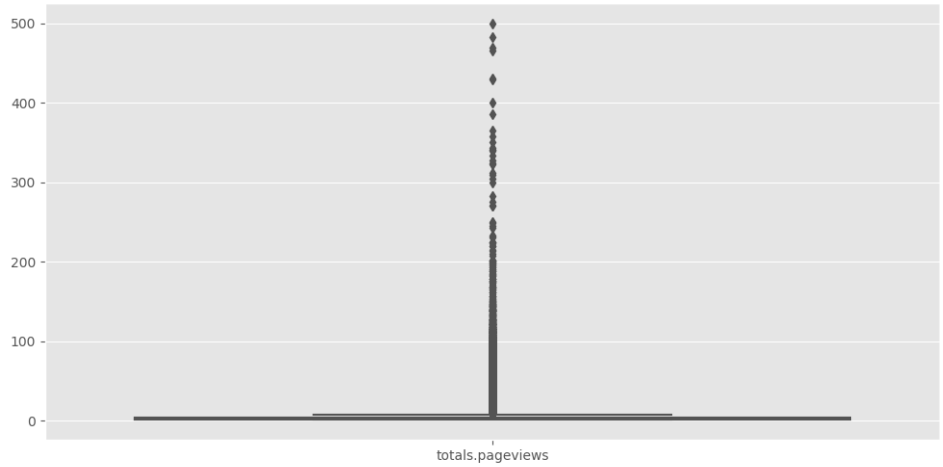

图 3.10 数值型缺失变量箱线图2

由图以及之前的字段说明可以看出，totals.bounces和totals.newVisits都是只有两种取值，缺失值只是为了存储的方便而并非为信息缺失，可以直接填补为0。而pageviews有接近千分之一的值为缺失值，但是考虑到正常情况pageviews不可能少于1（即用户进入google store不可能不经过任何一个页面），因此我们此处将缺失值设填补为1。trafficSouece.adwordsClickInfo.page缺失率高达97%，为了进一步探究缺失值数据是否有价值以及如何填补，我们统计了在该特征不同取值（包括缺失值）下购买的频率，结果如表3.6

取值	均值	方差
NaN	0.012538	0.111269
1.0	0.021253	0.144229
2.0	0.000000	0.000000
3.0	0.000000	0.000000
4.0	0.000000	0.000000
5.0	0.000000	0.000000
7.0	0.000000	0.000000
9.0	0.000000	0.000000
14.0	0.000000	NaN

表 3.6 缺失值字段情况（类别特征）

进一步利用箱线图描述数据分布如图3.11

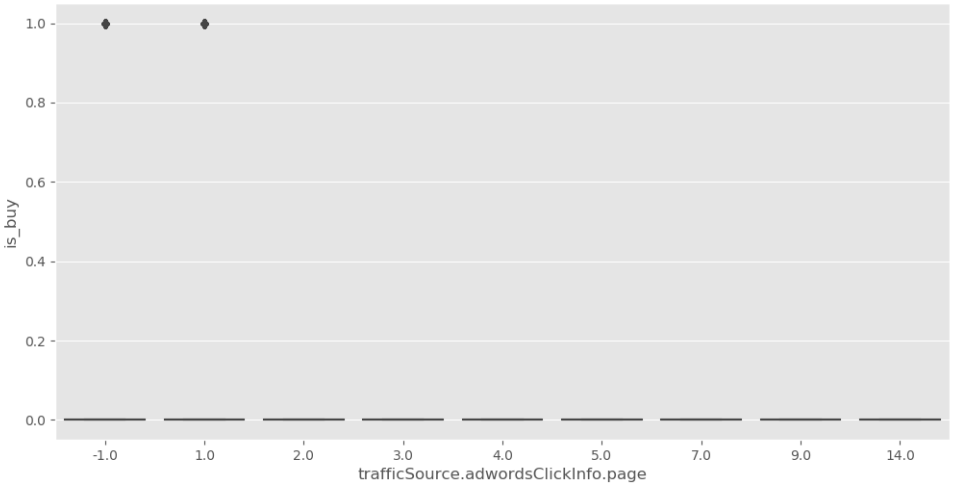

图 3.11 数值型缺失变量箱线图

可以看出缺失值当中存在着大量的信息，具体表现为方差为0.11¿0，因此不能舍弃该特征。而该特征上取值大于1的样本反而表示用户不会购买，因此可以对数据进行如下规约：对缺失值填充为-1，对大于1的取值填充为2，而取值为1的不做改变。

综上所述，对数值型特征的缺失值进行如下的填充策略：

1. 舍弃trafficSource.campaignCode和totals.transactionRevenue
2. totals.bounces和totals.newVisits的缺失值填充为0
3. totals.pageviews的缺失值填充为1
4. trafficSouece.adwordsClickInfo.page的缺失值填充为-1，取值大于1的统一重新取值为2。

§3.3.5 填充类型变量

首先是填充trafficSource.adwordsClickInfo.adNetworkType特征。缺失率虽然高达97%，但是从箱线图中可以注意到非缺失值中有重要信息：凡是取值为search partnet的用户都不会进行购买，因此该特征不能去除。注意到测试集中一共有四种取值，分别是Nan,'Google Search', 'Search partners' 和'Content'，但训练集中却没有'Content' 取值！而且'Content' 在测试集中还占据了绝大部分，因此将'Content'归为缺失值分析。同时将所有缺失值重新赋值为' not available in demo dataset'以便于其他特征统一。

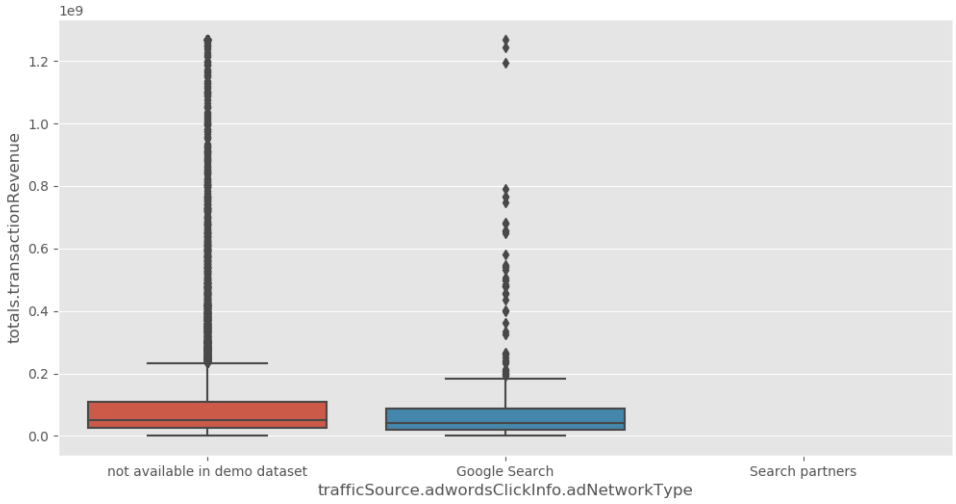

图 3.12 adNetworkType箱线图

而trafficSource.adwordsClickInfo.isVideoAd特征简单地将缺失值填补为True即可。

而trafficSource.adwordsClickInfo.slot特征中的revenue分布如下图3.13,因此直接将缺失值填补为'not available in demo dataset'。

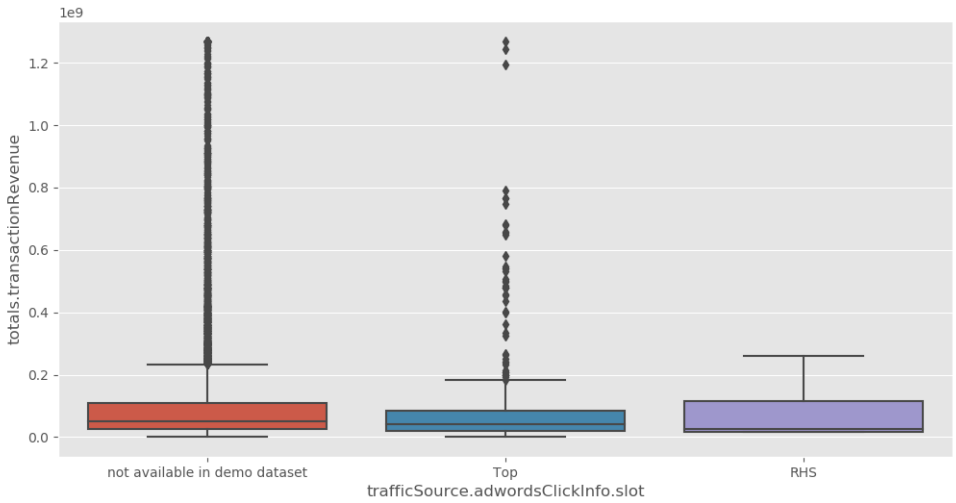

图 3.13 adwordsClickInfo_slot箱线图

但是对字段trafficSource.adwordsClickInfo.gclId、trafficSource.referral Path和trafficSource.keyword，由于取值数量非常多，难以直接看出填补缺失值规律，因此直接将缺失值填补为'not available in demo dataset'。字段trafficSource.isTrueDirect中的缺失值填

补为False。

因此最终填补缺失值的处理总结如下：

1. trafficSource.adwordsClickInfo.adNetworkType中对缺失值填补为not available in demo dataset，同时将Content也更换为not available in demo dataset。
2. trafficSource.adwordsClickInfo.isVideoAd中缺失值填补为True。
3. trafficSource.isTrueDirect 填补为False。
4. trafficSource.adwordsClickInfo.slot、trafficSource.adwordsClickInfo.gclId、trafficSource.referralPath和trafficSource.keyword中的缺失值都直接填补为'not available in demo dataset'。

§3.4 数据规约

数据集约一般包括两种：一种是取值上的集约，即将大量样本量很小的取值归为一类，以减少运算并提高每一类当中的信息含量；第二种是特征数量上的集约，即找到最小特征子集，使得其能最大程度上保留原数据中的信息，便于建模分析。

此处我们先进行第一种集约，第二种集约留作特征工程中完成。经过数据填补之后，接下来我们要去除重复值的情况。所谓重复值，是指内容实质上是一样的，但是由于处理、存储等过程中不注意加入了多余符号或者表达形式不统一，导致最终数据产生了多个版本。比如在trafficSource.adContent字段中既有出现'KeyWord:Google Merchandise' 也有出现'Google Merchandise'，从而导致后续模型在处理的时候误以为这是两种不同的取值，无法捕捉其中的信息导致出现偏差。另外如果需要进行one-hot编码操作，重复值会导致数据维度产生大量不必要的增加，造成模型难以捕捉特征之间的准确信息，对于模型准确率影响很大。常见的去除重复值的操作，一般是先统一大小写，然后之后只能根据数据取值进行人为制定规则判断。由于该部分工作量很大，需要经过对每一个字段的仔细分析之后才能得出规则，最终一共制定了接近30条规则。因此此处只展示需要清洗的字段以及部分规则，详见:

1. device.browser:凡是具有android、samsung、chrome、lunascape等名称在内的统一转换为mobile 等trafficSource.adContent:凡是具有google 在内的统一直接转换为google
2. trafficSource.source:凡是具有youtube的统一直接转换为youtube
3. device:将trafficSource.source与geoNetwork.country拼接在一起表示完整的国家层级的地址
4. campaign.medium:更新为原始的trafficSource.campaign与trafficSource.medium拼接在一起
5. browser.category:更新为device.browser跟device.deviceCategory拼接在一起的表示设备+版本的统一信息

6. browser.os:更新为device.browser跟device.operatingSystem拼接在一起的统一表示设备系统版本的信息

7. 其他规则

在之前的数据清洗中能够看出，有大量特征取值数较多，但是其中大部分取值所具有的样本数占比很小，为了增强模型的泛化能力，有必要对占比较小的取值进行数据规约，即对类别性特征而言将其取值重新设置为'others'。此处我们对测试集中样本数小于1000的取值进行规约。之所以选择测试集而不是训练集来对特征取值进行判断，是基于提升模型的泛化能力而非在训练集上的表现考虑的。最终进行规约的特征以及规约后的取值如表3.7：

特征	去除的取值数	最终在训练集上的取值数
device.browser	98	11
device.browser	15	8
geoNetwork.country	160	60
geoNetwork.city	656	77
geoNetwork.metro	86	24
geoNetwork.networkDomain	25689	62
geoNetwork.region	314	62
geoNetwork.subContinent	5	19
trafficSource.adContent	0	3
trafficSource.campaign	23	5
trafficSource.keyword	2409	8
trafficSource.medium	1	7
trafficSource.medium	2173	17
trafficSource.medium	36	13

表 3.7 数据规约（类别特征）

从表中我们可以看出，数据规约的效果十分明显，部分特征的取值数甚至减少了99.67%。这对于节省训练时间以及提高模型泛化能力都有极其重要的意义。

至此，绝大部分业务上重要的变量已经经过分析，但是由于变量数过多，其他含义不明的变量不再进行人工分析，而是在后续过程中由模型给出相应分析。至此我们得到了第二个数据集，即完成了数据填补和清洗后的数据。下面的特征工程以及训练都基于此数据集完成。

§3.5 基础模型

为了进一步探究特征之间的重要程度以及关系，需要使用模型进行初步训练和预测，通过模型获得新的洞察。在构建模型之前，笔者将训练集中的前600000个样本作为训练数据，后面超过300000个样本作为验证集，来观察不同模型的表现。目标变量为消费量加一后取自然对数，即$y_i = log(y_i + 1)$。最终使用RMSE来进行评价，即：

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{(y)}_i)^2}$$

§3.5.1 决策树回归模型

应用决策树分类的思想，通过对目标函数等作出相应的改变，即可实现回归功能。笔者使用sklearn下的DecisionTreeRegresssor模型，最大深度限制为4以确保模型的泛化能力，其他参数选择默认，即criterion='mse', max_depth=4, max_features=None, max_leaf_nodes=None, min_impurity_decrease=0.0, min_impurity_split=None, min_samples_leaf=1, min_samples_split=2, min_weight_fraction_leaf=0.0, presort=False, random_state=None, splitter='best'，最终在验证集上得到RMSE=1.75189的成绩。

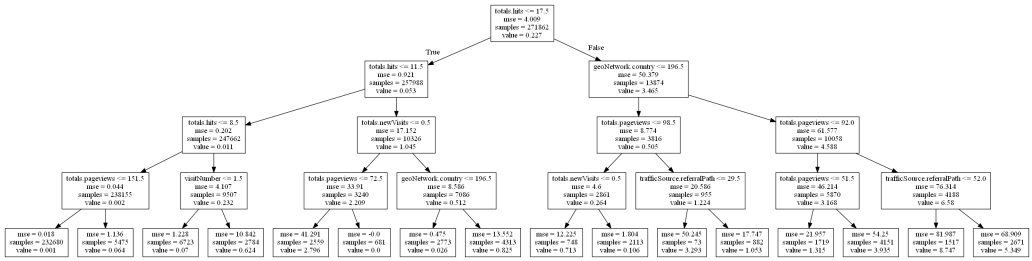

图 3.14 决策树模型

从决策树的第一层可以看出，点击数被决策树认为是最重要的特征，并且第一层根据点击数小于等于17.5或者大于等于17.5来进行区分。第二层同样继续考虑点击数，但是同时考虑了国家，决策树首先将国家编号小于等于196的国家划为一类，虽然直接对国家进行了整数编码而不是one-hot编码，但从决策树的算法可知，由于其非线性的划分方式，编码所引入的整数之间的相对大小关系对最终结果不会有明显影响。从第三层开始，决策树又一次划分了点击数，但是同时也考虑了是否为第一次到Google Store这个指示特征，并且考虑了该次访问的浏览页面数。另外从决策树的最左边即首先采用两次hits划分，再采用pageviews来划分的得到的mse仅为0.018，而采用hits、netVisits和pageviews进行三层划分后更是直接达到了mse=0的结果，说明了点击数hits、浏览页面数pageviews以及是否为第一次访问google store对于判断用户最终消费量都具有极其重要的影响。

§3.5.2 随机森林模型

鉴于决策树容易过拟合的缺点，随机森林采用多个决策树的投票机制来改善决策树，我

们假设随机森林使用了m棵决策树，那么就需要产生m个一定数量的样本集来训练每一棵树，如果用全样本去训练m棵决策树显然是不可取的，全样本训练忽视了局部样本的规律，对于模型的泛化能力是有害的产生n个样本的方法采用Bootstraping法，这是一种有放回的抽样方法，产生n个样本而最终结果采用Bagging的策略来获得，即多数投票机制. 在统计学中，Bootstrap 是依靠替换随机采样的任意试验或度量。我们从上文可以看见，决策树会受到高方差的困扰。这意味着如果我们把训练数据随机分成两部分，并且给二者都安置一个决策树，我们得到的结果可能就会相当不同。Bootstrap 聚集，或者叫做袋装，是减少统计学习方法的方差的通用过程。

给定一组n 个独立的样本观测值$Z_1, Z_2, ..., Z_n$，每一个值的方差均为σ^2，样本观测值的均值方差为σ^2/n。换句话说，对一组观测值取平均会减小方差。因此一种减小方差的自然方式，也就是增加统计学习方法预测精度的方式，就是从总体中取出很多训练集，使用每一个训练集创建一个分离的预测模型，并且对预测结果求取平均值。

这里有一个问题，即我们不能获取多个训练数据集。相反，我们可以通过从（单一）训练数据集提取重复样本进行自助法（bootstrap）操作。在这种方法中，我们生成了B 个不同的自助训练数据集。我们随后在第b 个自助训练数据集得到了一个预测结果，从而获得一个聚集预测。Bagging 方法最大的优势是我们可以不通过交叉验证而求得测试误差。回想一下，Bagging 方法的精髓是多棵树可以重复地拟合观察样本的自助子集。平均而言，每一个袋装树可以利用2/3 的观察样本。而剩下的1/3 观察样本就可以称为out-of-bag 观察样本(可以通过重要极限证明，准确数值应该为$\frac{1}{e}$)，它们并不会拟合一一棵给定袋装树。我们可以使用每一棵树的OOB 观察样本而计算第i 个观察样本的预测值，这将会导致大约有B/3 的预测值可以预测第i 个观察样本。现在我们可以使用和Bagging（平均回归和大多数投票分类）类似的聚集技术，我们能获得第i 个观察样本的单一预测值。我们可以用这种方式获得n 个观察样本的OOB 预测，因此总体的OOB MSE （回归问题）和分类误差率（分类问题）就能计算出来。OOB 误差结果是Bagging 模型测试误差的有效估计，因为每一个样本的预测值都是仅仅使用不会进行拟合训练模型的样本。随机森林的生成方法：

- 从样本集中通过重采样的方式产生n个样本
- 假设样本特征数目为a，对n个样本随机选择a中的k个特征，用建立决策树的方式获得最佳分割点
- 重复m次，产生m棵决策树
- 多数投票机制来进行预测

笔者首先采用sklearn当中的RandomForestRegressor，并且同样采用最大深度为4，其余参数默认的设置：bootstrap=True, criterion='mse', max_depth=4, max_features='auto', max_leaf_nodes=None, min_impurity_decrease=0.0, min_impurity_split=None, min_samples_leaf=1, min_samples_split=2, min_weight_fraction_leaf=0.0, n_estimators=10, n_jobs=1, oob_score=False, random_state

=None, verbose=0, warm_start=False。最终在验证集上得到的RMSE=1.740924，简单的设置就能优于之前设置的回归树模型。

由于随机森林涉及到的超参数数量比回归树多，因此笔者采用网格搜索法的方式来提升模型效果。sklearn当中grid search中有GridSearchCV类，可以很方便地实现网格搜索。网格搜索顾名思义就是通过等距或者按照log等距生成超参数样本，在每一个新的超参数下验证模型的性能，选择在验证集上效果最好的模型对应的超参数为最优超参数。为了确保验证效果准确，笔者采用了5折交叉验证，也就是将训练集又重新随机等分为5分，每一次取其中的四份进行训练，剩下的一份作为验证，最后所有数据都能被用于验证，以验证集上的损失的平均数作为模型的最终性能评分。

笔者首先尝试搜索随机森林的树数量即n estimators，搜索空间为10,20,30,40,50,60,70。最终确定10棵子树为最优超参数，也就是默认参数。由于树的最大深度增大时一方面能够提高模型在训练集上的拟合能力，但是另一方面会增大它过拟合的风险，因此max depth也是一个极其重要的超参数。同理，最小分割样本数越大，树往下分割的深度就会越低，越容易提早结束节点的分割，因此能够确保其泛化能力。笔者第二部就搜索这两个超参数，具体搜索空间为max depth=3,5,7,9,11,13和min samples split=50,70,90,110,130,150,170,190。经过35分钟的训练，最终得到最优深度为13，而最优min samples split为70。再经过其他一些超参数的搜索后，得到最佳超参数设置为：max depth=13,min samples split=70,min samples leaf=10，最终RMSE=1.67041，明显优于之前的默认参数下的随机森林。

§3.5.3 LightGBM

为了进一步明确变量之间的重要性和相关情况，需要借助有变量筛选能力的模型进行分析，此处笔者选择lightgbm框架进行分析。LightGBM是一个基于GBDT模型的框架。GBDT (Gradient Boosting Decision Tree)是机器学习中一个长盛不衰的模型，其主要思想是利用弱分类器（决策树）迭代训练以得到最优模型，该模型具有训练效果好、不易过拟合等优点。GBDT在工业界应用广泛，通常被用于点击率预测，搜索排序等任务。GBDT也是各种数据挖掘竞赛的致命武器，据统计Kaggle上的比赛有一半以上的冠军方案都是基于GBDT。因此此处笔者先介绍GBDT模型，再介绍LightGBM 框架的新特性。

梯度提升是一种用于回归、分类和排序任务的机器学习技术1，属于Boosting算法族的一部分。Boosting是一族可将弱学习器提升为强学习器的算法，属于集成学习的范畴。Boosting方法基于这样一种思想：对于一个复杂任务来说，将多个专家的判断进行适当的综合所得出的判断，要比其中任何一个专家单独的判断要好。通俗地说，就是“三个臭皮匠顶个诸葛亮”的道理。梯度提升同其他boosting方法一样，通过集成多个弱学习器，通常是决策树，来构建最终的预测模型。boosting方法通过分步迭代的方式来构建模型，在迭代的每一步构建的弱学习器都是为了弥补已有模型的不足。Boosting族算法的著名代表是AdaBoost，AdaBoost算法通过给已有模型预测错误的样本更高的权重，使得先前的学习器做错的训练样本在后续受到更多的关注的方式来弥补已有模型的不足。与AdaBoost算法不同，梯度提升方法在迭代的每一步构建一个能够沿着梯度最陡的方向降低损失的学习器来弥补已有模型的不足。

经典的AdaBoost 算法只能处理采用指数损失函数的二分类学习任务，而梯度提升方法通过设置不同的可微损失函数可以处理各类学习任务（多分类、回归、Ranking等），应用范围大大扩展。另一方面，AdaBoost算法对异常点（outlier）比较敏感，而梯度提升算法通过引入bagging思想、加入正则项等方法能够有效地抵御训练数据中的噪音，具有更好的健壮性。这也是为什么梯度提升算法（尤其是采用决策树作为弱学习器的GBDT算法）如此流行的原因。

对于一般的机器学习算法，都是构造一个从样本集到目标变量的映射$F : X \to Y$使得一定意义下的损失最少，以平方损失函数为例，也就是$\frac{1}{2}\sum_{i=0} n(y_i - F(x_i)^2) + \Omega$最小，其中$\Omega$为正则项，即约束模型的复杂性以确保其泛化能力。但是对于boosting 算法来讲，思路则会些变化：即通过构造一系列的模型$\{F_i(x)\}$，迭代改进来降低损失。也就是每一次生成一个新的模型来拟合前面模型所剩下的残差，因此对于第k 个模型，拟合的样本集为$(x_1, y_1 - \sum_{i=0}^{k-1} F(x_1)), (x_1, y_1 - \sum_{i=0}^{k-1} F(x_1)), \cdots$。因此GBDT模型可以认为是k个基模型组成的一个加法模型：

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i), f_k \in F \tag{3.1}$$

其中F为基模型空间，例如如果以树为基础模型，则F为全体树模型所构成的空间。对于boosting算法来讲，它对于(2)的求解采用的是前向优化算法，即从前往后逐步构造新的模型来比较目标函数：

$$\begin{aligned} y_i^0 &= 0 \\ y_i^1 &= y_i^0 + f_1(x_i) \\ y_i^2 &= y_i^1 + f_2(x_i) \\ &\ddots \\ y_i^k &= y_i^{k-1} + f_k(x_i) \end{aligned}$$

对于每一次寻找新模型逼近的过程，实际上就是最小化目标函数的过程，即：

$$argmin_{f_k} \qquad L(f_k) = \sum_{i=1}^{n} l(y_i, \hat{y}_i^{k-1} + f_t(x_i)) + \Omega(f_k) + C$$

对于$l(x, y) = \frac{1}{2}(x - y)^2$的平方误差函数而言，采用泰勒展开可以进一步将目标函数写成：

$$\begin{aligned} argmin_{f_k} \qquad L(f_k) &= \sum_{i=1}^{n} l(y_i, \hat{y}_i^{k-1}) + g_i f_k(x_i) + \frac{1}{2} h_i f_k^2(x_i) + \Omega(f_k) + C \\ &= \sum_{i=1}^{n} g_i f_k(x_i) + \frac{1}{2} h_i f_k^2(x_i) + \Omega(f_k) + C' \end{aligned}$$

其中g_i是损失函数的一阶导，h_i是损失函数的二阶导。

一般的基分类器为树模型，则模型可以进一步表示为$w_{q(x)}$其中$q : X \to T$为将样本集映射到叶子集上的映射，而$w : T \to R$为将叶子集映射到实数域上的映射。决策树的复杂度有显式的正则项表达式：

$$\Omega(w_q) = \gamma T + \frac{1}{2}\lambda \sum_{j=1} T w_j^2$$

，其中λ为用户自定义的惩罚因子，表示对正则项的惩罚力度大小。则最终目标函数可以写成：

$$argmin_{f_k} \qquad L(f_k) = \sum_{j=1} T[(\sum_{i \in I_j} g_i)w_j + \frac{1}{2}(\sum_{i \in I_j} h_i + \lambda)w_j^2] + \gamma T$$

值得一提的是，如果树模型结构已经确定，也就是q已经给定，那么求解w可以直接令一阶导为0求得，则叶子节点j 对应的值应该为：

$$w_j^* = -\frac{\sum_{i \in I_j} g_i}{\sum_{i \in I_j} h_i + \lambda}$$

所以接下来的工作就是枚举所有可能的树的结构，一般用贪心策略来实现：a、从深度为0的树开始，对每个叶节点枚举所有的可用特征

b、针对每个特征，把属于该节点的训练样本根据该特征值升序排列，通过线性扫描的方式来决定该特征的最佳分裂点，并记录该特征的最大收益（采用最佳分裂点时的收益）

c、选择收益最大的特征作为分裂特征，用该特征的最佳分裂点作为分裂位置，把该节点生长出左右两个新的叶节点，并为每个新节点关联对应的样本集

d、回到第1步，递归执行到满足特定条件为止

目前已有的GBDT工具基本都是基于预排序的方法（pre-sorted）的决策树算法(如xgboost)。这种构建决策树的算法基本思想是：首先，对所有特征都按照特征的数值进行预排序。其次，在遍历分割点的时候用O(data)的代价找到一个特征上的最好分割点。最后，找到一个特征的分割点后，将数据分裂成左右子节点。这样的预排序算法的优点是能精确地找到分割点。缺点也很明显：首先，空间消耗大。这样的算法需要保存数据的特征值，还保存了特征排序的结果（例如排序后的索引，为了后续快速的计算分割点），这里需要消耗训练数据两倍的内存。其次，时间上也有较大的开销，在遍历每一个分割点的时候，都需要进行分裂增益的计算，消耗的代价大。

而LightGBM基于Histogram的决策树算法，即先把连续的浮点特征值离散化成k个整数，同时构造一个宽度为k的直方图。在遍历数据的时候，根据离散化后的值作为索引在直方图中累积统计量，当遍历一次数据后，直方图累积了需要的统计量，然后根据直方图的离散值，遍历寻找最优的分割点。并且使用带深度限制的Leaf-wise的叶子生长策略，每次从当前所有叶子中，找到分裂增益最大的一个叶子，然后分裂，如此循环。因此同Level-wise相比，在分裂次数相同的情况下，Leaf-wise可以降低更多的误差，得到更好的精度。Leaf-wise的缺点是可能会长出比较深的决策树，产生过拟合。因此LightGBM在Leaf-wise之上增加了一个最大深度的限制，在保证高效率的同时防止过拟合。也就是深度搜索与广度搜索的区别了。

另一个优化是Histogram（直方图）做差加速。一个容易观察到的现象：一个叶子的直方图可以由它的父亲节点的直方图与它兄弟的直方图做差得到。通常构造直方图，需要遍历该叶子上的所有数据，但直方图做差仅需遍历直方图的k个桶。利用这个方法，LightGBM可以在构造一个叶子的直方图后，可以用非常微小的代价得到它兄弟叶子的直方图，在速度上可以提升一倍。并采用多线程优化、支持类别特征等一系列的优化，因此一发布就一跃成为了目前最广受欢迎的数据挖掘算法之一，被广泛应用于Kaggle等数据挖掘比赛中。

根据之前的网格搜索调参，笔者选择的LightGBM参数如下表：

超参数	取值
metric	rmse
num leaves	30
min child samples	100
learning rate	0.1
bagging fraction	0.7
feature fraction	0.5
bagging frequency	5
bagging seeding	2018
verbosity	-1

图 3.15 LightGBM超参数选择

LightGBM经过400次迭代就收敛，并且选择了在验证集上表现最好的第342次迭代模型作为最终模型，最终RMSE=1.64259，优于调参后的随机森林，并且训练速度非常快。另外基于树的模型都能给出特征重要性指标。具体而言一棵树中的特征的排序(比如深度)可以用来作为特征相对重要性的一个评估，居于树顶端的特征相对而言对于最终样本的划分贡献最大(经过该特征划分所涉及的样本比重最大)，这样可以通过对比各个特征所划分的样本比重的一个期望值来评估特征的相对重要性。因此直接使用LightGBM输出特征重要性图3.16。

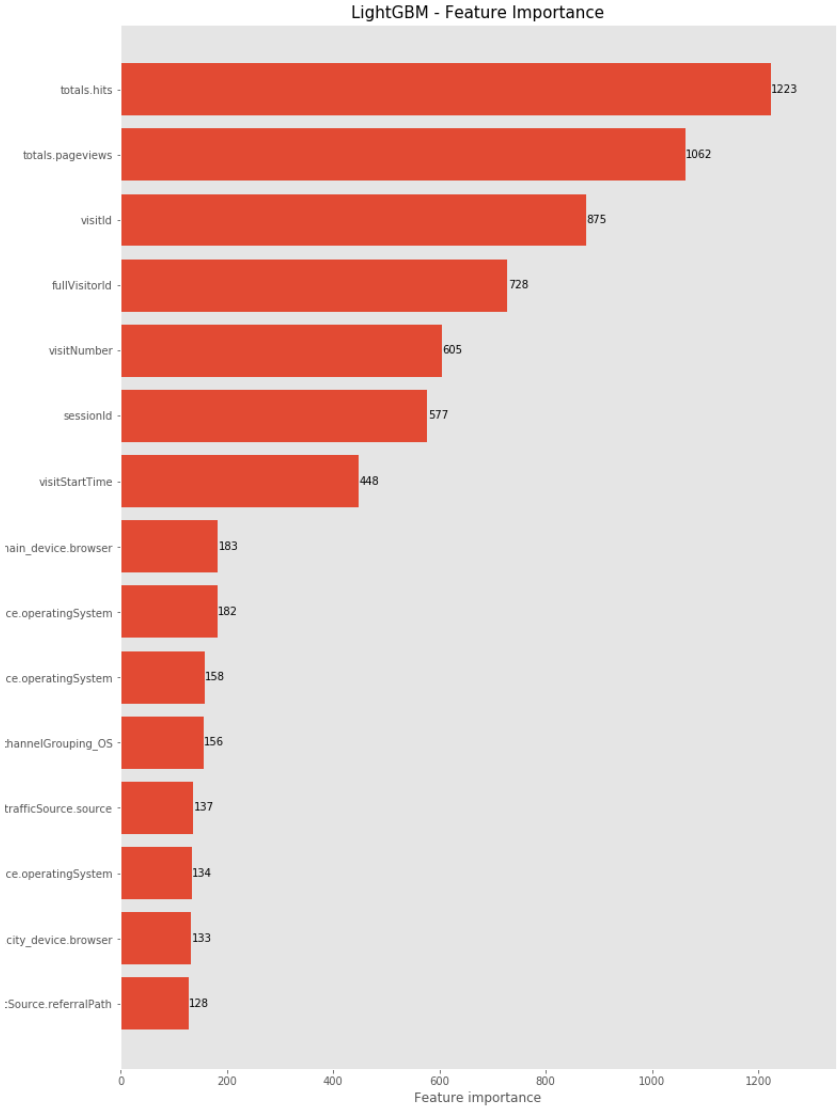

图 3.16 特征重要性

从LightGBM给出的特征重要性图中我们可以看出，有关用户访问的行为特征：hits、pageviews和visitNumber都是极其重要的，这也印证了我们之前的数据探索中的发现。其次visitStartTime、浏览器种类、操作系统、referralPath 这些在我们之前的数据探索中分析过的特征也都进入了特征重要性前20名的位置。因此之后的特征工程可以考虑针对这些特征进行重构，

提出新的更加有效的特征。

§ 3.5.4 神经网络

神经网络作为当前最热门的人工智能方向之一，其通过多层映射的结构获得的良好的拟合能力，使得它在语音识别、图像识别等方面取得了突破性的进展，因此此处笔者也尝试使用神经网络来预测用户本次的消费量。

在进行神经网络预测之前，需要对类别变量进行one-hot编码。考虑到数据的维度之大以及关系之复杂，笔者建立了一个四层的神经网络，每一层神经元数目分别是512,256,128,64,1，同时除了输出层之外每一层都使用relu 函数作为激活函数，每一层在relu函数之前都加入batch normalization以确保不会出现梯度消失的现象，同时保证学习速率，在relu函数之后都加入一层dropout 随机灭活一些神经元避免神经网络过拟合。

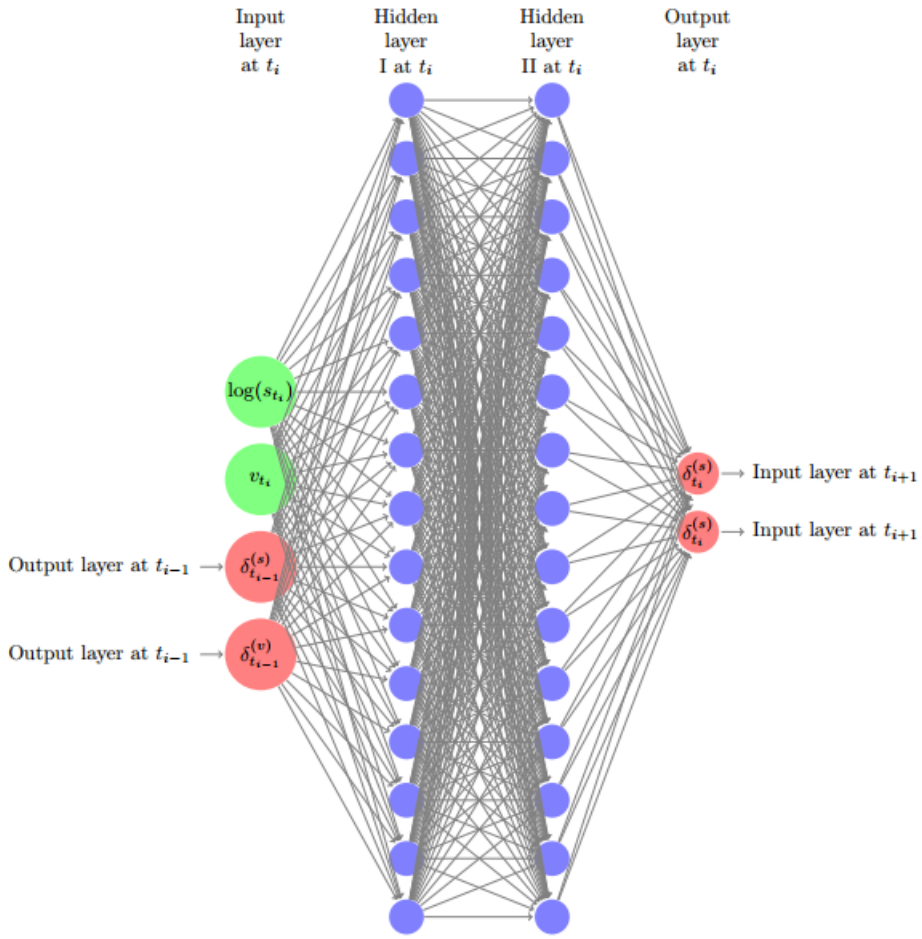

图 3.17 神经网络结构示意图（非本文结构）

在运行了8个小时之后，神经网络完成了迭代200次，训练过程如图3.18。

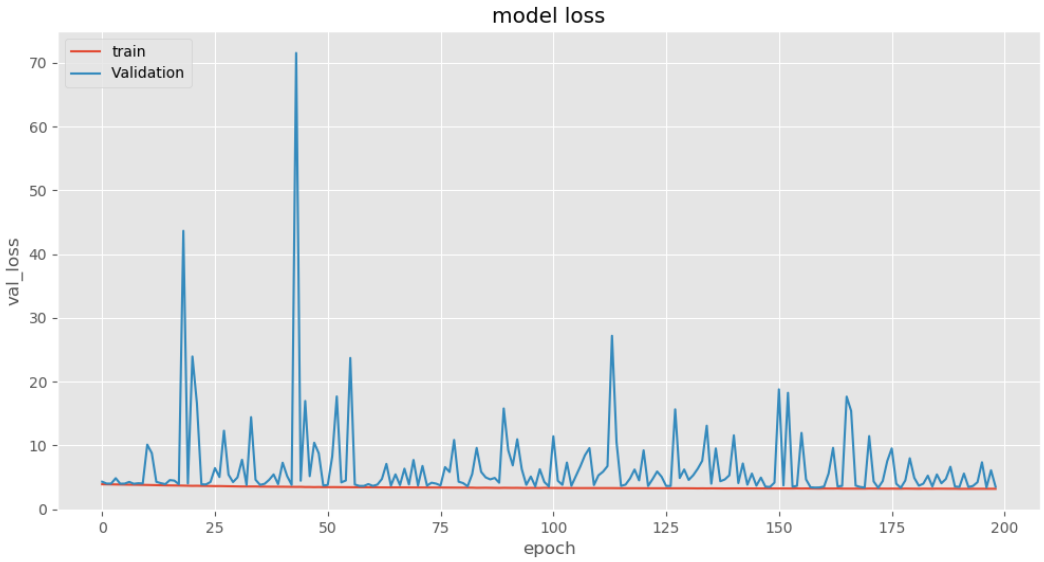

图 3.18 神经网络训练过程

可以看出随着训练的次数增多，模型在验证集上的损失也逐渐降低，并未出现明显的过拟合情况。最终在验证集上的rmse=1.8794，远高于之前的所有模型，因此判断在算力有限的情况下，神经网络不适合该数据的预测任务。

§3.6 特征工程

在业界有一句话广泛流传：数据和特征决定了机器学习的上限，而模型和算法只是逼近这个上限而已。也就是rubbish in, rubbish out。一般的模型并不能自动从原始数据中构造有价值的特征，神经网络虽然具有自动构造特征特别是高阶交互项的能力，但是需要大量的训练时间和恰当的网络结构设计，而且其效果一般难以保证具有鲁棒性。因此普遍特征都需要数据挖掘工程师根据对于问题的理解和分析，结合常用的技术手段来完成构建。构建特征工程的方法有非常多，包括生成交互项——多项式项等，并且针对不同特征具有相应的方法论。在本问题中特征同样具有多种类别，因此需要分别针对其特性进行相应的特征构造，从原始特征中构造出具有业务逻辑的特征，便于模型从中得到提升。

§3.6.1 时序特征

数据中具有访问日期该时间类别特征，考虑到购物行为与时间具有高度相关性，即周末空余时间较多，因此购物的可能性更大，而平时工作日大部分人需要上班，无暇购物。数据中的时间特征形式为字符串的“%Y%M%D%H%M%S”，为了充分获取其中信息，考虑将年、月、日、时、分、秒分别进行拆分为多个特征，同时构建二值特征——是否为工作日，直接得到与购物活跃度相关的指标。值得注意的是，笔者在实验中发现并不是将特征构建得

越多效果会越好，相反，过多的时间特征会导致尽管在训练集甚至在验证集上都取得较好成绩，但是在Kaggle网站上的Leadboard排行榜上效果不好，笔者认为是由于出现了过拟合现象。因此在挖掘时序特征的时候，需要进行适当的特征集约。笔者经过试验，统一将时分秒合并成为一个time特征，另外不引入年、月特征，只保留日以及是否为工作日的指示特征。

另外，考虑到两次购物行为之间一般会有一定的间隔时间，因为购物对于个人而言存在固定成本，即登陆网站、付款等耗费的时间以及其他因素，因此一次购物会购买超过当期消费所需要的量。同时在用户未流失的情况下，两次有效购物间隔的时间也不会过长，比如日常用品有固定的消耗时间、偶尔商家会举行促销活动来刺激消费等。尽管间隔时间因人而异，但是从大量的统计规律中可以挖掘出一般消费者两次有效购物发生的间隔时间分布情况，对于判断某次购物的真实消费量是重要的。因此笔者接下来构建一个新的数值型变量——表示该用户上一次购物距离这一次访问的时间间隔，单位为天。

另外笔者还注意到，登陆谷歌商城的用户来自世界各地，但是谷歌商城则是直接以其当地时间即UTC时间来进行记录，这样会导致用户当地时间和记录时间在日期上可能会出现一天的差别，可能直接导致“是否为工作日”的这个指示特征出现错误。从图3.19可以看出，以UTC时间记录的分布和以顾客当地时间记录的分布有着明显的不同。因此笔者将大陆、子大陆以及国家、城市按顺序拼接成一个新的特征，使用谷歌地图API 解析功能解析该特征获得其经纬度特征，并且使用谷歌地图API 中的时区转换功能进行转换。另外一个解决方法是统计同一个国家中购买量或者浏览量最多的时间，以之作为该国家18:00的时间（因为下午18:00属于一天之中最高峰的购买活跃时间），该方法较笔者使用的方法粗糙，但是简便许多，不失为一个巧妙的方法。

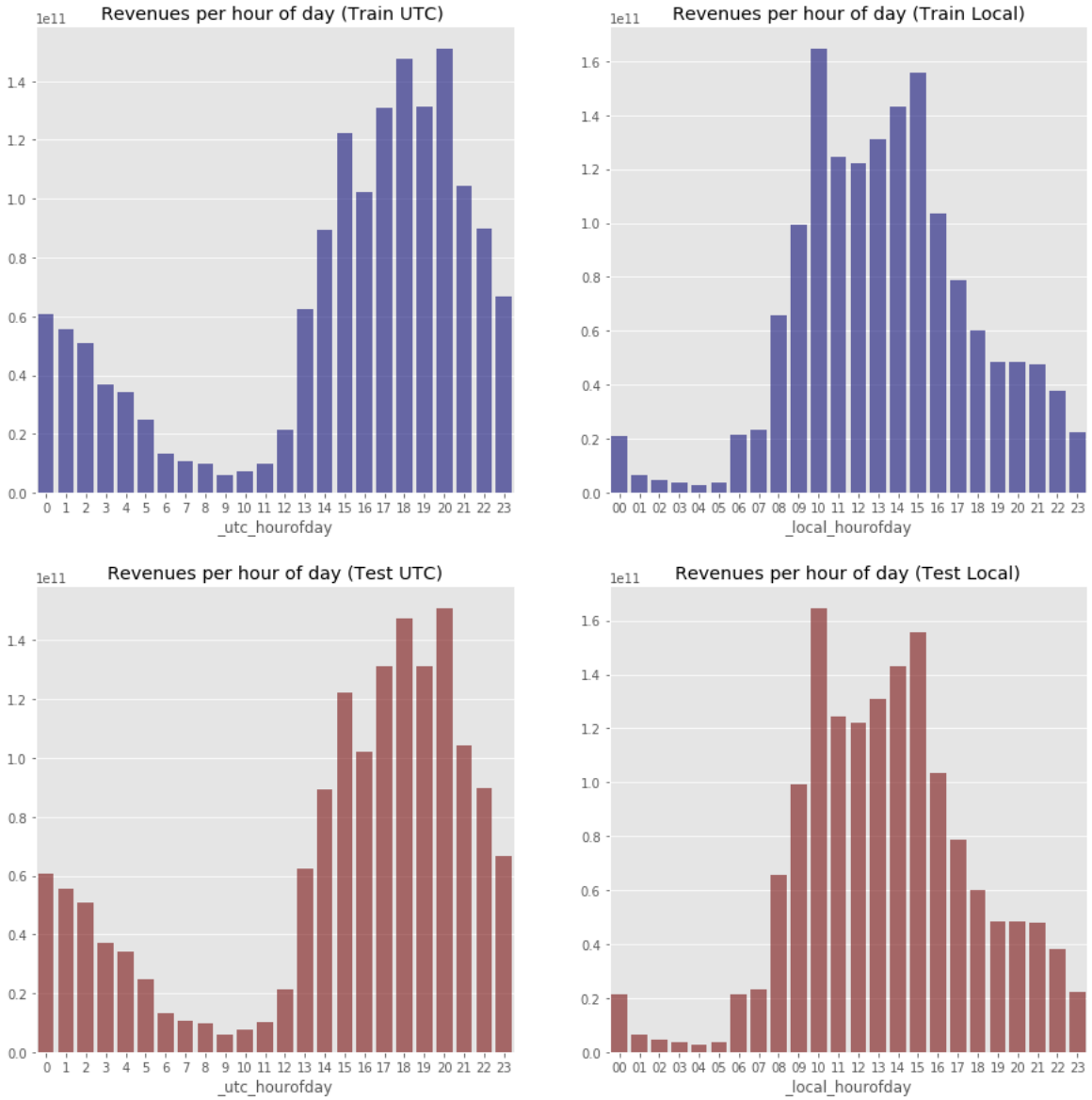

图 3.19 UTC时间和进行时区变换后的需求分布情况

§3.6.2 行为特征

从之前的LightGBM模型所给出的特征重要性可以看到，单次访问浏览页面数和点击数对于消费量的预测都是具有重要作用，模型甚至认为行为特征的重要性可以排入前三名。在一般的特征工程中，可以对这些行为特征的高阶项进行探索。但是该特征构造方法主要适用于回归类模型，即通过构建高阶项增强模型捕捉非线性关系的能力，但是在本文中笔者主要采用的树模型，树模型本身的非线性分割性、非参数化模型构造方式就决定了不需要进行该类特征构造。另外行为特征的交互项也可能会给模型效果带来提升，但是考虑到该数据集中行为特征含义较为类似，且行为特征数量不多，因此不做考虑。

但是行为特征在某一个用户身上的相对稳定的，即对于经常喜欢逛网上商店的用户而

言，每一次浏览的页面数一般会比不经常浏览的用户多。因此从行为特征的一些统计量可以较为稳定地刻画该用户的类型，从而利用用户本身类型做出预测。因此笔者构建了三个特征——平均每次访问浏览页面数、平均每次访问点击数以及用户在数据集中登陆谷歌商城的次数。最后一个特性则是直接刻画用户对于谷歌商城的依赖程度。

§3.6.3 类别性特征

在之前的数据探索中也发现，地理位置、设备情况等类别变量对于判断用户消费量也是具有明显的效果。但是笔者发现，其中大量特征变量具有重复意义，比如country和subcontinent、continent，即国家、子大陆和大陆，很明显这三个特征可以集约化为一个。另一方面，不同于行为特征，类别性特征所描述的维度十分广泛，涵盖用户使用的设备、使用的浏览器种类、网络端地址等，其中的交互效应可能会对模型的提升具有重要作用。因此笔者通过对特征变量进行"特征1_特征2" 的拼接实现类似于数值型特征的交互项特征构造。具体而言笔者构造了如下类别性交互特征：

- browser+deviceCategory,即考察使用设备和使用的浏览器种类的交互效应
- browser+operatingSystem，即考察使用浏览器种类和使用的操作系统的交互效应，因为在麦金塔系统中一般默认使用Safari浏览器，但是如果仍坚持使用谷歌的Chrome浏览器的用户，可能对于谷歌产品也具有较强烈的依赖感。
- adContent+country，即考察不同国家和所看到的广告类型的交互效应，即便是同一个广告，在不同国家、不同文化背景之下所代表的含义也不一定相同
- medium+country，即考察不同媒介和国家的交互效应，因为在不同国家有不同的使用方式习惯，比如在美国普遍使用谷歌搜索，但是在中国可能大部分人还是使用百度搜索进入，而使用谷歌搜索的人相对的可能会对于搜索质量具有更高要求，因此其他方面的特性可能也应该有所区别。

另外城市、国家、网络端这些表示地理位置的特征和设备种类、浏览器种类、操作系统、进入谷歌商城的来源等描述设备方面的特征，它们之间由于是从不同维度刻画用户特性，因此其交互项也依然值得研究。因此笔者使用其笛卡尔积再构造了20个交互特征。

§3.7 模型构建

由于预测任务最终是要预测每一位目标用户的消费量，但是数据只提供了每次访问的用户特征数据(下称会话级别)。尽管可以通过会话级别的数据预测出每一次访问所产生的消费量，然后再对每一位用户进行加总，但是这样就损失了用户特性。因此笔者提出两阶段预测模型，即先针对会话级别进行一次预测，然后将预测之后的数据构建成为新的用户级别的数据集，再进行进一步的预测。

§3.7.1 会话级别

该级别的模型属于第一阶段，因此所有训练需要的特征就是前文提及到的原始特征以及笔者构建出的新特征。值得注意的是，针对类别性特征笔者并没有进行one-hot编码，而是直接赋予其1,2,3...等值，one-hot编码在这样规模的数据集上需要大量的内存，笔者所拥有的算力无法支持该操作。另外更为重要的是，树模型并不会受到受到不同编码之间大小关系的影响，因此笔者直接采用该编码方式。

模型采用LightGBM模型，不同于之前使用LightGBM判断特征重要性，此处模型进行了5折交叉验证训练过程，即将数据随机切分为5等分，每次取其中的四份进行训练，最后一份进行验证，根据验证集上表现最好的模型来对全数据集进行一次预测。因此最终5折训练完后，实际上已经完成了在5份验证集上表现最好的5个模型对全部数据集的预测，一共5次预测。根据集成学习的思想，将这5次预测进行简单平均后作为最终的预测数据。会话级别的模型训练完毕后，最终的RMSE=1.5501。也就意味着经过特征工程以及更为细致的调参、交叉验证之后取得了较大的进步。

§3.7.2 用户级别

根据残差预测和集成学习的思想，上一个模型的输出可以作为下一个模型的输入以提升预测的准确性。为此采用以下方式构建用户级别的数据集：对所有已有特征都根据用户进行均值，以消费量为例，某用户在数据集中一共进行了5次消费，那么在新的数据集中将这5次消费量取平均作为该用户的新的特征，其他特征也采用相同方式构建。另外第一阶段对于用户的消费量已经进行了一次预测，因此可以直接加入每一位用户的所有历史会话中预测的消费量。另外为了更加高度描述预测的型，笔者按照每一位用户取平均获得的平均预测消费量加入新数据集中。但是取平均的方式会损失大量预测分布的特性，因此通过加入从更多维度描述分布的统计量－－中位数、中位数和总和来丰富预测信息。因此最终加入的新特征为：

- 所有已有特征的均值
- lgbpred_1,lgbpred_2,$\cdots$表示该用户第一次会话中预测的消费量，第二次会话中预测的消费量，以此类推
- t_mean表示用户所有访问中预测的消费量均值
- t_median表示用户所有访问中预测的中位数
- t_sum表示用户所有访问中预测的消费量加总

新数据集中目标变量就是按照每一位用户的加和消费量。最终用户级别的数据集有360个特征。

使用新数据后，再次使用LightGBM模型从新数据集训练以获得特征重要性评价如图3.20:

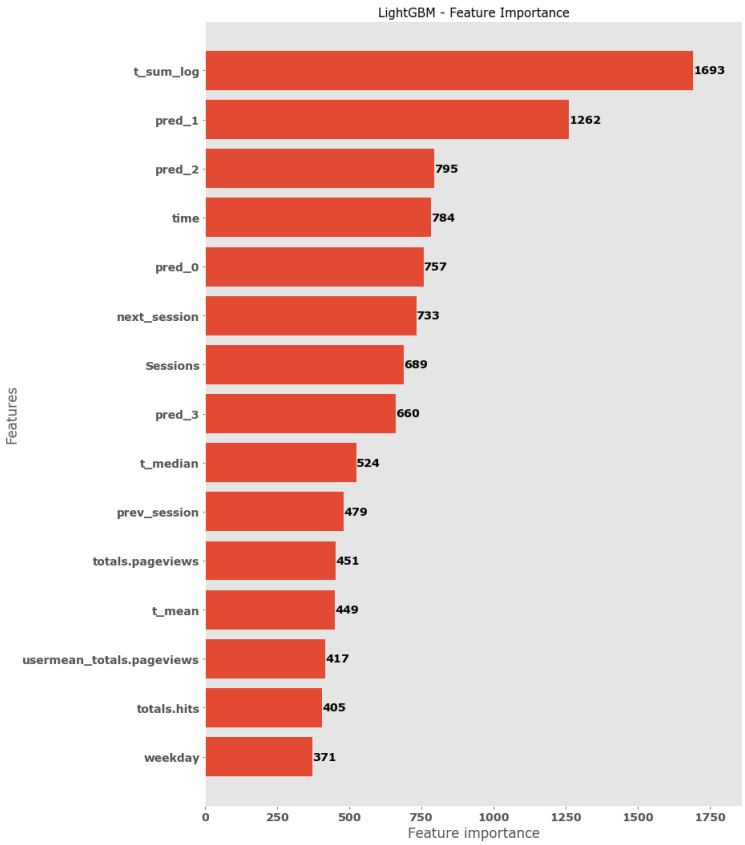

图 3.20 特征重要性

从图中可以看出，对LIghtGBM模型预测最重要的前15个特征中，虽然依然有之前的page views、hits、weekday等原始特征，但是大部分特征都是使用特征工程新构造出来的特征，next_session即距离下一次访问的时间间隔，prev_session即距离上一次访问的时间间隔，以及weekday即是否为工作日。另外重要性排名前三个特征——预测值加和、第二次访问的消费量预测、第三次访问的消费量预测，以及其他例如pred_0，t_median等，这些都是在用户级别模型中的输出数据，因此也表明了笔者提出的两阶段模型第一层模型构造特征的有效性。

§ 3.7.3 Stacking

Stacking是Kaggle比赛中常见的集成学习框架。一般来说，就是训练一个多层(一般是两层)的学习器结构，第一层(也叫学习层)用n个不同的分类器(或者参数不同的模型)将得到预测结果合并为新的特征集，并作为下一层分类器的输入。一个简单的示意图如图3.21。在stacking中，通过第一层的多个学习器后，有效的特征被学习出来了。从这个角度来看，stacking的第一层就是特征抽取的过程,第二层则是基于抽取后的新特征进行最终的预测。为了达到提高预测精度的效果，多个分类器应该尽量在保证效果好的同时尽量不同，以分别从不同角度获取数据中的信息，因此stacking 集成学习框架第一层对于基预测器的两个要求：差异化要大和准确性要高。在这一点上看，Stacking 对于效果的特征其本质跟神经网络并无太

大不同，都是通过多层表征式学习提高效果。

但是Stacking的第二层则要求模型尽可能简单，因此Stacking本身有很高的过拟合风险，在第一层通过大量模型尽可能捕捉数据信息之后，第二层需要引入简单的如线性回归模型来进行最终的预测，并且可以通过加入正则项来进一步降低过拟合的风险。笔者第一层使用了参数差异尽可能大的5个LightGBM，4个XGboost和3个Catboost。其中Catboost也是一种目前主流的实现GBDT算法的框架，其特点在于能够很好地处理类别型特征的梯度提升算法库。首先对所有样本进行随机排序，然后针对类别型特征中的某个取值，每个样本的该特征转为数值型时都是基于排在该样本之前的类别标签取均值，同时加入了优先级和优先级的权重系数。这种做法可以降低类别特征中低频次特征带来的噪声。其一，它在训练过程中处理类别型特征，而不是在特征预处理阶段处理类别型特征；其二，选择树结构时，计算叶子节点的算法可以避免过拟合。

第一层的模型多样性主要包括改变其树的最大深度来尝试过拟合以及欠拟合，改变抽样的比例、改变抽取特征数的比例、改变学习率等。第二层笔者尝试了比赛当中常用的xgboost，以及笔者选择的Lasso回归，最后是笔者自己手动进行的简单加权平均。其中Lasso回归模型是在普通线性回归模型损失项中加入 $gamma \sum_{i=1}^{n} \beta_i$，即对模型的复杂度进行惩罚，以提高模型的泛化能力。最终训练时间超过20 小时，使用xgboost 最为第二层模型在训练集上得到的RMSE=1.4694，而使用$\gamma = 0.05$ 的Lasso 回归得到的RMSE=1.4961, 而使用简单的加权平均得到的RMSE=1.4962。三者均明显低于会话级别的模型训练，也就意味着第二阶段的训练对于最终效果的提升确实具有重要帮助。同样很明显可以看出xgboost 取得了最低的RMSE，但是从提交结果来看，反而是最为简单的加权平均取得了最好的成绩，也就意味着xgboost的确存在着过拟合的现象。

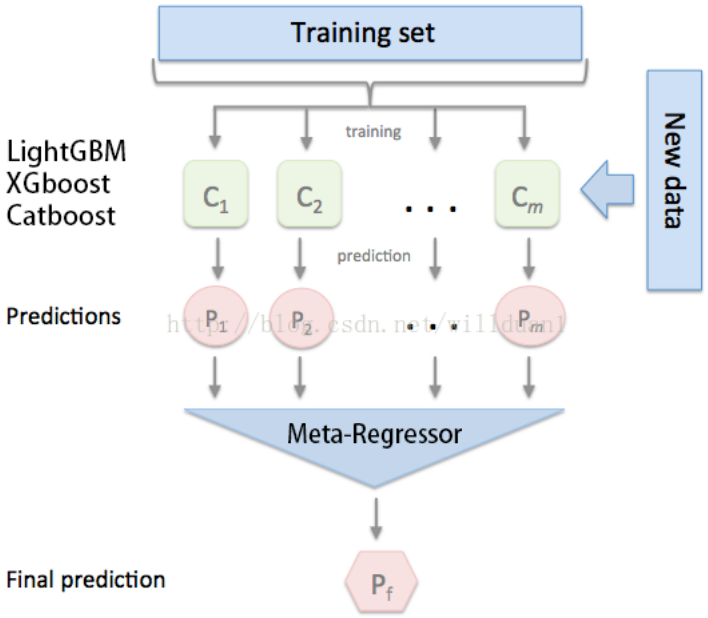

图 3.21 最终预测模型图

§3.8 结论

在本次项目中，笔者从零售界用户需求预测入手，通过分析谷歌线上商场2016年8月至2017年8月实际销售的顾客行为特征数据以及消费记录，根据顾客访问级别的行为特征、地理特征、设备特征、网络特征等，经过了完整的数据探索、数据清洗，以及多维度的特征工程、特征集约，基础模型构建和比较、模型调优、整合模型和集成学习等流程，实现了对谷歌商城用户级别的消费量预测，从比赛排行榜上看，本文流程与模型构建比较有效，取得了3238 支参赛队伍中的第96名，也就是前3% 的好成绩，同时也验证了特征工程中对于原始数据构建方式的有效性。本文的研究工作如下：

- 在特征构造层面，通过多阶段特征构造过程，即第一阶段直接从原始特征入手，从时序性特征、地理性特征、设备型特征等多维度进行有针对性的特征构建，通过加入“未来特征”、类别特征交互项，以及时区转换，实现异常值检测和处理、数据规约、特征规约，确保了模型的泛化能力和提高了训练速度。

- 在初步模型构造层面，通过引入决策树、随机森林、xgboost、神经网络以及LightGBM等新型框架，分析比较它们在需求预测方面的优劣，并通过网格搜索展示了超参数的调优过程，并通过参数调优得到了显著的模型提升，为之后更为复杂的多模型调参工作提供帮助。

- 提出两阶段需求预测模型，即在访问层次的需求预测模型之上，构造新的特征并作为第二阶段的模型特征输入，最后再次使用Stacking集成学习方法进一步捕捉如此大量特征之中丰富的信息，并使用简单的第二层模型确保模型的泛化能力，预测效果比起一阶段模型有了显著的提升。

通过描述性的分析和模型的特征重要性评分，最终筛选出了一些对于预测具有重要作用的特征，因此可以对管理者需求预测有如下启示：

- 关注用户两次访问的时间间隔，可以构建更为精准的分布模型描述其访问间隔分布，从而准确捕捉会产生实际消费的顾客进行相应的营销

- 用户在本次访问的行为特征对于判断用户消费量具有重要作用，因此可以记录更加丰富的用户行为特征，比如在每一个页面的停留时间、在广告上的停留时间等

- 进行预测的时候要注意当地的时间，可以建立更为精确的模型描述各地购买活跃度的时间分布。

- 进行营销的时候必须注意用户的当地文化特性，不同国家甚至不同城市的消费习惯特别是网上购物习惯不同，每次消费行为也有不一样的特征，应该进行更加细致的营销活动

- 需要对每一位用户构建相应的用户画像，实时跟踪每一位用户的兴趣变化

虽然本文考察了不少目前主流的特征构造方法以及算法框架，但是仍存在以下几点不足之处需要进一步改进：

- 数据中出现的异常情况由于无法知道其出现的准确原因，只能强制性删除，可能会导致信息遗漏
- 最终集成学习中基本都是采用树模型，没有进一步考虑特征之间更为高阶的交互效应。可以考虑引入FM模型和神经网络模型，增强模型对于高阶交互效应的捕捉能力
- 对超参数调节不够充分。由于算力和时间有限，本文最后仅能采用人工调参的方式，在算力允许的情况下可以尝试贝叶斯调参或者遗传算法调参

通过这次历时长达两个月的课程项目，笔者从课程开始就计划课程论文写作，并参与世界最顶尖的数据挖掘平台Kaggle比赛中，在此期间学习到了大量目前主流的工程界的数据挖掘方法，并从供应链的实际问题出发，以一个较大规模的数据入手，完成了完整的数据挖掘流程，课堂知识也得到了检验和提升，并且更加激发笔者对于数据挖掘的兴趣和热情。第一次参加Kaggle比赛就能获得前3%的成绩也是远超出笔者预期，希望今后能够继续加深学习，并且利用Kaggle等优秀的数据挖掘竞赛平台，实践结合理论不断深化，最终能够探索出供应链需求更加精准的预测方法。过程中一共提交了超过40次成绩，其中大部分提交都没有能够提升效果，但每一次提升都是振奋人心的，排名情况以及对应的线上成绩RMSE总结如图3.22

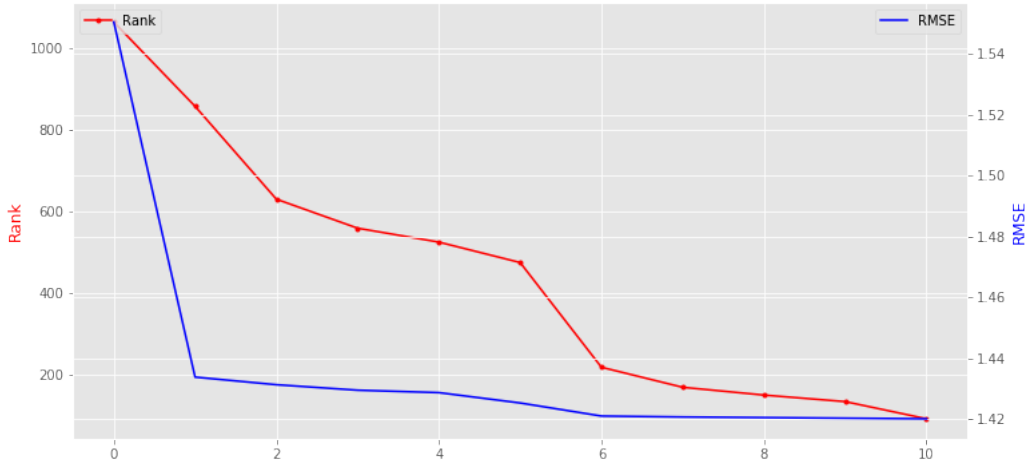

图 3.22 RMSE和对应的排名图

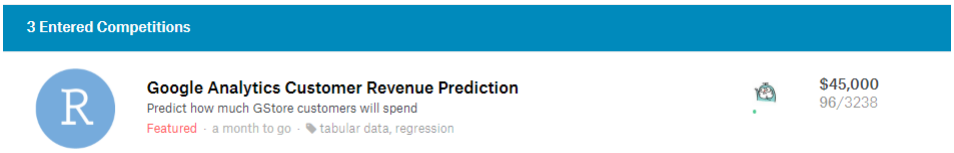
3 Entered Competitions
Google Analytics Customer Revenue Prediction
Predict how much GStore customers will spend
Featured · a month to go · tabular data, regression
$45,000
96/3238

图 3.23 最高排名图

致　　谢

本课程论文得到中山大学张宏斌老师的耐心指导，以及岭南学院信息管理中心对我提供服务器的算力支持。没有老师的理论教导和相应的设备支持，我难以完成如此规模的数据挖掘任务并最终取得好成绩。非常感谢老师的支持和指导!

参 考 文 献 表

[1] Lee, Hau L., Venkata Padmanabhan, and Seungjin Whang. "Information distortion in a supply chain: The bullwhip effect." Management science 43.4 (1997): 546-558.

[2] Cao, Jin, Zhibin Jiang, and Kangzhou Wang. "Customer demand prediction of service-oriented manufacturing incorporating customer satisfaction." International Journal of Production Research 54.5 (2016): 1303-1321.

[3] Tsai, Chih-Fong, and Mao-Yuan Chen. "Variable selection by association rules for customer churn prediction of multimedia on demand." Expert Systems with Applications 37.3 (2010): 2006-2015.

[4] Aburto, Luis, and Richard Weber. "Improved supply chain management based on hybrid demand forecasts." Applied Soft Computing 7.1 (2007): 136-144.

[5] Fiala, Petr. "Information sharing in supply chains." Omega 33.5 (2005): 419-423.

[6] Sheu, Jiuh-Biing. "A multi-layer demand-responsive logistics control methodology for alleviating the bullwhip effect of supply chains." European Journal of Operational Research 161.3 (2005): 797-811.

[7] Taylor, James W., and Roberto Buizza. "Using weather ensemble predictions in electricity demand forecasting." International Journal of Forecasting 19.1 (2003): 57-70.

[8] Tso, Geoffrey KF, and Kelvin KW Yau. "Predicting electricity energy consumption: A comparison of regression analysis, decision tree and neural networks." Energy 32.9 (2007): 1761-1768.

[9] ZHANG, Xue-fei, et al. "Prediction of urban water demand in Tangshan City with BP neutral network method [J]." Journal of Safety and Environment 5.5 (2005): 95-98.

[10] 后锐, 张毕西. "基于MLP 神经网络的区域物流需求预测方法及其应用." 系统工程理论与实践25.12 (2005): 43-47.