互联网平台智能风控实战

王永会 / 著

清華大學出版社
北京

内 容 简 介

风控是互联网平台业务的重要环节。随着业务的迅速发展，黑灰产问题逐渐突出，大数据和人工智能技术的普及为智能风控提供了强力支撑。本书以作者实践经验和总结为基础，介绍了搭建智能风控系统对抗黑灰产的方法。全书共10章，按如下思路组织内容：

（1）**认识问题**：第1章介绍常见的黑产类型，解密黑产运转的内幕，带读者认识建立风控体系的必要性。

（2）**分析问题**：第2章分析风险范围和种类，提出智能风控的系统框架和需要的各项能力，从整体上介绍智能风控系统，避免一开始就陷入技术细节。

（3）**解决问题**：包括第3~8章，介绍搭建智能风控系统需要的软硬能力，包括理解业务和避免过度黑盒的实用方法、数据建设、常用技术手段（监督学习、迁移学习、GNN等）、量化评估和可视化效果呈现。

（4）**总结与展望**：包括第9章和第10章，总结搭建智能风控系统的注意事项，简单展望智能风控技术的发展趋势和前景。

本书内容源自作者的工作实践和经验总结，适合风控从业人员（技术管理者、分析师、算法工程师、产品经理）以及其他对互联网风控感兴趣的人员阅读。

图书在版编目（CIP）数据

互联网平台智能风控实战 / 王永会著. —北京：清华大学出版社，2022.1
ISBN 978-7-302-59434-5

Ⅰ. ①互… Ⅱ. ①王… Ⅲ. ①网络公司－风险管理－研究 Ⅳ. ①F490.6

中国版本图书馆CIP数据核字(2021)第225964号

责任编辑：王中英
封面设计：郭 鹏
责任校对：胡伟民
责任印制：宋 林

出版发行：清华大学出版社
网 址：http://www.tup.com.cn，http://www.wqbook.com
地 址：北京清华大学学研大厦A座 **邮 编**：100084
社 总 机：010-62770175 **邮 购**：010-83470235
投稿与读者服务：010-62795954，jsjjc@tup.tsinghua.edu.cn
质 量 反 馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
课 件 下 载：http://www.tup.com.cn，010-83470236
印 装 者：三河市铭诚印务有限公司
经 销：全国新华书店
开 本：185mm×260mm **印 张**：24.25 **字 数**：595千字
版 次：2022 年 1 月第 1 版 **印 次**：2022 年 1 月第 1 次印刷
定 价：99.00元

产品编号：089197-01

推荐语

随着现代互联网科技的发展，以互联网平台为支撑的业务遍地开花，由此引发的黑色产业不断滋生，营销、欺诈、信用风险等方面都面临一定的挑战，如何建立并迭代企业的智能风控体系显得尤为重要。

《互联网平台智能风控实战》一书全面介绍了风控体系建立的背景、如何理解业务场景、数据的处理流程、常用风控手段以及实际案例，从系统开发到业务实践，全方位展示了智能风控系统在互联网平台的实战应用，深入浅出并系统地讨论了智能风控体系进化历史和发展全景，对于想要快速了解智能风控的从业人员是一本难得的参考书，同时对于业务及研发人员具有很好的参考和借鉴价值。

耿艳坤，顺丰速运集团 CTO、顺丰科技 CEO

《互联网平台智能风控实战》一书侧重讲解搭建智能风控系统的思路框架，引导读者从全局视野了解风控，从业务全流程去思考，采用多种技术路径去实施，并归纳出智能风控管理应做到全链路覆盖、依托丰富的大数据、从业务源头上减少风险、灵活管控，非常具有参考价值。

唐会军，数美科技 CEO

在互联网的江湖里，黑灰产是一股神秘又恼人的力量，而与黑灰产的对抗又是互联网平台无法回避的问题。《互联网平台智能风控实战》一书结合作者多年的一线作战经验，给出了一整套与黑灰产对抗的智能风控体系，既能从技术手段给出实操建议，又能跳出技术，从更高视角来理解风控，从整体上思考解决方案，相信会给读者带来有益的启发。

巴川，竞技世界首席数据科学家、CCFTF 数据科学 SIG 主席

随着互联网领域技术的快速发展，网络黑灰产已经形成成熟的产业链。据测算，网络黑产从业人员已过百万，市场规模达到千亿元级别。同时，黑产技术和风控手段也在随着攻防对抗过程不断演变，虽然有关黑产以及攻防技术的文章发表不少，但是都缺乏系统性。《互联网平台智能风控实战》一书作者长年在一线工作，在同黑产对抗中积累了丰富的实战经验，对黑色产业链有着深入的研究和总结。本书首先介绍了黑色产业链，然后从业务、数据、规则防控、实战、评估、平台等维度系统讲解了对应的风控解决方案，深入浅出，是当下业务安全领域少有的专业图书，非常值得一读。

陈成，快手业务安全负责人

推荐序一

在硝烟四起的互联网平台大战结束之后，留给人们最深的印象可能就是无处不在、越来越多的平台补贴了。商家收获了潮水般突然涌来的用户，用户体验到了远高于价格的超值服务，平台快速扩大了市场占有率，投资人眼看着公司估值节节攀升——大家似乎各取所需，皆大欢喜。然而，盛宴之后很少有人会真正关心互联网产业的长期健康发展，更少有人会认真思索如何借助基于数据和算法的先进技术手段，去挤出繁荣里的虚假泡沫，维护平台上的良性增长。

在《互联网平台智能风控实战》一书中，作者基于自己在一线互联网平台公司的长期实战经验，帮助大家建立起一套科学地收集、加工、评测和呈现大数据的智能风控方法论，能够指导工作在互联网行业不同场景下的业务、产品和研发人员，能够更加及时、精准和全面地掌握商业活动的真实状况，制定更能贴合实际的商业决策。在当今互联网平台经济迅猛发展的风口浪尖，本书所倡导的“技术要理解业务、服务产品”的思路，将能从根本上有效对抗黑产风气，最终带来整个行业的有序发展。

智能风控技术虽然来源于数据和算法，但又不限于冷冰冰的规则和指标。风控技术对抗的是利益背后的人性。相信读者们能够从本书的实战经验中，体会到那一场场攻防对抗战之下隐藏的复杂利益关系和算计取舍。智能风控之所以能够有效地建立起保护正常商业行为、促进合理商业生态屏障的逻辑，最终依赖的也是技术专家在设计模型与系统时，综合考虑数据的价值、损益的平衡、信息的隐藏和权力的制约，等等，这些都是需要反复博弈的因素，也是本书最引人入胜的地方。

所谓魔高一尺、道高一丈，互联网平台上的智能风控战争永远不会有终结的一天，不过这也是这项事业最让人着迷的一点。希望大家跟随作者对智能风控技术发展趋势的解读，更加深刻地理解这门学问中需要反复揣摩的门道，在未来更为广阔的数据安全治理空间中，创造更大价值，发现更多商机。

蒋凡，京东科技集团智能城市副总裁、《智能增长》作者

推荐序二

为什么互联网平台需要智能风控？智能风控是什么？如何系统化地构建智能风控体系以应对平台运营中面临的各种风险挑战？这三个问题不仅需要我们从理论层面去探索和研究，更需要我们从各互联网平台的业务实践中去总结和认识。

互联网的发展极大地推动了我国数字经济的发展，与此同时，平台在拉新、留存促活、交易、内容生产与传播等运营环节也面临着越来越严峻的欺诈风险。黑产遍布在互联网的各个领域，无论一个平台提供的是什么样的产品服务，都会成为黑产的套利目标。

如果平台做拉新活动，黑产就会造大量的虚假账号“卖”给平台；如果平台做促活活动，黑产就会造大量的虚假活跃“卖”给平台；在下单交易中，黑产可能采用盗刷、恶意退款等方式，让平台货、款两空；如果平台上存在有价值的数据，黑产就会盗爬数据并进行倒卖；如果平台存在榜单，黑产就会通过提供“刷榜”服务来获利；如果平台存在 UGC，黑产就会通过发送大量违法诈骗广告进行套利。总之，对于黑产来说，产品是什么不重要，利益才是唯一的思考角度。面对每天金额庞大的损失，对平台来说打击黑产刻不容缓。

最近几年，网信办对互联网行业监管趋严，重拳打击了很多涉政治敏感、黄赌毒以及三俗内容的产品。互联网不是法外之地，在此背景下，加强互联网平台业务和内容风控意识，建立完善的风控制度，构建以反欺诈反垃圾为核心的风控系统和机制显得尤为重要。而互联网的飞速发展也让我们清醒地认识到，对于不同规模、不同实力及处于不同经济发展水平的平台来说，风控的侧重点、切入点及路径也应有所不同。平台需要从自身条件出发，制定适合自身情况的规划。

本书侧重讲解搭建智能风控系统的思路框架，引导读者从全局视野了解风控，从业务全流程去思考，采用多种技术路径去实施，并归纳出智能风控管理应做到全链路覆盖、依托丰富的大数据、从业务源头上减少风险、灵活管控。

本书分为 10 章，从黑产、解决方案、业务、数据、模型、应用、工具、痛点、挑战与未来等角度，对互联网平台智能风控进行了精辟的总结分析。

最后，特别感谢王永会先生邀请我为本书写序，数美科技作为一家专业的在线业务风控解决方案服务商，将持续关注智能风控领域，为塑造智能风控商业价值和实现健康向上的互联网生态贡献微薄的力量。

唐会军，数美科技

前 言

互联网平台业务发展迅猛，产品迭代以快节奏见长，营销推广多以线上为主，这些特点是平台业务的优势，同时也深受黑灰产的喜爱。黑灰产有发达完善的情报和监控体系，也有成熟配套的刷量工具和研发资源，任何产品的漏洞和玩法都会被黑灰产在短时间内研究得明明白白，刷量一触即发。

本书所讲的风控面向的就是与黑灰产对抗的问题。时至今日，黑灰产已经发展得非常成熟，软件架构和基础工具都能够根据不同产品和业务快速调整定制。反观风控，在不少中小企业中遇到刷量问题时，既无风控人员又无风控措施，只能临时抱佛脚，花费人力、物力不停地趟路，研究对抗之法。实际上很多大型互联网公司早已积累沉淀了成熟的方法论，但由于风控内容较为隐秘，无法较为直白地系统化描述。本书试图通过笔者在实践中的一些经验体会，讲述构建智能风控系统的粗略框架，希望能够给还在趟路的从业人员提供参考。

本书不追求具体潮流的某个技术、某个模型来解决单个具体案例，而是强调搭建智能风控系统的思路框架，引导读者从整体上思考解决方案，从更高的视角来理解风控。全书共 10 章，按照如下思路组织内容：

（1）**认识问题**：第 1 章介绍常见的黑产类型，解密黑产运转的内幕。

（2）**分析问题**：第 2 章提出智能风控的系统框架和需要的各项能力，从整体上介绍智能风控系统，承上启下。

（3）**解决问题**：包括第 3~8 章，介绍搭建智能风控系统需要的软硬能力，包括理解业务和避免过度黑盒的实用方法、数据建设、常用技术手段、量化评估和可视化效果呈现。

（4）**总结与展望**：包括第 9 章和第 10 章，总结搭建智能风控系统的注意事项，简单展望智能风控技术的发展趋势和前景。

书中第 6 章介绍了一些场景和案例，一方面便于读者更好地理解技术手段的应用，另一方面也想通过这些案例说明一个问题：任何一个被黑产盯上的场景，想要彻底解决问题，都需要多手段、全链路地防控。书中多次强调数据打通，有些案例的解决方

案读者可能会觉得有些别扭，原因就在于数据无法获取。笔者认为大数据的维度丰富性一定程度上需要跨产品跨平台的数据打通才能做到，体现到业务上就需要全链路的防控，全链路防控也是避免单节点不断暴露问题的必经之路；而智能则只是把大数据的维度发挥出来的一个技术手段。

在写作之初，笔者信心满满。写到中间处，已觉自己站在愚昧之巅，随着在工作中认知的不断变化，书中内容写了删、删了又写，一度进入绝望之谷。书中内容多是自身体会，难免有认知错误之处，恳请广大读者批评指正。

书中很多想法的落实，离不开领导和团队成员的大力支持，在此诚挚感谢耿艳坤、崔代锐、蒋凡、戴少伟的指导和培养，感谢刘梦宇、谭星、徐龙飞、魏尧、李子杰、田会会、张正龙等兄弟们的努力和贡献。起笔容易，坚持不易，时间和精力有限，感谢家人的理解和支持。

谨以此书献给我的孩子。虽然书中内容还有很多可完善之处，虽然拖拖拉拉持续了很长时间，但最终还是坚持写完了，希望你长大以后能克服惰性、坚持长期主义，不要贪图瞬时满足感，哪怕学个滑板车，想要灵活自如地滑行也需要长久地多多练习。

王永会

2021 年 10 月

目 录

第 3 章
核心：理解业务、服务于产品

第 4 章
关键：数据的重要性

第 5 章
手段：规则、模型和监控

第 6 章
场景：反制手段的应用

第 7 章
评估：损失与收益的平衡

第 8 章
管理平台：直观的可视化工具和管控工具

第 9 章
风控的挑战与智能风控系统的搭建原则

第 10 章
风控的未来技术

第 1 章

解密黑产

这是一个最好的时代。互联网如此普及，市场有无限可能，大量的创业公司崛起，生活服务无不能连接到互联网，让人有了更多想象。这也是最坏的时代。伴随着新型业务的出现和增长，黑色产业（以下简称黑产或黑灰产，本书不对黑产和灰产做具体区分）正在野蛮生长，而每个企业在初期都会缺乏风险控制意识，都曾为此付出过沉重代价，Uber 打车、拼多多、ofo、外卖、无数的 P2P 平台、无数的广告主……

1.1 认识黑产 <<<

羊毛党、水军、僵尸粉，这些都是大众熟悉的黑产，因为他们能够被大众接触到或者感触到。然而黑产远不只这些，他们早已经形成了庞大的产业，始终紧盯着企业营销运营的手段，并让企业付出巨额代价。

1.1.1 黑产的危害

据不完全统计，中国网络的黑色产业规模已达千亿元的级别，手段五花八门，刷单让平台防不胜防，因被刷单而导致重大损失的案例举不胜举。图 1.1~ 图 1.4 分别是 Uber、外卖、ofo、消费金融被黑产侵蚀的例证。从中可以看出，刷单黑产已给企业造成严重的经济损失，刷单严重者，每日给企业造成的损失高达千万元，对于很多创业阶段的小企业来说，可谓致命一击。为此，国家针对刷单事件进行了立法支持，每年因为刷单而追究刑事责任的案件不在少数，全国第一个刷单案由杭州市余杭区人民法院公开宣判，“90 后”刷单者李某某因犯非法经营罪被一审判决五年六个月，连同原判有期徒刑九个月并罚，决定执行有期徒刑五年九个月。2017 年 11 月 4 日全国人大常委会通过了新修订的《中华人民共和国反不正当竞争法》，对刷单相关的法律条款进行了完善，明确规定了惩罚力度。


刷单危机下Uber：每日被刷走金额几乎上千万

2015年06月29日 08:42 北京商报 微博 我有话说(90人参与) 收藏本文

【推荐阅读】法国出租司机反Uber大罢工 巴黎抗议活动酿暴力

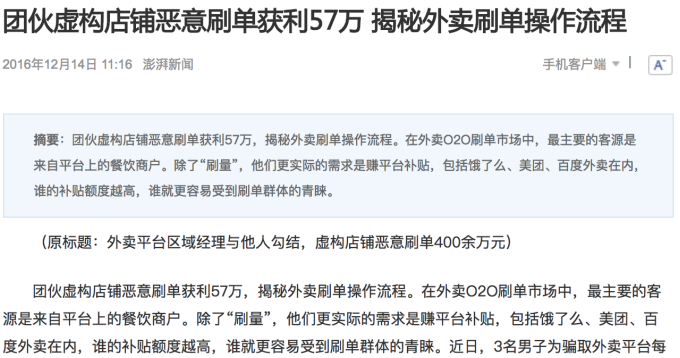
刷单危机下的Uber

Uber CEO特拉维斯·卡兰尼克不久前刚刚透露“Uber每日在中国市场的订单量逼近100万单，相当于6个月前Uber全球业务总量”。

但随着刷单灰色产业链的曝光，Uber到底有多强或者Uber冤大头到底有多冤，似乎成了这家跨国公司在华绕不过去的槛，而这背后除了技术和制度原因，难以脱身的中国互联网式价格战也许是最大的坑。

图 1.1 Uber 被刷单（图片来源于网络）

刷单影响到了每个互联网人。刷单严重的平台失去了消费者信任；面临刷单不公平竞争排名的情况，平台中安分经营的商家被逼着刷单，形成恶性循环。传统线下场景中，消费者可以看到实物，通过观察可以与商家建立初步信任，而互联网时代，消费者看不到实物，往往通过平台的信用体系（如评分、评论）做出判断。刷单破坏了这个体系，久而久之破坏了平台的生态。



团伙虚构店铺恶意刷单获利57万 揭秘外卖刷单操作流程

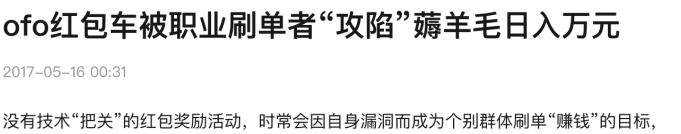
2016年12月14日 11:16 澎湃新闻 手机客户端

摘要：团伙虚构店铺恶意刷单获利57万，揭秘外卖刷单操作流程。在外卖O2O刷单市场中，最主要的客源是来自平台上的餐饮商户。除了“刷量”，他们更实际的需求是赚平台补贴，包括饿了么、美团、百度外卖在内，谁的补贴额度越高，谁就更容易受到刷单群体的青睐。

（原标题：外卖平台区域经理与他人勾结，虚构店铺恶意刷单400余万元）

团伙虚构店铺恶意刷单获利57万，揭秘外卖刷单操作流程。在外卖O2O刷单市场中，最主要的客源是来自平台上的餐饮商户。除了“刷量”，他们更实际的需求是赚平台补贴，包括饿了么、美团、百度外卖在内，谁的补贴额度越高，谁就更容易受到刷单群体的青睐。近日，3名男子为骗取外卖平台每单几毛钱到几十块不等的订餐补贴，在两个月内疯狂刷单400余万元，获利57万元。目前，该犯罪团伙的主谋田某被上海普陀警方抓获。（12月14日）

图 1.2　外卖被刷单（图片来源于网络）



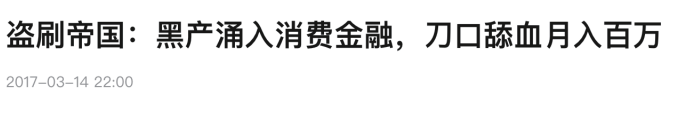
ofo红包车被职业刷单者“攻陷”薅羊毛日入万元

2017-05-16 00:31

没有技术“把关”的红包奖励活动，时常会因自身漏洞而成为个别群体刷单“赚钱”的目标，ofo红包车再次证明了这点。近日，ofo红包车刚刚上线就被职业刷单者“刷单”，甚至有网友声称每天刷ofo红包车日入万元。

4月16日，ofo共享单车推出红包车活动，在其平台APP内寻找带有 红包标识的区域，在该范围内解锁车辆骑行超过10分钟、距离达到500米后，便可领取现金红包，红包最高金额达5000元。

图 1.3　ofo 红包车被刷单（图片来源于网络）



盗刷帝国：黑产涌入消费金融，刀口舔血月入百万

2017-03-14 22:00

近两年，消费金融上升为互联网金融的头把交椅，火爆异常。一群盗刷银行卡的黑产人员，在消费金融崛起后，尾随而来。他们整合各个渠道泄露的用户信息，就像完成一幅拼图般，精心拼凑。一旦锁定目标，他们招数百变地专营各种漏洞，配合新式设备，进行大规模清洗。

图 1.4　金融黑产（图片来源于网络）

然而黑产之大，绝非刷单一种形式，盗刷、诈骗、攻击、木马等花样层出不穷。消费金融的空前繁荣，让黑产有机会顺水推舟，涌入新兴产业中。黑产人员能通过搜集并整合各个渠道泄露的用户信息，用完成一幅拼图般的耐心，精心拼凑出每个用户信息。银行卡盗刷、透支让很多受害者几近倾家荡产，还要忍气吞声。目前信用卡相关法律尚不健全，持卡人的信用卡被盗刷后，需要向银行提供非本人刷卡的证明。即使是这样，很多银行要求在没有破案前，持卡人先对盗刷金额买单。除了盗刷，欺诈更是不容忽视，据第三方统计，消费金融领域，超过 50% 的损失是由于欺诈导致的。欺诈即骗贷，某医美分期平台负责人曾公开表示，整个医美市场的贷款量大概是 60 亿元，其中就有 15 多亿元被骗贷者们攫取。黑产危害，可见一斑。

1.1.2 黑产的产生

1. 有利益的地方就有黑产

新型互联网事物在推广期往往会采用补贴的手段吸引新客加入，补贴变相产生的诸如红包、赔付、优惠券等形式，成为黑产的关注焦点。创业者在业务不断扩张、求生存的阶段，重点更多放在产品的体验和业务不断完善上，加之在产品设计环节缺少抗风险意识，导致黑色力量与正常用户同时成长，甚至超越了正常用户的体量而使企业破产的情况也时有发生。外卖行业里的满减优惠（指用户下单达到一定金额后减免一定额度），本意是吸引用户下单，提高客单价，但如果能打破满减次数限制，或者买卖双方串通起来，就会演变为作弊；金融领域的借贷，通过盗用他人的身份信息或者购买虚假信息骗贷；出行领域的“幽灵”司机则是团伙作案，牟取暴利（见图 1.5）。

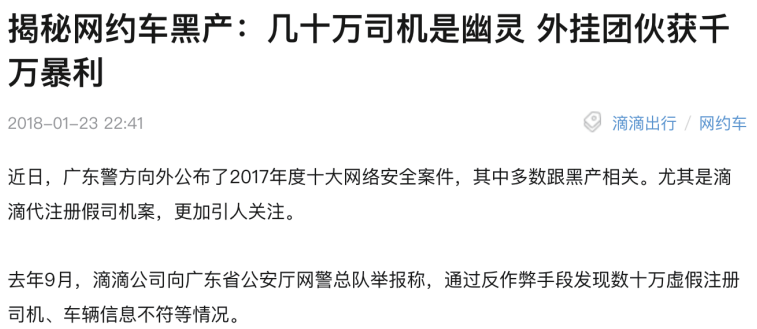


图 1.5　网约车黑产（图片来源于网络）

2. 泡沫效应

利益的驱使让黑产总有可乘之机。让很多人想不到的是，一些表面的繁荣其实只是海市蜃楼，主动刷单在很大程度上又助力了黑产的发展。一直以来，刷榜刷量行为是业内的一个比较隐晦的话题。应用程序开发者在应用商店上架自己的 App 后，苦于下载量、安装量上不去，开始尝试寻求刷榜的力量，一旦看到效果，便会变本加厉。在这个产业链中，刷榜已是常态（见图 1.6），就像正常商品一样明码标价，不少公司为了曝光自家的 App 都有此行为，刷榜行为几度让苹果应用商店更新了排名算法。微信公众号刷量事件更是戳破了行业泡沫，电商平台、外卖平台的投机商家也主动寻找中介帮自己提高销量。刷，已成为一些个体 / 企业宣传推广的常用手段，已成为一些群体的固定职业，已成为一些推广机构、中介平台的生存手段。在这个泡沫中主动要求刷量的行为助长了黑产气焰，而因此带动的黑产繁荣会给他们带来大量的损失。

因此，不论是平台漏洞导致的被动利益吸引，还是主动的“推广营销”行为，都为黑产的产生和增长提供了空间和助力。这两方面却又存在矛盾：针对平台漏洞，平台往往需要考虑风险控制措施来打压黑产；而主动的“推广营销”却让黑产继续“合理”存在，反过来再把技术手段应用于漏洞导致的利益，可谓环环相扣，冤冤相报。

图 1.6 App 刷榜

除此之外，国内征信体系的不完善和大数据风控人才的欠缺，也给黑产的发展提供了机会。大数据风控的普及在国内还面临诸多痛点和难点，甚至被挂以虚名、实则无风控，让大数据风控在监管层面遇阻，这些都为暗处的黑产带来了便利。

1.1.3 黑产的类型

黑产都是为了利益而存在的，根据平台业务涉事方参与黑产能否直接获得金钱利益，可以分为两类：

（1）直接套利型。主要的形式包括外卖平台上的套补贴，金融类的骗贷，共享单车的刷红包，打车行业的套补贴、线上推广的刷下载量和安装量，广告的刷曝光点击量等，以及为达到此目的而产生的虚假实体（如虚假用户、虚假商户、虚假司机、虚假媒体等）。这种类型的涉事方一般通过作弊、刷单，或者通过接活中介直接就可以获利。

（2）间接套利型。主要的形式包括电商中的刷量、刷排名，社交媒体上的刷榜、刷量，社区中的水军，以及以广告诈骗、垃圾灌水、黄色信息推广为主的内容型黑产等。这种类型的涉事方一般不能直接获得金钱利益，中间往往还需要转化，如刷排行的因为排名靠前导致更多曝光机会，进而带来更多新增用户；在贴吧发古怪文字加链接的水文，因为用户单击链接的次数增多而带来更多下载量。当然对于接活的黑产工作室来说，不存在直接、间接的概念，总是可以获利的。

根据黑产活动对业务的影响，又可以分为业务安全型和内容安全型：

（1）业务安全型。指对业务正常运转有影响的黑产，会导致业务的资产损失、生态

破坏（劣币驱逐良币）以及体验变差、信用口碑下降等问题。包括上述的直接套利型和间接套利型中的刷量部分。

（2）内容安全型。主要是指内容不符合监管要求，以及不满足平台的内容管理规范和价值观的黑产。黑产作恶后的直观表现是在图、文、视频上肉眼可见。

事实上，不同的黑产类型在作案手法上差别较大，对抗的理念也会不同。直接套利型的黑产在不同业务场景上虽然手法区别不大，但与业务结合后，会产生“变异”，想要复制其他业务上的对抗方法，成本依然相当高，不过其命脉在于“利”，在获利空间上下功夫可以实现很好的效果。业务安全型的黑产在数据表征上就是异常，它们与绝大多数正常用户行为不同，因此，对抗业务安全型实际就是在做异常发现。

1.1.4 黑色产业链

黑产在实际操作过程中需要多方配合，他们以盈利为目的，有组织、分工明确地团伙式作案，追求低成本、高回报。根据分工不同，主要包括以下利益方。

（1）信息收集人员。主要负责盗取用户身份证号、银行卡号、手机号、手机 SIM 卡、微信账号等隐私信息。可以通过技术手段（如拖库、撞库等）偷到用户信息，也可以通过灰色交易收购大量旧手机、手机卡（一般称之为卡商）等囤积号码。手机卡一般来源于物联网卡、虚拟运营商的未实名卡（卡商可以通过收集网络数据进行实名化）、海外卡以及企业内部违法倒卖的实体卡号等，其中虚拟运营商的未实名卡（黑卡）占大比例。细数这里面的人员，包括办黑卡的卡商、收黑卡的卡商、制作猫池设备（参见图 1.7）的厂家，以及号商（出售微信账号、支付宝账号等）。随着国家对隐私信息的保护力度加大，以及用户隐私保护意识的增强，以上身份信息获取难度大了很多，但依然有很多企业内部人员，利用职务之便非法倒卖用户数据。2020 年 7 月，铁岭市公安局破获了一起兜售实名微信号的黑产案件，数额巨大，而数据的来源就是某企业数据中心工作人员和某平台工作人员。

图 1.7　猫池设备（图片来源于网络）

（2）验证码供应商。主要负责提供短信验证码、语音验证码。卡商提供的手机卡通过有通信能力的猫池设备接入验证码接码平台，专门用来收发短信，根据获取渠道不同，每条短信的费用从几分钱到几块钱浮动。通过自动化管理软件，管理众多手机号在不同平台的出现次数，同一个手机号可以在多家平台以新用户身份使用。此外，一般还配备可视化平台，对验证码发送请求量、成功率等指标有较为详细的监控，资费标准明码标价。提供这种服务的平台数不胜数，如易码、讯码、51 接码、E 码、火云、芒果、神话等，关键还有数不清的低调到没有名字的平台。这两年国内很多公司开始瞄准国外市场，针对国外的短信接码平台也涌现了一大波。图 1.8 所示的是正在使用易码平台接收短信验证码，图 1.9 所示的是 Z-SMS 提供的在线短信接码服务，无须登录即可看到短信内容。

图 1.8　易码平台接码示例（图片来源于网络）

#	电话号码	短信详情
1	106**********1224	【ZAO】你的验证码是612783
2	106*******3083	【ZAO】你的验证码是012298
3	106**********5870	【众安科技】您的验证码是598307，请在5分钟内完成验证。
4	106**********3201	【优益快借】尊敬的用户，您的登录验证码为:7857

图 1.9　Z-SMS 在线短信接码

除了短信验证码和语音验证码平台，还有打码平台，如图 1.10 所示。在注册或异常登录环节，一般会设置图片验证码来屏蔽机器人，这种图片中含有字母、数字或者汉字等，人肉眼很容易识别，但是对于机器来说，需要大量的样本进行算法训练进行识别。而打码平台既提供机器识别能力，又提供人工解码能力，即雇佣人去解码，标注和解码一举两得。参与打码的人一般称为打码工，按打码数量计算收入，当然也会计算正确率。

图 1.10　某打码平台（图片来源于网络）

（3）技术研发人员。主要负责模拟器、手机参数修改器、定位修改器、按键精灵等软件研发，这类技术研发并不一定是为了用于黑灰产，但却可以为黑灰产所用。例如，手机参数修改器可以做到一键修改手机参数；定位修改器可以让手机定位“穿越”到任何想去的地方，并可以模拟出轨迹。目前国外已经出现了采用人工智能技术进行刷好评的做法，效果逼真。国内一些电商平台也开始出现技术刷好评的现象，可以预见，未来黑产领域也将智能化。这不，以前黑灰产用单机加抹机神器（图 1.11）就可以刷出一片天地，今天就连模拟器也换成了云手机，采用云指令技术，云端操作，可实现一台手机变多台，可以一键新机，可以改定位，成本也就日均一块钱。

图 1.11　抹机工具

（4）刷单中介 / 工作室。主要负责制定刷单攻略、联系客户、组织并实施技术刷单操作。把刷单攻略制作成详细可操作的教程，一份教程可卖几十元到上百元，通过 QQ 群、微信群等社交工具雇佣刷单人群。中介对各大平台的业务流程都很熟悉，承包任务，伪造信息，甚至公司内外勾结。目前很多工作室除了有很多“拉活儿”的销售商务，也具备信息采集、接码、技术研发等能力，能够在模拟器、云手机、按键精灵、脚本、木马等方面做很多研发工作。

（5）刷手。主要负责具体的刷单操作，这类群体中不乏学生、家庭妇女、乡村农民

等各类型的刷单职业工作者和不知情被利用的人群，职业者往往遍布于各个刷单的论坛社区、QQ 群、微信群中。

图 1.12 描述的即为黑色产业链的上下游，其中每个角色在实际中并不是单纯负责一个环节，比如卡商可以提供号商的能力，接码平台本身也可以兼作卡商。产业链中主要涉及设备、卡、号和软件，整个产业链围绕的核心就是解决这些资源的货源问题，而刷单中介 / 工作室主要是解决资源整合和人的问题。所以，做黑产与开一个网上店铺很相似，以前是自己解决货源问题，自己拉新做活动，而现在平台可以提供很多赋能工具。

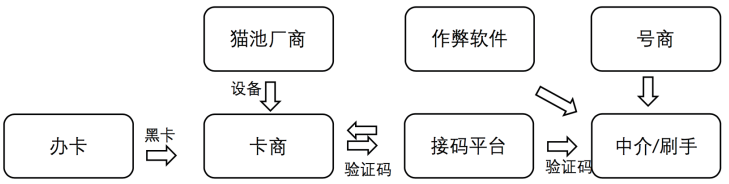


图 1.12　黑色产业链

1.1.5　黑产的焦点

只要有利益的地方就有可能存在黑产，但从黑产的角度看，是否值得黑，关键就在于事情的成本与收益的衡量，所以黑产往往聚焦于那些低成本高收益的行业，比如互联网金融行业、电商行业、O2O 行业。对于低成本低收益的行业，通过量的积累，也可以快速达到高收益的效果，比如初入市场阶段的创新型互联网产品，如 ofo 共享单车、滴滴打车等。针对相对稳定的互联网产品，因新增功能的推广涉及的利益引诱依然是黑产关注的焦点，比如支付宝推广阶段的分享红包。

据笔者不完全统计，目前黑产关注最多的行业有互联网金融、电商、O2O、社交、新兴行业、游戏（游戏行业的黑产与前面几个行业的黑产有很大区别，主要涉及外挂、私服、黑卡、盗号、挂马等手段，因笔者经验有限本书较少涉及）。

1. 互联网金融

互联网金融中有两类比较常见的欺诈场景，一是刷新用户，二是假身份借贷（俗称骗贷）。自从 2013 年余额宝引爆互联网金融以来，各种金融产品层出不穷，为了吸引更多用户加入，各个平台都在新用户注册这块砸下重金。刷单者利用手机黑卡到各互联网金融平台大量注册新用户，平台补贴的新用户奖励大量落入刷手口袋中，效果大打折扣。骗贷曾经形成了专门的产业链，由贷款中介推动，办理假资料，伪造账单、消费记录，钱一到位就彻底消失。消费分期和现金贷等小额贷款已成重灾区，骗贷者往往使用同一批资料，利用平台之间的信息不互通，短时间内在多家平台连续骗贷。目前业内逐步建立了数据共享机制和行业黑名单，防范骗贷现象，国家在对待金融风险方面也下了很大功夫，2019 年以后，互金相关的黑产随着行业发展逐步落幕。

2. 电商和 O2O

电商和 O2O 领域的刷单基本是相通的，一般联合商家刷优惠、刷信誉、刷销量、刷排名等。刷优惠吸走了平台的资金，直接导致经济损失，其他的刷单则对平台的评价体系注入垃圾，损害平台口碑、误导用户，影响平台整体的生态平衡。在电商和 O2O 领域，一般涉及 B 端（商户）、C 端（用户）、物流和销售四部分，涉及的业务链条长，产品细节多，单一角色防控难以产生效果，往往一波未平一波又起，因此是黑产关注的行业里最复杂的一类。本书的很多案例也主要围绕电商和 O2O，后文也会重点介绍针对此类场景的风控解决方案。

3. 社交行业

社交行业的黑产主要在社交平台大量注册小号，从事发广告、刷粉、刷阅读量、充当网络水军、传播色情内容、进行网络诈骗等。被黑产关注最多的社交平台主要是流量大的渠道，如微信、QQ、微博、陌陌、贴吧等。

4. 新兴行业

不断涌现的新兴行业始终是黑产关注的重点，行业新生的产品往往因为快速抢占市场，在风险控制方面比较薄弱。典型的例子如共享单车、打车、众筹、P2P、区块链。2017 年 4 月，ofo 推出红包车，打开 ofo 手机 App，在红包区范围内开锁，且骑行时间超过 10 分钟、距离达到 500 米，即可获得现金红包。因为 ofo 没有 GPS 定位，无法定位到每一辆小黄车，而且用户结束行程不用锁车，所以用户只要在规定的“红包车区域”内，选取任意一辆小黄车，输入 ofo 车牌号（甚至可以输入不在此区域的红包车）并获取密码，就能开启“骑行”状态了。操作十分容易，很快成为职业刷单者眼中的肥肉，正因为如此轻松就能赚钱，导致大量用户也加入刷 ofo 红包的活动中，可谓损失惨重。

随着新兴行业的逐渐成熟，风险会逐步可控，但又会继续出现新的行业，黑产就像野草般“春风吹又生”。

1.2 黑产运作 <<<

通过 1.1 节的介绍，我们对黑产及其产业链有了一定认识。黑产范围广大，产业链涉及诸多利益方，这些角色在欺诈链条中是如何配合运作的呢？孙子云，知彼知己，百战不殆。只有摸清黑产的运作才能制订正确的风控方案。实际上，在不同的领域中，实施欺诈的差异很大，很难统一而论，但从采用的伎俩上看并无明显区别，不论是电商和 O2O 领域的刷单，还是广告领域的欺诈，作案的手法都是一致的，因此这给我们研究黑产运作和欺诈相关的风控减少了不少成本。本节将从 O2O 外卖行业中的欺诈切入，阐述黑产链条的运作过程。

1.2.1 外卖领域的黑产运作

1. 外卖业务流程

黑产难以防范的原因有很多，其中一个是它伴随正常业务流程，产生于其中的细枝末节，也正因如此，实施风险控制往往需要**非常熟悉业务**，一则要了解业务中的风险所在，二则避免风控措施误伤正常流程。这是做好风控工作的首要前提。

读者朋友对外卖的业务流程应该不会陌生，但实际上据笔者调查发现，绝大部分人熟悉的只是在线上点一份外卖的流程，除此之外并不了解太多。

1）用户熟悉的外卖业务流程

一般来讲，打开手机上的外卖 App，会看到一个附近的商家列表，用户选择自己喜欢的商家，进而选择适合自己口味的菜品加入购物车，确认下单，下单前会填写收餐人姓名、联系电话以及收餐地址，而后在线支付即可完成下单操作，如图 1.13 所示。这个过程与在电商平台上购物并无明显区别，是任何一个有过点外卖或者网上购物经验的人都很熟悉的流程。

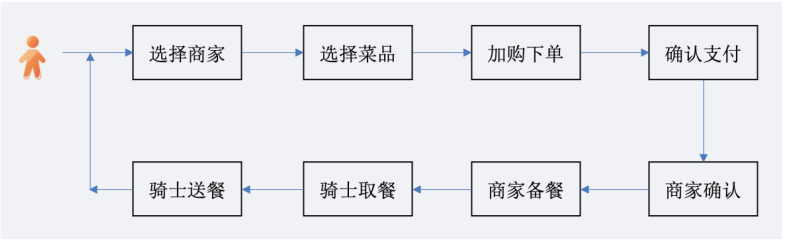


图 1.13 用户视角的外卖业务简易流程

商家收到用户的下单请求后，根据实际情况确认订单，然后开始备餐（准备发货），外卖平台的后台调度系统会把订单任务分配给某个骑士（物流），骑士赶至商家处取到餐，最后送到收餐地址处的收餐人手中（配送）。虽然与电商在物流调度环节稍有不同，但不影响整体过程的一致性。

2）用户视野之外的角色——BD

到这里，都是普通用户比较熟悉的流程，但这个流程已经涉及了三个角色：用户、商家、骑士。习惯成自然后往往容易忽略两点，一个是平台中的商家是怎么入驻到平台中来的，另一个是下单的用户是怎么来到平台的。这就引出了另外一个角色，我们称之为 **BD（Business Developer，业务拓展人员**，下文会把商务、销售、业务等人员统称 BD）。BD 的职责主要包括：签约商家入驻到平台、市场营销、拉新（此处更多指线下拉新，线上拉新一般由运营人员负责），具体介绍如下。

（1）商家入驻平台是需要通过一套信息系统录入很多信息的，包括商家的门脸照、后厨照、营业执照、卫生许可证、身份证、结算使用的银行账号、商家地址、名称、联

系方式、线上营业时间等几十项信息，考虑到大部分商家不擅于使用信息系统，加之很多信息填写不规范等情况，为了方便快捷，信息录入一般由 **BD 协助完成**。

（2）**市场营销**是指在业务发展过程中，为了提高市场占有率，针对商家和用户开展的一系列营销活动，比如经常看到的满减活动。BD 有权限进行活动的配置。

（3）**拉新**是指为了吸引源源不断的新用户，BD 开展的地面推广活动。例如，在路边、校园、商场或者商家门口，撑起易拉宝（海报架），拉几张桌子，通过送小礼物的方式吸引用户下载 App 然后下单。

笔者注意到，因为骑士的工作时间往往集中在午高峰和晚高峰，其余时间段相对空闲，因此曾经出现过骑士拉新的做法，具体执行形式不限，这倒是一个不错的思路，只可惜由于执行原因，收益甚微且风险较多。

针对线上拉新，除了运营人员的策划和推广，还有付费形式的推广合作，即买量，比如资讯类 App 中的广告投放，在使用这类 App 时，信息流中会掺有一定数量的广告，点击广告后会让用户下载另一个 App。

从以上流程中不难看出，外卖的线上业务虽然简单，但背后的运作过程还是很复杂的，涉及了用户、商家、骑士、BD 等角色，这也正是 O2O 的特点。

2. 暗涌

用户在平台上向商家下单，这一正常行为在黑产这里变得很有意思。以“满 50 元减 20 元”的优惠活动为例，当用户购物车中的菜品满 50 元时，不考虑配送费和打包费的情况下，用户只需要支付 30 元。而此时商家收到的是多少呢？实际上，商家收到的依然是 50 元，给用户减免的 20 元则由平台补贴给了商家（即平台烧钱）。此处便是**黑产的焦点**所在，我们来看这个过程，如下。

如果下单用户与商家串通，那么商家不需要备餐，并返还用户支付的 30 元，同时再额外给用户一笔 5 元的“辛苦费”，那么商家和用户分别获得 15 元和 5 元的“好处”，可谓空手套白狼（假设不考虑平台抽佣）。前面提到了订单会被后台调度系统分配给骑士，骑士到商家处取餐，商家可以告知骑士餐已被用户自行取走，骑士或许觉得略微惊讶，但自己省了一趟配送，自然也很乐意接受商家的说辞。此情况遇见多了，骑士很快就会发现商家的秘密，有人说骑士可以选择举报，没错，但这与举报车辆违法行驶类似。于是骑士与商家达成一致，收到此类订单，商家不用出餐，骑士也无须再白跑一趟来取餐，从此骑士和商家双双过上了幸福的生活。

再来讲讲 BD 可能做的动作。

BD 有权限进行活动配置，当然可以与商家串通，倾斜补贴力度。“满 50 元减 20 元”调整为“满 50 元减 30 元”，与商家各自分得 5 元好处，何乐而不为？

BD 签约商家时协助完成信息录入工作，这里也成了舞弊的点。除了完成自身 KPI

需要而造假，BD 自然也可以算清楚用户与商家合谋的收益，这样只需拥有几个“商户”，就可以同时兼收多重收益了。

地面拉新可以说是 BD 负责的事情中最有油水的。针对拉新的考核，一般有**新用户量和复购率**两项指标，快速发展期甚至只有新用户量一项指标，这很容易出现作弊。对于新用户的“新”如何定义，这也是很关键的。仅以一维账号（如手机号）作为判定标准的情况非常容易被刷，此时 BD 与刷单中介合作，轻而易举完成拉新任务。拉新中另一类风险便是低效拉新，BD 往往通过廉价小礼品吸引老年人下载 App 下单（实际是拿老年人手机代操作），因为这类群体本身不会操作也非目标用户，造成极低的复购率，导致资金低效使用。

3. 小结

从上述的暗黑流程中，我们可以得出以下几点结论。

（1）职业刷单者以用户身份介入，其他角色虽产生于业务内部，却是黑产的重要发动机。

（2）用户和商家可以串通刷单，即商家与职业刷手合谋，当然也存在商家与散户薅羊毛的串通，但实际中规模和危害远小于前者。这种模式也是电商领域的刷单主流。

（3）骑士和商家可以串通刷单。

（4）BD 可以和商家、职业刷单者串通。

实际上，还有更多，如商家自身也可以作为用户来刷单，即商家自导自演；其实任何其他角色自身都可以是刷单用户；更进一步，BD 可以同时兼任 BD、商家、骑士多重身份，自导自演一场关于虚假商户下的虚假订单如何虚假配送的大戏，订单的所有信息不过是按照正常流程走了一遍，但实际上并没有线下（offline）的环节。笔者在现实业务中确实发现不少 BD 利用系统间信息不打通的漏洞，身兼数职牟取利益。

在外卖业务中，除了众所周知的用户薅羊毛（仅以用户身份进行的作弊和欺诈，不依赖其他角色的配合），更多的是上述业务内部角色（商家、骑手、BD）与外部职业刷单者的串通联合。实际上这种现象不仅出现在外卖中，其他行业里外勾结式的黑产模式也不在少数。

1.2.2 出行领域的黑产运作

1. 正常业务流程

出行行业中涉及的角色相对较少，主要是乘客和司机，可以将他们类比为用户和商家，即 C 端和 B 端。如图 1.14 所示，乘客打开 App，输入自己的出发地（一般是定位自动识别）和目的地，发起打车请求，司机在附近范围内抢单，抢单成功后到乘客出发地接上乘客，

然后驶往目的地；到达后由司机结束行程，用户发起支付流程，一个订单到此便完成了。司机的信息注册自行完成即可，不需要 BD 的参与。因此，相较于外卖场景，业务流程上不涉及过多线下环节。

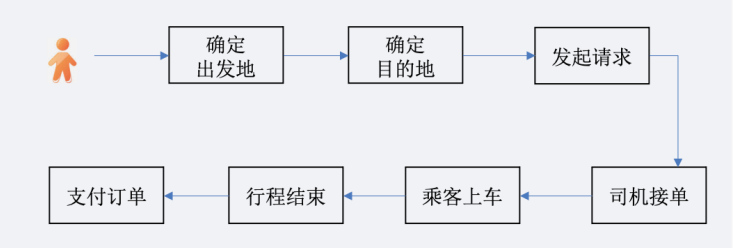


图 1.14　用户视角的出行业务简易流程

2. 刷单运作过程

既然一个订单同样涉及 B 和 C 两端，刷单也必然存在 B 和 C 的串通联合。这其中存在两种情况：

（1）通过增加发单乘客来起量，这与一般的薅羊毛手法并无不同，通过注册大量账号，结合接码平台和刷机软件即可做到。

（2）大量伪造虚假司机，通过非法购买他人信息批量注册司机。

两种情况均需要作弊软件修改 GPS 位置信息，模拟行程轨迹。

除此之外，还有类似于外卖中的商家与用户同身份的手法薅羊毛。2015 年 4 月和 8 月，滴滴出行分别向北京、上海公安机关报案：部分账户异常，存在一人同时担当司机、乘客两重身份，出现多单司机与乘客账户重复、虚构打车交易的现象。

1.2.3　金融领域的黑产运作

与外卖和出行领域相比，互联网金融领域的欺诈运作稍有不同，金融领域的欺诈（金融产品众多，这里主要指借贷相关业务）发生在借贷者与平台之间，缺少了 B 端这一环，因此，黑产主要的工作在于如何获取可以用来借贷的身份信息，而不像上述两类场景中存在 B 端与 C 端勾结串通，但是黑产为了使获取的身份能够成功借贷，在平台上运作留下的表象往往具有相似性。主要的黑产玩法如下。

（1）多头借贷。即向多个平台借贷。起源于拆东墙补西墙的做法，而黑产可以用来投机，往往同期多头借贷。提供借贷的平台多且信息不互通时，就给黑产提供了便利。相关调查发现，小额现金贷人群中，有多头借贷行为的用户占比超过 50%。由于网贷信息不记入央行的征信系统，网贷平台之间信息共享程度又较低，所以导致了多头借贷的爆发。多头借贷本身并非不可取，关键取决于最终**是否还款**，不还款才是黑产的特点，

因此如果对多头借贷行为一棒子打死，会误伤有真正需求的用户，这给风控提出了挑战。

（2）中介“助”贷。主要有两类，一是中介负责伪造信用记录等信息，帮助那些资质不过关的用户申请贷款，挣取提成。另一类则是张罗人，本质上是借用他人的信用，往往是在农村地区，付给参与的农民一笔好处费，让他们去申请贷款，实际借贷最后落入中介手中，俗称刷村。每人申请额度一般不大，靠数量累积，多数平台会放弃对这种情况的催收。

（3）职业薅羊毛。每个行业中总有一批黑产用户始终盯着每个平台，一旦发现口子，通过他们的大本营（一般是论坛、QQ 群、微信群等）快速传播，薅一笔便消失。

（4）员工作案。企业员工通过获取自己公司内的大量员工信息卖给外部黑产团伙，黑产团伙再用这些信息骗贷。

（5）技术手段。包括用注入、撞库结合网络收集拼凑的方式获取用户信息。

金融领域的产品形态众多，涉及的黑产欺诈手法也远不止这些，对黑产而言，只要能拿到可用的信息，任何方法都值得尝试，毕竟金融行业中黑到一笔钱的收益，远高于前面介绍的外卖和出行行业。

1.2.4 黑产的特点

了解了黑产的产业链和运作过程，不难发现黑产就像一张极其庞大的网，又像一个极其庞大的商业帝国。它在人员构成上具有等级和分散的特点，在运营上具有专业化、职业化、利益链复杂、团伙化和伪装性特点。

（1）专业化。黑卡、猫池设备、自动化管理软件、接码平台、群控系统、代理、各色各样的作弊软件、完整易操作的作案教程，以及对各种行业漏洞细节的把握，都体现了黑色产业的专业性。

（2）职业化。黑产分工明确，组织有序，不少已形成公司规模，明目张胆招聘各种岗位，既有售前销售人员拉活儿，又有后台研发人员研发自动化工具。据称，O2O 领域曾出现代理商下某 BD 成立刷单公司的惊人之举，从事刷量、下载、安装激活等的工作室数不胜数。

（3）利益链复杂。黑产运转的各个环节涉及不同利益角色，上下游之间存在利益关系；既有流程上的配合，又有内外串通共同牟利之举。前面案例中多次存在一端角色身兼多职的情况，这其实是黑产自身缩短利益链条的做法。

（4）团伙化。黑产发展为“以人为本”的操作模式后，极易形成团伙作案。加之风险口子在黑产圈快速传播的效应，人工黑产模式主动或被动地形成了团伙化。这对于风控来说既是好事又是坏事：好处是便于识别发现；坏处是团伙一旦得逞，规模较大、损失较大。

（5）伪装性。一方面一线的刷手可能是正常的人，比如人肉刷单模式、诱导行为；另一方面程序化模拟逐渐逼真，导致异常行为越来越接近于正常用户行为。这对风控的发现和识别工作带来了很大挑战，既不能误伤正常用户，又要尽最大范围召回异常行为和用户。

1.3 黑产技术演变 <<<

本节并不讲述黑产技术的发展演变过程，而是介绍黑产找准目标后，在一款产品上不断与平台的风控政策抗衡的过程。

黑产的手段变化一方面取决于平台的风控力度，另一方面受限于成本与收益的考量。实际上，真正在一线执行欺诈的黑产群体——职业刷手仅是这个庞大组织中最不具备技术能力的一方。得益于互联网，为这些职业大军提供后盾的角色业务精细且五花八门，有专门负责注册账号、提供 IP 资源的，有研发自动发评论工具的，有提供手机号和短信验证的，有研发手机窜改软件的，有研发定位修改器的，等等。这些对于职业刷手来说都是武器装备。

1.3.1 风险控制的发展阶段

一般来讲，一个新行业的产品面世后，在风险控制方面会经历如下几个阶段。

（1）无风险控制阶段。这个阶段的产品设计基本不考虑风险，仅仅是在逻辑上做一些简单限制，而这个限制也未必是从风险角度出发的。例如，外卖产品刚上线时允许一个用户每天享受两单优惠，而这个两单的限制更多是从用户用餐习惯和预算角度考虑的。ofo 红包车刚上线时允许一个用户每天随意骑，只要出现在红包区域即可。这个阶段主要是业务刚起步，需要打市场，研发团队中往往也没有风控岗位。

（2）简单风险控制阶段。在上一阶段运转一段时间后，很容易被黑产盯上，往往会曝出被刷单的新闻，或者数据上有明显的异常。此时一般立即采取管控措施，主要以不成系统化的规则为主，并启动以运营审核为辅的控制手段。

（3）稳定的风险控制阶段。损失继续加剧之下，公司开始搭建风控系统，系统化解决黑产问题。该阶段逐渐完善规则和模型，以及人工审核机制，稳定后达到一个可控的平衡阶段。

（4）冷宫与极乐阶段。这是个分裂的阶段，这个阶段的风控能力已经能够快速应对大部分突发事件，有相对之前较为完整的善后流程。但风控的意识在企业内往往也随之被弱化。只要利益诱惑依然存在，黑产技术就会不断演变，但此时风控可能停滞不前，

止于修修补补的重复性工作和人工审核而不得解脱，姑且称之为冷宫阶段。如能始终保持魔道相争不松懈，逐步提高黑产的作案成本、降低防御的人工成本，倒也可以不断进步，通往另一极，极乐阶段。

与这几个阶段对应，技术对抗层面同样呈现出阶段性，从简单规则和黑名单库，到规则系统，到有监督的机器学习，再到无监督学习、知识图谱和深度学习技术，大数据风控也需要一个慢慢建设的过程。

1.3.2 黑产的发展阶段

在不了解它之前，黑产就像一个为了争抢宝座的武林高手，对手用几分功力抵抗，它就用几分功力打击。

（1）简单技术刷单阶段。在无风控的阶段，黑产只需突破简单的限制即可大量获利。还是以前面提到的例子来说，如何才能突破每人每日两单的限制呢？这里的“每人”一般是指一个账号，而刷单者通过账号注册机便可获得不限量的账号。对于账号是手机号的，如不涉及接收短信验证码，只需伪造，否则通过接码平台便可获取。红包车的限制仅仅是红包车区域，通过 GPS 定位修改软件即可把定位移至任何区域。这个阶段对于黑产来说，每单成本最多几毛钱。

（2）技术刷单阶段。这个阶段因为风控的阻挡，黑产只得祭出多种技术手段。例如，如果每人的口径升级为一个账号 + 一个设备，那么刷手就得多使用窜改器，才能把一部手机玩出花。如果继续升级为一个账号 + 一个设备 + 一个支付账号，那么刷手需要购买更多支付账号才能多刷。为了解决这些组合中的维度短板问题，刷手在这个阶段付出的成本会比较高，一般需要准备多部手机，平摊到每单上，成本大概上升至几块钱。对于依赖盗用信息的欺诈来说，撞库也是常用的技术手段。

（3）人工阶段。随着风控力度的加强，当技术手段的成本升高，以至于收益空间较小时，或者技术手段很容易就被侦破时，黑产就会采用人工模式。当然黑产迭代升级的大前提是，利益空间再小也要远大过付出的成本。通过社交手段如 QQ 群、微信群等组织团伙刷单，但参与刷单的人往往来自各地，有些产品对于地理位置有限制，因此还需要配合一定的刷单教程。由于参与刷单的个体都是正常自然人，因此防范起来难度就会加大。以外卖为例，通常会有一个群主坐镇指挥，每天刷哪些商家、哪些人参与刷、什么时间点执行任务、如何选择定位地址、选哪个菜品下单、如何填写收餐地址等细节都布置清楚，刷手完成一单后在群里截图反馈结果，然后结算。人工模式已经成为目前黑产的主流，成本低廉，不易被发现。现金贷以及消费分期业务中的欺诈基本都以人工模式为主。无论是多头借贷，还是中介主导的刷村，均是人海战术。对于目前国内的众多风控系统来说，对人工模式的识别很难取得很好的效果。这就需要从更多角度变相管控，从这个角度来说，风控解决方案不能只以单纯的拦截为目的，这在后文还会继续探讨。

总结一下，无论处于以上哪个阶段，黑产都不会只以一种手段生存，凡是我们了解到的，都是过去的，他们无时无刻不在研究新的方法，目前甚至已经出现了使用人工智能技术刷好评的案例。那么把这些手段写出来还有什么意义呢？一方面作案手法虽然在表象上不同，但在数据层面有相似性；另一方面即便是被用过的手法，他们也在不停地尝试再用，不信你可以故意开放一个“口子”，保证立马就会被围攻，这些方法用在新的产品中又可以饱餐一顿。正因如此，黑产无严格的阶段性，而是混搭并见利驱使，有阶段的只是风控能力罢了。

第 3 章

核心：理解业务、服务于产品

理解业务、服务于产品是智能风控系统搭建的核心。往小了说，风控是一个注重细节的工作；往大了说，风控是影响业务收入的卡点。不理解业务，不熟悉产品，细节工作就难做好，处罚就会做不动。仅仅我们理解业务还不够，还需要让业务人员理解风控。这是一个双向关系问题。

3.1 风控、业务和产品 <<<

理解业务、服务于产品是智能风控系统搭建的核心，但非智能风控系统本身的核心，这一点需要说明。第 2 章中已经多次提到风控需要理解业务，渗透于产品。为什么要这么做？怎么做才叫理解业务呢？

3.1.1 风控工作的生存困境

做风控是相当不容易的，尤其是在业务属性很重（线下环节多）又存在较大风险的行业，是一个公认的“脏活儿”。目前国内的互联网企业中，风控从业者数量相较于其他工种少太多，并且企业中也缺乏基本的风控意识，都是在被“薅”走了很大一笔后才清醒认识，又在长期没有被“薅”后放松了警惕，在这样的背景下，风控工作的生存困境是存在的，归纳如下。

（1）风控工作一定是跨团队的，对协作要求高。在 2.1 节里提到了各种各样的风险，这些风险本身就来自多个团队，因此，风控的工作必然是跨团队协作的。相信大家应该有所体会，由于各团队目标不能高度统一等实际问题，跨团队的工作往往是比较困难的，风控的跨团队工作以其他团队配合为主。再加上国内很多公司对风控缺乏足够的认识，甚至公司中绝大部分员工不清楚风控是做什么。如果不能建立流畅的沟通机制，晓以利害、精准打击风险，推动起来就不那么容易。

（2）做风控对理解公司业务和产品的要求高。如何能很好地跨团队协作，让其他团队知晓利害呢？这就要熟悉业务策略和产品细节，还要能够洞察其中的风险，辅以数据分析。想要精准打击风险，同样需要熟悉业务策略和产品细节，还要清楚黑产的手段，才能把解决方案做准确。

（3）风控团队很容易被挑战和“背锅”。一条策略不慎造成误伤，影响用户、影响业务数据，被其他团队批判并拿来当反面案例（bad case）研究——这是常有的事情，因此而“背锅”的情况也并不罕见，所以风控需要把握准确率与召回率。还有一种经常被挑战的情况——机器学习模型给出的一个黑盒概率分数不好理解。所以风控人员需要理解外界，也需要考虑如何更好地被外界理解。

（4）风控是技术，更是一种业务，需要建立机制来评估风险。既然风控是业务，那么它的运转必然需要某种机制。实际中，业务变化很快，产品不断迭代更新，风控人员难以做到全面了解各个细节，尤其是当公司存在多种产品形态时，这就需要建立一定的机制评估风险。比如需要业务团队配合采集哪些参数、上报哪些数据，以便于统一接入风控服务；处罚权是交给风控团队还是交给业务团队；分级处罚标准和处罚流程如何制订；项目的什么阶段需要风控的介入，是在需求环节提前进行风险评估，运营活动的报

备审批经过风控环节，还是事后发现。建立适合公司自身情况的风控流程和机制，能够提高风控意识，规范流程，对于风控业务运转来说，已经成功了一大半，剩下的才是识别风险。

3.1.2 如何理解业务

一般在其他技术图书或者文章中，对于业务的认知往往是指产品，的确，在互联网圈里，技术的业务方是指产品方。但是在本书中，业务包括线上、线下运营相关的做法及人员，而产品包括线上的网站、App 等资源展现形式，以及背后的策略逻辑和人员（产品经理）。以外卖为例，如何把一个个商家接入平台上营业、如何把用户吸引来下单，这属于**业务**的范畴；接入平台的过程是需要信息化的，需要把很多资料上传到系统里，这属于**产品**功能；吸引用户下单并不断使用这个外卖 App，这属于**产品和运营**策略。从这个例子也可以看出，**业务重线下、重打法，产品重线上、重流程，运营起到衔接作用**。

我们讲风控要理解业务、服务于产品是指，要在弄清楚业务的运转逻辑、搞明白产品的策略和细节的前提下，来制订风控方案，再去影响产品和业务；本质目标是让业务运转更健康和风险可控，不能因为风险控制过度而左右产品和业务发展，当然也需要禁止出现风险明显大于收益的产品和业务形式。

那么如何能很好地理解业务？理解业务就是要熟悉业务的人和事，人的方面是比较容易的，而事的方面需要从很多侧面了解，甚至需要形成一些机制来保障。

1）风控流程和机制保障

主要是指评审流程和应对机制，清楚地规定风控何时、以何种形式介入产品业务中，是在产品评审环节就介入，还是产品上线后运转一段时间、出了问题再介入，以及规定发生风险时与业务协同应对的机制。当然我们鼓励在产品未上线时就能介入，提前评估可能会发生的风险，熟悉数据链路并做好预案。

流程和机制有助于我们快捷方便地了解业务和产品新动向，及时发现漏洞和风险，也是风控执行落地的必备举措。形成了稳定有效的流程和机制之后，能够变被动为主动，随时跟进产品和业务的新动向，而不再是被动去分析研究，同时能够较早介入，提前发现风险。

具体的措施包括在产品评审环节进行风险评估，对报备的运营活动进行风险评估，准备风控 SDK 的接入，明确数据从端到仓库的链路，清晰风险的应对措施和处罚手段，确定好接口人，等等。

2）多用多体验

主要是对风控策略相关人员的要求，需要对业务流程进行完整的梳理，形成业务流程图，对每个环节都能非常清楚其中的业务逻辑。如果业务只是涉及线上环节，有产品

功能对应，还是比较容易梳理清楚的。但如果涉及线下环节，没有线上功能与之对应，那就需要把线下的运转细节也了解清楚，弄明白线下与线上的对接是如何进行的，因为看不见的地方更容易出现问题，数据上的异常也不好解释。

笔者曾在外卖订单的备注里发现这样一种现象：一个商户的订单虽然来自不同的用户，但备注却都是一样的，而且是一串不明含义的短数字。了解后才知道，地面拉新人员在线下拉新时，为了标记每个用户是谁拉来的，让用户在下单时备注一个工作人员的编号，便于后续的统计。

再例如，点外卖订单时，收货地址只能在商家的配送范围内，但总有一些订单的地址出现在配送范围之外，单看线上数据必然会认为是异常。事实上，在线下拉新时，选择的商户与拉新地点可能不在一处，就会修改商户的配送范围。当事后看到订单时，商户的配送范围早已恢复成了原来那样。如果计算发生在事后，那么就会出现配送范围之外的情况。

其实很多企业都有让研发人员到线下体验业务流程的传统，为的也是更好地熟悉和理解业务运转过程。

3）紧跟公司的战略目标

业务一定是紧跟公司战略目标的，熟悉战略目标便于理解很多事情为什么要做，为什么有轻重缓急，如何权衡利弊，也便于对齐目标。另外，建议最好了解一下行业的产业链、竞品的模式和优缺点，你会知道谁更容易“招来”黑产。

4）了解业务发展的历史

这个看似不相关的问题其实很常见，很可能成为工作中的一个坑点。我们知道，采用机器学习建模是需要样本的，样本往往都是历史数据，历史数据经常存在不同于当前的特殊取值，这跟当时的业务背景强相关。

比如有一个特征是订单一次享受的优惠额度，正常取值范围一般在 100 元以内，但实际中还是发现了取值超过 200 元甚至 300 元的例子，而它属于特殊渠道的补贴场景，是历史某个阶段的产物。如果不了解这种特殊情况的存在，很容易影响准确率。实际工作中，因为历史数据的缺失或者区别于当前形势而导致的模型效果差不是小概率事件。

3.1.3 业务理解的认知表现

怎么才叫理解了业务，并没有严格的标准定义。根据笔者经验，可以从以下三方面考虑。

1）清楚业务的目标、考核内容以及成本核算

业务的目标决定了事情的空间，也决定了利益的焦点和矛盾的焦点。若业务的目标

是要扩大市场规模、实现增长，那么短时间内可能会有快速发展，会更关注拉新、新增日活、月活等。而矛盾的地方在于，量的快速增长之下可能埋藏着风险，比如当日新增用户的次留、7 日留很差，这在一定程度上就决定了风控的关注重点。风控往往对接很多条业务线，清楚每个业务的目标，有助于灵活调整方向。反过来说，能够灵活调整风险管控政策，也可以反映出对于业务理解的深度，毕竟始终不变的风控政策是很少的。

了解业务的考核内容，一方面可以更具体地熟悉业务的目标，另一方面便于关注业务人员的做事焦点，这些地方很容易出现违规问题。比如，业务考核“新用户数”一项，那么就需要弄清楚，此处对“新用户”的定义口径。

了解成本与收益的核算方式。这个其实很重要，从业务角度看，隐含了业务打法；从风控角度看，有了利益与风险衡量的基准。公司想要盈利，可以根据用户的长期价值与成本核算，也可以根据单订单的收益与成本核算，黑产也是类似的思路。与外卖为例，对于外卖平台来说，一条订单，不考虑其他费用的情况下，假设涉及的成本是物流配送费用（A）和优惠补贴（B），收益是用户支付价格（C）和商家抽佣（D），那么需要 C+D 大于 A+B 才能有正向收益。而对于刷单作弊者而言，获取的补贴要大于付出的成本才有空间，所以如果能核算出订单有没有刷单收益空间，那么对于判断是否存在刷单嫌疑有很大帮助。

2）知晓业务逻辑和产品细节

前面讲到要多用多体验，才能对业务流程和产品细节非常清楚，清楚的标准是能够非常熟练地讲出主要流程逻辑、产品细节，具体到有哪些输入、哪些输出。比如 App 的某个页面是 H5 的还是 Native 的，一条订单有哪些字段，需要用户填写的有哪些，会做哪些验证，还要注意，像“备注”之类需要用户随意发挥的地方，很容易出现涉敏涉黄暗语等风险。

再比如关于优惠的形式有红包、代金券、会员等，这里面的钱由谁来承担要清楚。因为这才是影响风控策略制订的关键。同样是“满 50 减 20”，减的这 20 元里企业承担多少，商户承担多少，代理商又承担多少，不同比例要区别对待。如果跟普通用户一样不关心这里面的区分，或者默认都是一方承担的，那就说明你没理解透。

再比如，对细节的理解还体现在能够跳出流程本身，洞察其中可能被黑产利用的点。我遇见过这样一个例子，某平台为了推广平台上的超市购，特意补偿用户大额优惠券（比如“满 200 减 150”），一个账号、一部手机、一个身份每天可以购两单，每单都能使用。微信和支付宝为了抢占支付场景，也曾多次通过超市购物进行补偿活动（比如“满 100 最高减 50”），一般是在每次活动期间，一个账号、一部手机、一个身份只能享受一次。不知道读者朋友是否注意到了这两种优惠方式的风险区别。

3）了解业务和产品产出的数据

如果说前面两点还比较泛泛、无法量化衡量的话，那么这一条便是具体要求。用户

端看到的商家数据从何处来？有哪些主要维度？涉及哪些处理流程？用户订单包括哪些字段？这些都要清楚。这些不同数据的产出和维护一般是由不同部门支持的，每份数据的责任方要清楚，数据口径尤其重要，特别是新业务由于不断更新迭代，往往忘记通知使用方数据字段。

根据以上内容，我们可以整理成表 3-1，如果都能很好地回答出来每个条目，那么理解业务这一关就可以算作通过了。

表 3-1　了解业务的内容

	谁	做什么	看什么	有什么	为什么
业务 1					
业务 2					

其中，

- 谁：指哪些业务方，包括业务侧、产品侧和研发侧的接口人。
- 做什么：指业务流程和策略细节。
- 看什么：指业务关注哪些指标，其中哪些与风控相关。
- 有什么：指业务上产生哪些核心数据，尤其是用户数据、设备数据、行为数据和交易数据。
- 为什么：指业务上的一些细节决策为什么要那么做，对数据上会产生哪些影响。

3.1.4　业务理解的行动表现

理解业务不仅要熟悉业务的人和事，还要从业务的角度看问题，不要站在业务的对立面。我们将其总结为几项注意事项，遵循这些事项会让风控更容易被理解、被接受，更好地服务于产品和业务：

1）注重用户体验

风控的目的是控制风险，减少企业损失，但很难灭绝风险。为了惩罚作弊刷单的可疑用户而有损正常用户体验是难免的，但是这个平衡度一定要把握好。关注用户体验主要从下面几个方面着手。

- **管控手段要分级**：不同的风险等级要对应不同的管控，不能一概而论。如图形验证码、短信验证、语音验证、人脸识别、实名验证、禁部分功能、封号等，对用户体验的影响程度不同，需要根据不同风险等级执行。业界一般通过打扰率指标来监控。
- **文案描述避免争议**：对于盗号等影响用户资产的风险，应给出非常明确的提示；

针对用户的分级管控需要有文案提示的，应注意文案的语气和措辞。

- **寻求更好的交互方式**：通过产品化的解决方案来规避风险永远是上策。通过简单的产品逻辑和流程的调整来避免风险，而且不影响用户体验，是最低成本的解决方案，也是最合理的方案。
- **应对投诉要有力**：为客服提供风险分析报告和部分管控权限，便于给用户提供解释或执行封禁 / 解封等操作，缩短问题解决的链条长度。

2）灵活管控、重检测、快响应

行业内存在一个基本的原则：轻管控、重检测、快响应，这里做了稍许调整，将“轻管控”改为“灵活管控”。

轻管控原则是指在判断有风险嫌疑、需要限制用户操作时，限制的动作宜轻不宜重。这里有两个原因，一是从用户体验角度出发的，二则是因为想要在一个路径节点上对风险判断准确是十分困难的事情，特别是前面的节点和中间节点，所以需要全链条风控，当限制发生在路径比较靠前的节点上时，如果发生误判对用户伤害就会很大，因此需要从轻处理，让用户付出较小成本二次验证，走向下一个节点。

轻管控对于实时风控来说是尤其合适的，比如能尽量在页面上通过验证码验证就不要短信验证，能短信验证就不要禁止访问。这背后的逻辑是对策略的置信度和用户风险等级的考虑。

离线场景稍有不同，判断依据的信息量更大、更准确，对于非 To C 的业务来说，如 To B 和内控，则可以从严管控。原因是这部分业务一旦存在风险，数额较大，影响较为恶劣，再加上离线场景下的分析准确率更高，特别是在需要规范秩序和打造口碑的阶段，比如餐饮行业对于营业执照和卫生许可证的要求，随着监管日趋严格，一旦发现商户的证件缺失或造假，应当立即下线。当然，从严处理同样也是基于策略的置信度和风险等级考虑，对准确性要求更高。

无论是从严还是从轻处理，都需要根据业务所处的阶段灵活调整。比如同样是刷销量，在大平台上刷销量和在一个小的电商平台上刷销量，得到的惩罚一定不同；即便同样是在大平台上刷销量，10 年前和现在的惩罚力度也是不同的。

综上，笔者认为把“轻管控”调整为“**灵活管控**”更合适。

重检测强调一方面是要尽可能分析更多的数据，更全面、准确地进行风险判断，另一方面是要检测先行，即做到知彼，对平台的风险以及风险影响的范围和严重程度都能了如指掌。可以不管控，但是不可以不知晓。

快响应原则很好理解，发现问题及时处理，尽可能减少损失。但是说起来容易做起来难，想要做到快速响应，需要风控系统的实时能力支持，需要能够随时清楚业务细节的变化。做风控遇到需要快响应的事件实在太多了，我遇到过因为风控系统无法快速支持、

需要业务团队修改逻辑临时补救的，也遇到过不清楚细节就快速修改了风控策略、导致产生新问题的状况。支持紧急突发事件，要做到快，离不开实时规则引擎能力。

3）注重风险收益平衡

注重用户体验和灵活管控，也都是平衡原则的体现，平衡原则的主要体现是准确率和召回率的取舍。这种取舍与很多因素有关，比如业务发展阶段、风控系统发展阶段，甚至风险发生的规模和时机。

4）注重数据友好

数据友好包括友好的可视化和策略的可解释性。基于大数据的风控模型往往都是黑盒的，如何能将关键数据以可视化的方式直观展现给决策者和运营人员，方便他们使用，这是非常非常重要的。关于模型可解释性的研究当前也有一些可选方案，3.2 节会详细介绍。

3.1.5 数据和模型论

看到做一个合格的风控从业人员需要这么透彻地理解业务，很多读者估计要崩溃了，甚至觉得这是在胡扯：“我只是想做一个建模工程师啊，难道需要这么熟悉业务才行吗？我对业务就是不感兴趣怎么办？”“我没有去熟悉业务，模型效果也很不错啊！”的确，很多风控的算法研发者唯数据和模型论，但是风控不是一堆模型的集合。

如果不去理解业务，到底能否做好风控呢？可以明确告诉大家，不可能。那么理解了业务就一定能做好风控吗？答案也很明确，同样不能。因为风控范围太广了，不单是技术。

但不理解业务就做好一个模型，还是存在可能的。最典型的例子就是一些模型比赛，比赛给出的数据往往都是脱敏的，以 $f1$、$f2$、$f3$ 等形式给出，参赛者可能完全不知道是什么含义、更别提理解业务了，但这不影响建模，参赛者会用到一些技巧寻找有利的特征。例如 Kaggle 的 Avazu Click-Through Rate Prediction 比赛任务里，冠军队伍在数据分析上就找到了一个绝密武器——不采用完整数据集建模，而是把数据集分成两部分单独建模。这其实需要一定的数据分析技巧和数据敏感度。而大部分情况下，都是在做数据探索，比如可能的方法包括但不限于以下这些：

- 观察数据。比如每个变量的分布情况、变量两两之间的相关性、与目标的相关性等。
- 构造统计特征。单纯理解数据最常用的方法便是对原始特征的扩展，比如历史均值、方差、对比值等。
- one-hot 编码。统计每个取值的出现频率，抓住 top 的取值进行 one-hot，剩余的小类别归到其他。

- 向量化。向量化可以把稀疏高维特征转变为稠密低维特征，可以把对象间的复杂关系用数字量化表示，相似的用户或者相关联的用户在向量上也具有相似性，这会减少很多复杂的特征工程工作。
- 模型融合。采用 stacking 和 ensemble 思想融合模型，弥补单个模型的缺陷。

事实上，比赛中大量的数据探索工作，放在实际工作中，就是在熟悉业务和数据的细节。而单纯只看数据不结合业务，太容易走弯路和采坑了。不仅如此，风控的模型还需要形成数据的闭环才能不断地迭代改进，而这些都需要理解业务背景、清楚数据流。

要站在更高层次思考全局解决方案，而不是陷在局部最优解里。当你站在更高层次往下看的时候，会发现模型之于风控可能只是很小一部分，训练数周的模型比不过一条业务规则的作用，这是常有的事情。所以实际工作中，理解业务与使用技术手段相结合，才能够事半功倍。

3.1.6 理解业务的风控实例

本节通过几个小的实例，讲述在产品细节上如何兼顾用户体验、把握风险收益平衡的度。

1）实例一：打车体验

用过某打车 App 的朋友可能会留意到，曾经有个功能，在连续两次取消叫车后需要等待几分钟才可以再次叫车，这个功能大概率是为了防止恶意刷单而进行的频率控制操作。打车需要设置出发地和目的地，出发地可以通过定位默认填充，当然也可以通过用户挪动地图上的位置或手动输入来改变出发地；目的地默认是根据用户出行记录预测的，当然也可以修改，在紧急情况下很容易出现两地点设置错误、点击了打车按钮，只能取消再打。

从数据分析上看，取消两次的比例应该是很少的，因此权衡了体验和风险后，采取了取消两次需要等待的这样一个设计方案。实际上，在兼顾体验上，仍有空间可做，而不是采取所有用户一刀切的方式。例如可以根据对用户的分析建模，对正常用户上调取消的次数限制，对可疑用户采用原来方案，并可以实时根据用户取消的次数来控制可使用取消的机会。

2）实例二：外卖异地下单

订外卖时有一个细节，如果一个常住地在北京的人突然出现在上海下单，那么一般在下单时需要进行短信验证甚至语音验证。这也是从风险角度做出的决策，认为异地下单的账号有刷单可能性。

类似地，这依然有继续优化体验的空间，例如高健康度的用户异地下单时可以免去

验证，以减少对用户的打扰；而可疑用户则采用验证方案。

3）实例三：线上付款码

某线上支付 App 使用付款码的设计上曾经有个功能：对 App 的用户开放扫码收钱能力，且不需要商家扫码设备，用手机扫一扫即可收钱。付款码下方还有一个数字串，数字串等同于付款码，若能获取数字串就可以进行收钱，这样就能套取支付账号绑定的信用卡金额。

这个例子完全照顾了体验，而没有过多考虑风险。当然该 App 很快将付款码做成了动态变化的，并隐藏了起来，且改为了只能商家扫码收款的模式。

这三个例子后来的产品方案都做到了兼顾用户体验和控制风险，虽然有的还有优化空间，但可能受限于实现成本。从这几个例子也可以看出，风控的这个度并不难把握，风控也并不是业务的对立面，而是一道盾牌，有力保护业务的安全运转。

3.2 风控需要被理解 <<<

要做好风控，需要风控从业人员深刻理解业务，也需要业务人员能够理解风控，因为后者涉及风控的执行力。在前几年，除了金融企业，公司内了解风控是做什么的员工比例极低，更谈不上如何默契合作了，更多的是很多质疑：为什么要加这个采集数据的需求？为什么要拦截掉这些用户？为什么好不容易谈来的合作伙伴被风控影响了合作？面对这些问题，怎么才能更好地让业务人员理解风控呢？很可惜这不是一个单点问题，也不单是一个技术问题，需要综合手段治理。比如我们可以：

- 经常以邮件或者卡片等形式向公司其他员工介绍风控团队的职责、分工和对接人，加强教育。
- 邮件通报风控发现和处理的案例。
- 建立风险评估流程，介入业务线和产品的评审研发环节。

以上这些手段可以强化风控的职责存在感，使得外界能够对风控形成像正常业务一样的熟悉度，知道如何配合工作。

本节介绍的“风控需要被理解”的侧重点不在这里，主要是指风控结果的可接受问题，尤其是风控模型结果的可接受问题，这是技术能解决的部分，即可解释性。

风控是集问题发现与处罚于一体的，处罚才能被其他团队感知，而处罚一个对象就要有依据，就好比对一个疑犯定罪，要有证据才可以。没有理由支撑结论，业务团队就会与风控团队之间存在信任鸿沟。尤其是采用机器学习建立的风控模型，单纯一个打分结果是站不住脚的，需要提供“证据”。

3.2.1 模型可解释性

那么怎么定义模型可解释性？Zachary C. Lipton 在 *The Mythos of Model Interpretability* 中认为可解释性要求模型具有透明度，过程可以被模拟。就好比一段代码的流程图，按照流程图的步骤，输入数据可以人工推演出来预期的结果，称之为可模拟性。

笔者认为这个观点更多针对计算过程，而这里讲的风控模型可解释性是要让人能看懂为什么得到这样一个预测结果，来解决模型的信任问题。一个打分结果，如果业务团队看不懂就会抵触，哪怕在总体上有准确率的效果数据摆在那。好的可解释性追求的是让没有技术背景的人也能看明白（interpretation is the process of giving explanations to Human）。这其实是很难的，所以其中会有不同程度、不同角度的解释方法。虽然 Lipton 在文章中也多次批判了信任问题，但实际工作经历告诉我们，它确确实实存在。

关于模型可解性的研究还是一个相对新兴的领域，还没那么系统化，尚没有一套完整的标准，不过这不妨碍我们先学习一下先贤们是怎么做的。无论是哪种定义和方法，都对于理解风控模型是有益的。Christoph Molnar 在他的 *Interpretable Machine Learning, Making Guide for Making Black Box Models Explainable* 一书中，提到了一些对可解释性方法分类的参考标准。

（1）是基于模型内在结构解释，还是事后再做解释。内在解释性要求一些机器学习模型本质上是具可解释性的，比如线性模型、树模型。事后解释意味着先选择和训练一个黑盒模型，在训练后应用可解释性方法，比如通过特征重要性、部分依赖关系图等。

常见的基于模型内在结构解释的方法主要有以下几种：

- 基于规则的方法，规则本身就是可以解释的，所以规则系统在风控领域大面积应用也是理所当然的。
- 经典的线性模型，比如线性回归、逻辑回归、广义线性回归等，这些是应用最广泛且可解释性最高的方法，据说全世界每秒使用量上千万次。
- 基于决策树的方法，决策树本质上是一堆 if-else 规则，同样具有较好的解释性，基于决策树的模型如随机森林、XGBoost，虽然不够直观，但也算是可解释的。
- 传统基于聚类的方法以及知识图谱表示，本身就具备直观解释性。

事后解释方法与模型训练过程是分离的，比如通过显著特征图（Saliency Map）解释 CNN 网络的 CAM 与 Grad-CAM 方法、通过局部建模的代理模型方法 LIME，ICML 2017 年的最佳论文还提出了利用影响函数理解黑盒模型预测结果的方法。

（2）是针对特定模型的解释，还是与模型无关的通用解释。特定于模型的解释方法与模型强相关，取决于每个模型的能力和特征，比如系数、p 值、与回归模型相关的 AIC 分数、决策树的规则等。而与模型无关的解释方法依赖于事后对模型的分析，可用

于大部分机器学习模型。这种方法通过分析特征的输入输出对来逼近模型的预测结果，无须关注模型内部。

（3）是局部解释，还是全局解释。局部解释是针对单个样本的预测结果进行解释，又叫个体解释；全局解释则是针对整个模型行为，也称为总体解释。

从这里也能看到，上述划分并不是正交的，而是有重合的，如 LIME 方法既属于事后解释也属于局部解释方法。接下来就展开介绍一下其中典型的几种解释方法，这里从全局解释和局部解释的角度划分，因为局部解释用到的场景更多，在局部解释中再分别介绍模型相关和模型无关的解释方法。

3.2.2 全局解释

全局解释（即总体解释）着眼于模型本身，关注哪些维度起到了主要作用，因此便于抓住主要因素，忽略次要因素，利于做决策。总体解释在风控模型中多用于特征选择、发现作弊的主要区分点、总结作弊规律、制订策略和优化产品逻辑。

1. 基于特征重要性的总体解释

总体解释的方法当前以特征重要性为主，做过机器学习的读者应该都很熟悉图 3.1 的特征排名，这是根据 xgboost.plot_importance 方法或者 feature_importances_ 属性绘制输出的排名靠前的重要特征。需要说明的是，以下内容中的示例部分基于的是 Kaggle 竞赛题目“Synthetic Financial Datasets For Fraud Detection”提供的数据集训练模型，该数据集是在某移动支付公司的交易日志基础上合成加工出来的。

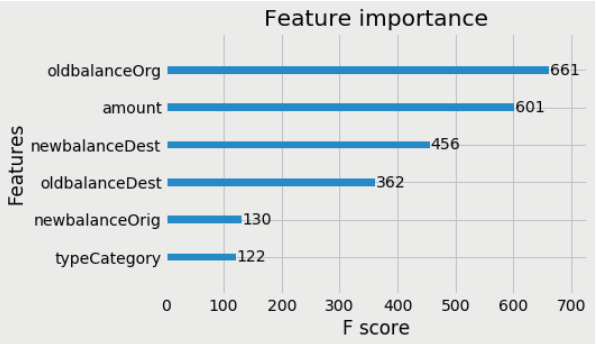


图 3.1　特征重要性排名

在图 3.1 中，柱状图的横轴表示 F 值（F score），纵轴表示特征（Features）；amount 表示交易金额；oldbalanceOrg 表示源账户交易前余额，newbalanceOrig 表示源账户交易后余额；oldbalanceDest 表示目的账户交易前余额，newbalanceDest 表示目的账户交易后余额；typeCategory 表示交易类型，比如转账、提现。图 3.2、图 3.3 中的词含义相同，不再重复解释。

2. 不同模型的总体解释方法

不同模型在总体解释特征重要性的具体实现上方法不同，如 RF 和 GBDT 模型都会采用 Gini 重要性，RF 还可以通过特有的袋外数据错误率计算特征重要性，而 XGBoost 则可以按权重（weight）、增益（gain）和覆盖（cover）三种方式来计算，LightGBM 则可以按划分（split）和增益（gain）两种方式来计算。Gini 重要性即常用的 feature_importances_ 计算原理，值越大代表特征越重要，在 scikit-learn 官方文档中是这样解释的：

> *The importance of a feature is computed as the (normalized) total reduction of the criterion brought by that feature. It is also known as the Gini importance.*

意思是，在树的构建过程中，每个特征都会有按某种标准计算带来的划分增益，它所带来的总增益即为 Gini 重要性。

随机森林 RF 模型因为其特殊性，存在一定比例的数据没有被采样，称之为袋外数据（Out of Bag，简称 OOB），这样在计算一个特征的重要性时就可以借助袋外数据得出每棵树在袋外数据上的错误率，对该特征加入噪声干扰后再次计算袋外数据上的错误率，二者之差表示为该特征对模型预测结果的影响力，所有树上的差值平均就作为该特征的重要程度。

我们再来看一下风控模型中另一个常用算法 XGBoost，如何计算它的特征重要性。在源码 python-package/xgboost/plotting.py 文件里可以发现，XGBoost 的特征重要性计算有三种方式：weight、gain 和 cover，原文的解释是这样的：

> *importance_type : str, default "weight"*
>
> *How the importance is calculated: either "weight", "gain", or "cover"*
>
> ** "weight" is the number of times a feature appears in a tree*
>
> ** "gain" is the average gain of splits which use the feature*
>
> ** "cover" is the average coverage of splits which use the feature*
>
> *where coverage is defined as the number of samples affected by the split*

解释如下。

- weight（权重）是指一个特征出现在树中的次数，也就是被用来分裂节点的次数，图 3.1 就是按权重排序的结果。
- gain（增益）是指使用该特征分裂时的平均增益，即平均的训练损失减少值，图 3.2 是按增益排序的结果。
- cover（覆盖）指的是一个特征被用来分裂时，会影响一定数量的样本数，所有

样本数的总和除以用于分裂的次数，所得的平均值即为覆盖，图 3.3 是按覆盖排序的结果。

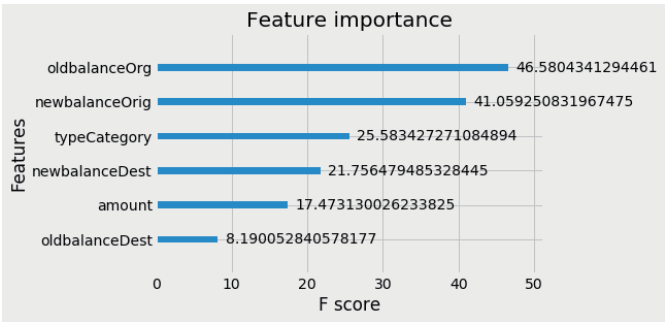


图 3.2　特征重要性 gain 类型

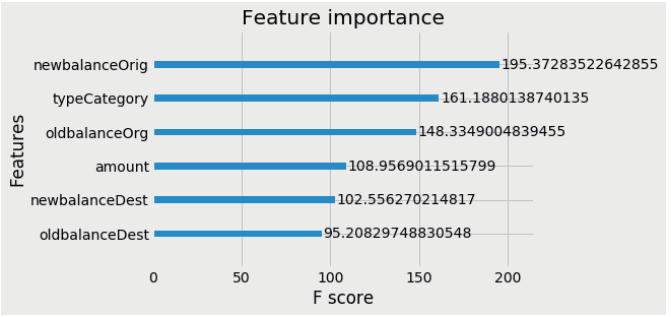


图 3.3　特征重要性 cover 类型

3. 统一解释方法

通过图 3.1~ 图 3.3 可以看出，选择不同的类型，输出的结果是不一样的，那么到底该选择哪个？使用者可能会有很大的疑惑。除此之外，还可以用 Partial Dependence Plot（部分依赖图，简称 PDP）方法来计算特征重要性，如图 3.4 所示展示了 amount 和 oldbalanceOrg 与预测结果的关系，它在固定其他特征不变的情况下，通过改变观察变量的值来看模型结果的变化，从而计算特征重要性的。

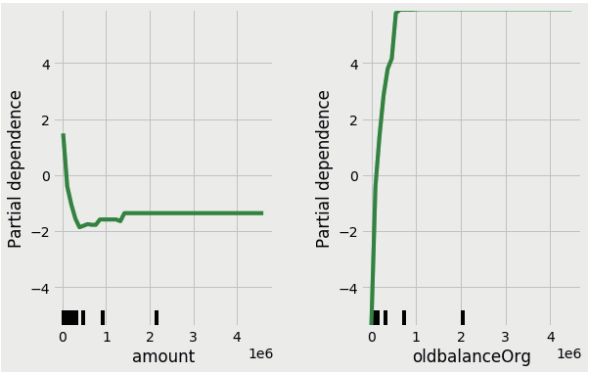


图 3.4　Partial Dependence Plot 方法

有没有满足一致性和精确性的特征重要性解释方法呢？这就是即将介绍的 SHAP 方法，它可以给出两种形式的特征重要性，如图 3.5 和图 3.6 所示。图 3.5 是以柱状图形式给出的特征重要性，与图 3.1~ 图 3.3 类似；图 3.6 则与 PDP 方法有一定相似性，即给出了单个特征对结果影响的正负向，这里先不关心 SHAP 方法的计算原理。

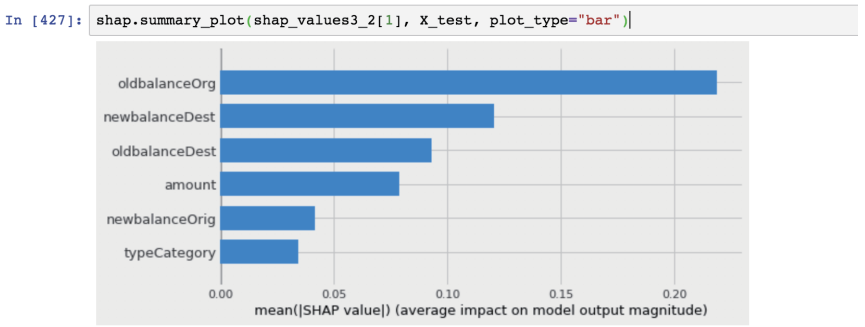
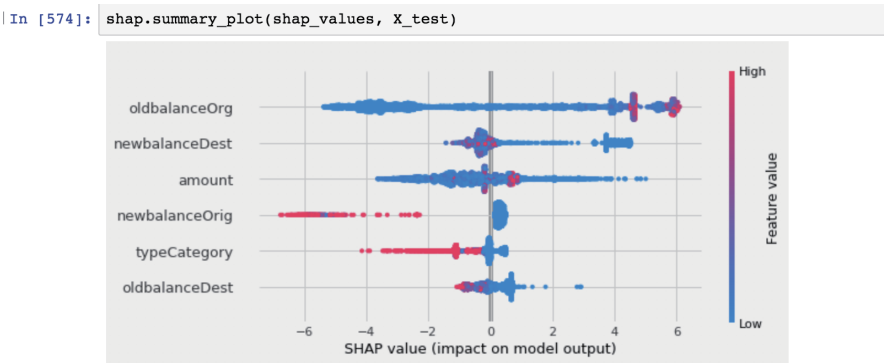


图 3.5　SHAP 全局均值法输出特征重要性（1）



扫码看彩图

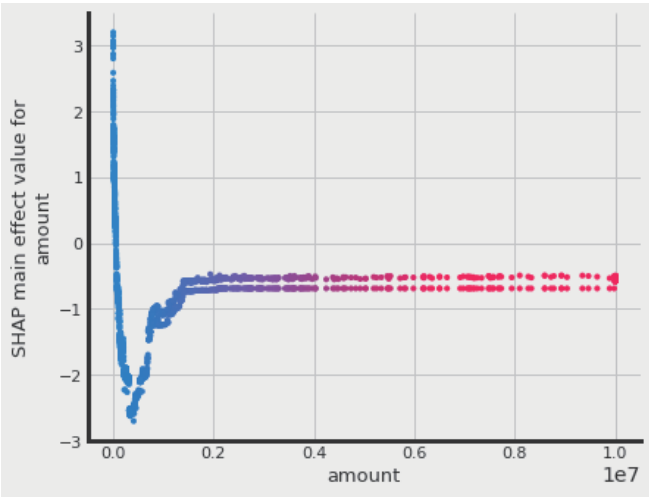
图 3.6　SHAP 全局均值法输出特征重要性（2）

图 3.6 是 SHAP 特有的输出方式，其左侧的特征重要顺序由右侧的 High-Low 方向标示，与图 3.5 一致，是在样本集上的总体重要性排序；每个样本在每个特征对应的行上都有一个点，点的颜色和位于中轴线（横坐标为 0）的位置决定了该特征往哪个方向影响样本类别倾向，及其影响大小。比如特征 oldbalanceOrg 对分类结果的影响最大，覆盖的样本数也最多，随着源账户交易前余额越来越大，风险也在增加；而 oldbalanceDest 影响的样本数就比较少，而且余额大反倒降低风险；newbalanceOrig 特征表明，交易后的源账户余额越小越增加风险，反之降低。

我们还可以像 PDP 那样观察单个特征对结果的影响，如图 3.7 所示，amount 在 0.1（单位为 1e7）以内时，最终的预测结果受其他特征的影响大；而到 0.2 以上时，影响就很稳定，这说明交易金额大小超过 200 万[①]时风险较大，但分出两条线。根据图 3.8

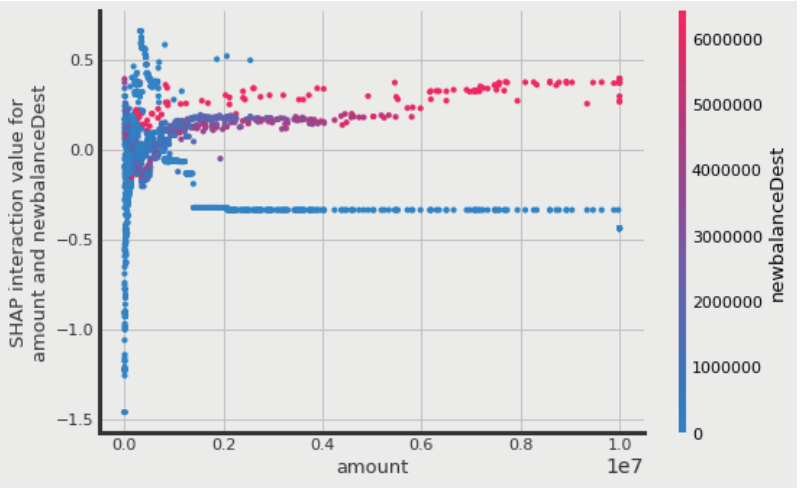
① 数据的实际单位对模型来说没有意义，数据集里已经统一了量纲，在这里只关注数据本身。数据集官方给出的解释是以当地货币单位计。

所示的 newbalanceDest 与 amount 的交叉影响结果看，一部分样本在 amount 大于 0.1 和 newbalanceDest 低于 3 500 000 时，SHAP 值表现稳定，风险较小，即图中的横线部分；而 newbalanceDest 高于 4 000 000 时，SHAP 值也相对稳定，风险较大，这也就可以解释为什么图 3.7 中有两条平行线。这里的 SHAP 值稳定都是指在这种情况下，这两个特征的影响是占主要的，其他不稳定的情况都会受其他特征影响而导致预测结果浮动。



扫码看彩图

图 3.7　amount 特征对结果的影响



扫码看彩图

图 3.8　amount 与 newbalanceOrig 的交叉影响

SHAP 方法也支持多个特征的交叉，我们再来看下排名第一的特征 oldbalanceOrg 与 typeCategory 的交叉结果，如图 3.9 所示，源账户交易前的金额大于 1 千万时，此时 typeCategory 均为 TRANSFER 类型而非 CASH_OUT，对于风险预测的影响是占主要部分的，其实不考虑 typeCategory，在 oldbalanceOrg 大于 300 万时，对模型的影响也是很大的。

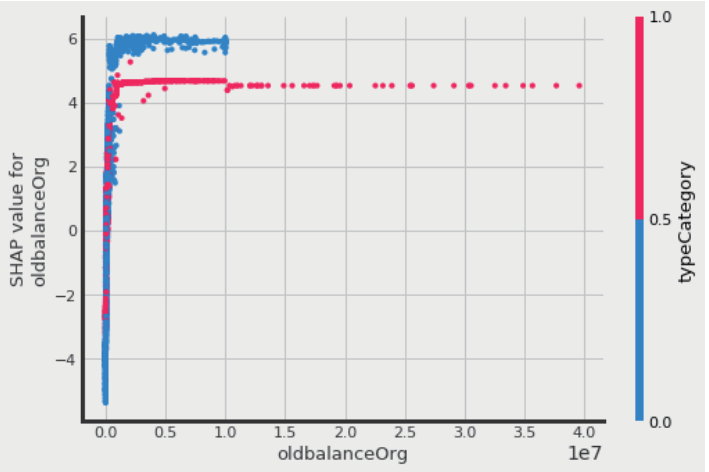


图 3.9　oldbalanceOrg 与 typeCategory 的交叉影响

4. 深度模型的总体解释方法

在接触 SHAP 方法之前，我们知道对于经典的机器学习模型来说，都是可以使用 feature_importances_ 来分析特征重要性的，但对于深度学习模型并没有这种东西，怎么解决呢？ PDP 提供了一种思路，即对每个特征进行随机 shuffle，观察模型指标的变化。这里我们介绍另一种敏感性分析方法——基于方差的敏感性分析（也叫 Sobol 方法），其他敏感性分析方法还有 OAT/OFAT、FAST 方法、Morris 筛选法等，感兴趣的读者可以参考相关文献。

Sobol 方法是一种比较古老的方法，它把模型当作黑盒，因此无关线性和非线性。假设一个模型有两个输入和一个输出，可能会发现 80% 的输出方差是由第一个输入的方差引起的，10% 的输出方差是由第二个输入的方差引起的，5% 的输出方差是由两个输入之间的相互作用引起的，这里的百分比就是敏感性。如何计算出敏感性度量呢？这需要采样来生成自变量，输入模型中获得因变量，然后根据下面的一阶和总阶敏感指数公式计算出来。

$$S_i = \frac{\mathrm{Var}_{X_i}(E_{X_{\sim i}}(Y \mid X_i))}{\mathrm{Var}(Y)}$$

$$S_{Ti} = \frac{E_{X_i}(\mathrm{Var}_{X_i}(Y \mid X_{\sim i}))}{\mathrm{Var}(Y)}$$

其中，$Y = f(X)$，X_i 为第 i 维特征；$X_{\sim i}$ 表示除 X_i 之外的特征集合；S_i 为 X_i 的一阶敏感度，表示 X_i 的影响；S_{Ti} 为 X_i 的总指数。采样的基础方法为伪随机的蒙特卡罗方法，但低差异的准蒙特卡罗方法可以使采样更均匀，敏感度系数计算更准确。不过依然需要较多的采样数据以保证计算精度，而且随着维度增加，需要计算的方差和期望更是呈指数级增加，后来又提出了一些改进方法，比如 Polynomial Chaos 和 GPR（Gaussian Process Regression）方法，感兴趣的读者可以参考相关文献。

根据下面这个例子，我们简单看一下敏感度分析的用法。为了便于直观，我们把用到的特征都显示出来，同时使用 XGBoost 模型替代深度模型，实际中可以把深度模型的

预测结果保存到文件中，然后加载进来。

```
    from SALib.sample import saltelli
    from SALib.analyze import sobol
    cs = ['amount','oldbalanceOrg', 'newbalanceOrig','oldbalanceDes
t','newbalanceDest','typeCategory']
    problem = {
        'num_vars': 6,
    'names': cs,
    #// 此处为各个特征取值的上下界
          'bounds': [[X_undersample['amount'].min(), X_
undersample['amount'].max()],
                  […], […], […], […], […]]
    }
    # 采样 1000 个
    param_values = saltelli.sample(problem, 1000, calc_second_
order=True)
    ### 以下代码可通过 yy = np.loadtxt("deep_module_outputs.txt", float)
替换为深度模型的结果，这里随便拿一个训练好的模型示例
    yy = np.zeros([param_values.shape[0]])
    for i, sx in enumerate(param_values):
        sx2 = {'amount':[sx[0]],'oldbalanceOrg':[sx[1]],'newbalance
Orig':[sx[2]],'oldbalanceDest':[sx[3]],'newbalanceDest':[sx[4]],'t
ypeCategory':[sx[5]]}
        sx3 = DataFrame(sx2,columns=cs)
        ret = clf.predict_proba(sx3)
        yy[i] = ret[0][1]
    ####
    Si = sobol.analyze(problem, yy, print_to_console=False)
    # 输出一阶敏感度指数
    print(Si['S1'])
```

```
    # 输出总指数
    print(Si['ST'])
```

程序的输出结果如下：

```
    [1.99030736e-01 6.31223187e-02 3.15869487e-04 6.92140204e-04
     2.29707415e-03 7.40510412e-01]
    [0.21977177 0.06871204 0.00207501 0.00184471 0.00202167
0.74083361]
```

从总指数看，敏感度从高到低依次为：'typeCategory' → 'amount' → 'oldbalanceOrg' → 'newbalanceDest' → 'oldbalanceDest' → 'newbalanceOrig'。但这里面有一个问题，特征 'typeCategory' 取值只有离散值 0 和 1，但在采样时变成了连续值，我们对采样后的数据进行修正，使得随机一半为 0，一半为 1，再来看总指数变化：

```
    [0.78891953 0.25285969 0.12526614 0.1131204 0.12708109
0.10249429]
```

这次敏感度从高到低变成了：'amount' → 'oldbalanceOrg' → 'newbalanceDest' → 'newbalanceOrig' → 'oldbalanceDest' → 'typeCategory'，而且多实验几次，结果并不稳定。显然，这与 SHAP 方法给出的结果还是有明显不同的，这取决于抽样的量级和抽样数据是否与原始数据集的分布一致有关。其实，SHAP 方法不单单可以应用于树模型，还可以应用于深度模型，它提供了 DeepExplainer 解释器。对于深度学习的整体性解释，我们还是首推 SHAP 方法。

5. 总体解释的局限性

总体解释在特征选择和模型的训练调优阶段有很大的辅助作用，当给出的特征重要性与经验大相径庭时，模型可能存在错误。更重要的是，它会从宏观上告诉你影响某个事件的主要因素是什么，可以据此做一些业务决策。

但用来解释个体则行不通，根据我与业务人员近几年的沟通发现，业务人员关注的往往都是一个个具体案例，而不是整体结论。为什么这么说呢？比如审核，他要把待审的案例一个个看一遍，每一个情况都不完全一样，如果你告诉他某个用户被判定有欺诈风险，理由是图 3.1~ 图 3.3 的那几个重要性特征，他肯定不会接受。他需要的是这个用户被判断为有风险的具体原因，而不是整体上如何。所以说，针对个体的解释是很有必要的，业务人员是要把事情落地的，是最接地气的，他们不需要也不关心你采用的是深度学习方法还是规则方法。

那怎么来解释个体呢？这里面又可以分为模型相关的解释和模型无关的解释方法。

3.2.3 模型相关的解释方法

我们知道，线性模型是最容易解释的，因为模型是满足可加性的，如下式，所有特征值与其系数乘积的和，即为最终的结果或变体。

$$f(x)=\beta_0+\beta_1x_1+\beta_2x_2+\cdots+\beta_nx_n$$

在做个体解释时，需要以 β_ix_i 作为特征的贡献，而不能以系数的绝对值来解释个体上单特征的重要性。但很多问题不是线性可表达的，而树模型的结构在表达复杂性和可解释性上满足了想象力，其本身的结构和基于信息理论的构树过程，使得其解释起来符合判断逻辑。问题在于当树的深度超过 3 层时，就很难做出可被接受的解释描述了，更不用提 RF、GBDT、XGBoost 等模型深度超过 3 层又有多棵树的情况了。

为此，这些模型在解释个体时都有一些特定于模型的方法，比如 RF 可以采用基于样本分布变化的特征贡献度方法。这是论文“Interpreting Random Forest Classification Models Using a Feature Contribution Method”提出的一种方法，下面通过论文中的一个具体例子来解释一下计算过程。图 3.10 所示为使用 RF 模型训练的树结构，样本集一共 10 个样本，4 个特征，LD 表示每个节点的训练样本集合，Y^n_{mean} 表示每个节点的 LD 中正样本的比例，LI^c_f 表示每个节点在分裂特征 f 作用下的正样本比例增量。

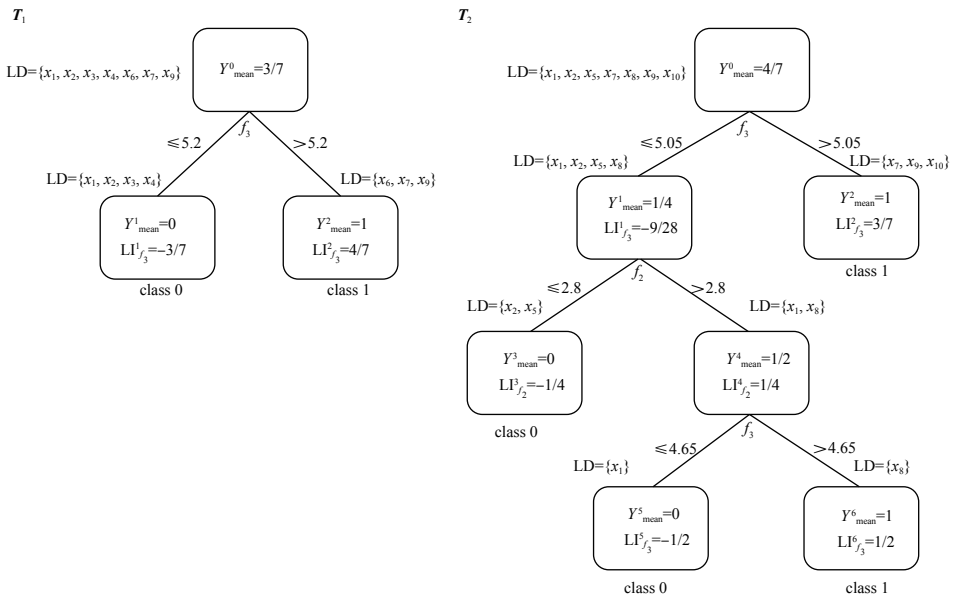


图 3.10　基于样本分布变化的特征贡献度解释 RF 模型示例

对于实例 x_1，如果要计算特征 f_3 对其最终预测结果的贡献度，可以这样计算：$\text{FC}^{f_3}_{x_1}=\frac{1}{2}\left(-\frac{3}{7}-\frac{9}{28}-\frac{1}{2}\right)=-0.625$。$x_1$ 在两棵树中都有出现，在第一棵树 T_1 中的路径是

$n_0 \to n_1$，在第二棵树 T_2 中的路径是 $n_0 \to n_1 \to n_4 \to n_5$。在 T_1 中，从根节点到叶节点的 $n_0 \to n_1$ 这条路径，经过 f_3 分裂后，节点中的正样本比例从 $\frac{3}{7}$ 变成了 0，因此增量为 $-\frac{3}{7}$；在 T_2 中，从根节点到叶节点的 $n_0 \to n_1 \to n_4 \to n_5$ 这条路径，两次经过特征 f_3 分裂，节点中的正样本比例分别从 $\frac{4}{7}$ 变成了 $\frac{1}{4}$、从 $\frac{1}{2}$ 变成了 0，增量分别为 $-\frac{9}{28}$ 和 $-\frac{1}{2}$。因为两棵树是独立的，最终将所有增量求和后取平均值即得出上述结果。

从这个计算过程不难看出，方法的核心要点是计算每个节点的正样本比例，以及从父节点到子节点的正样本比例变化的增量；然后罗列出要解释的实例在所有树中的路径；对路径上的每个参与分裂的特征累加其增量，作为总贡献度；最后求取在整个森林里的平均值，作为特征的贡献度。

GBDT 模型也有类似的解决思路，但 GBDT 与 RF 有很大不同，一是树之间并不是独立的，而是有前后关系的；二是对分类问题而言，GBDT 的输出结果是一个分数而不是类别，论文“Unpack Local Model Interpretation for GBDT”给出了特有的解决方式。由于只有叶节点才有分数，即样本经过特征分裂落到叶节点时获得的分值，只要能把这个分数回溯到根节点，就可以像前面那样计算增量。

我们仍以一个具体例子来解释这个过程，如图 3.11 所示，左图中的节点 6 经过特征“feat5 是否小于或等于 1.5”进行分裂，若是，则落到节点 11，并获得 0.085 的分值；否则落入节点 12，获得 0.069 的分值；以两个叶节点分值的平均值作为父节点 6 的分值估计，即 $\frac{1}{2}(0.085+0.069)=0.0771$（见图 3.11 右图节点 6），并以此进行向上回溯，回溯所有的中间节点，直至根节点。这样每个节点有了分值之后就等同于有了前述方案的 Y^n_{mean}，然后就可以计算增量了，其余步骤都是类似的，稍有不同的是在计算单个特征对一个实例的最终贡献度时，不需要像随机森林模型那样最后再取平均，而是把所有树上的贡献度加和即可。由于从父节点分裂到左右两个子节点的样本数并不一定相同，所以在求父节点的平均值时，可以使用样本数加权的方式。

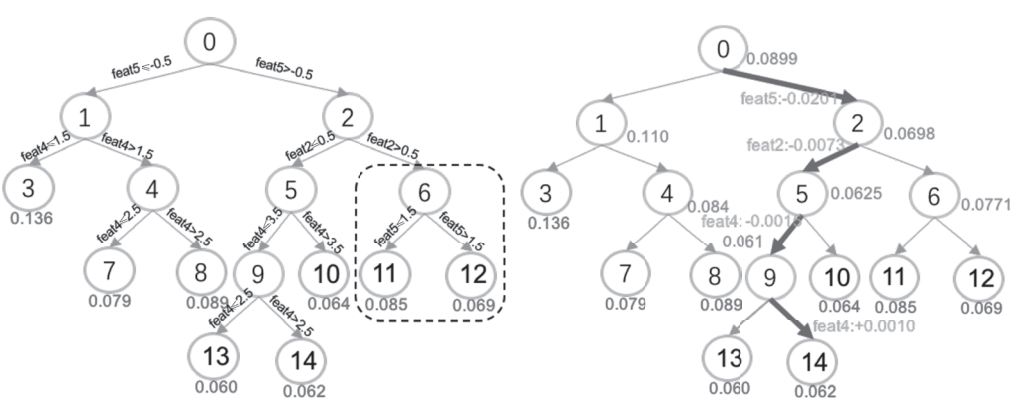


图 3.11　GBDT 模型计算特征贡献度示例

在深度学习方面，也有很多跟模型相关的解释方法，比如针对CNN模型的CAM（Class Activation Mapping）和Grad-CAM（Gradient-based CAM）解释方法，针对RNN模型的FSA（Finite State Automaton）方法。其实对于采用FSA方法来解释时序模型，如果想要在风控领域应用且要面向非技术人员的话，还是不太能够被接受。而对于图像类的识别，如果只是分类问题，很多风控的场景不需要解释，因为一看到图片就明白了。这时的诉求与解释模型的内部决策过程还不一样，但对这种认知的捕捉，为借助指标的可视化呈现来辅助理解提供了可行性，这也是为什么单独写了可视化的内容（第8章）来辅助增加可解释性的原因之一。

CAM是由论文“Learning Deep Features for Discriminative Localization”提出的，我们知道，对于深层卷积神经网络，多次卷积和池化后，最后一层卷积层包含丰富的空间信息，而CAM方法则在最后一层卷积层之后利用GAP（Global Average Pooling）方法重新计算并替换掉全连接层。GAP针对卷积层输出的特征图（feature map）计算每个的均值，再通过加权求和得到输出，如图3.12所示，对每一个类别C，每个特征图 k 的均值都有一个对应的权重 w_k^C。

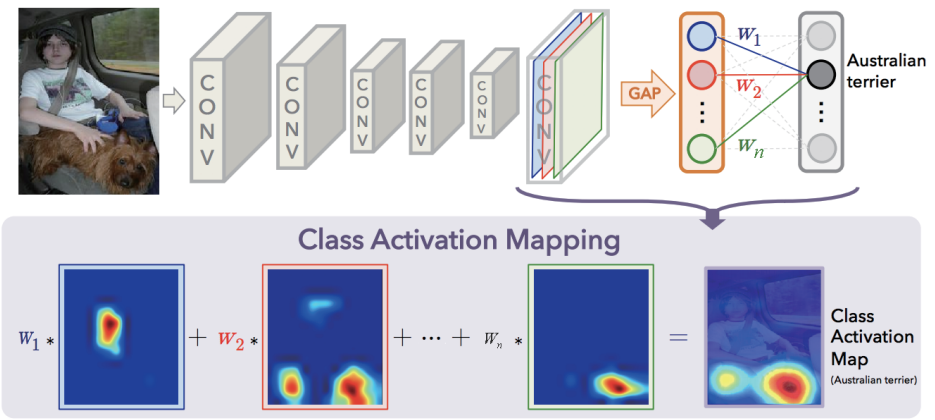


图3.12　CAM过程示意图

这个方法的另一个要点是如何可视化。比如图3.12，要解释为什么分类的结果包含狗，要先把狗这个类别对应的所有 w_k^C 取出来，求出对应特征图的加权和。GAP的好处就是池化的大小与整个特征图是一样的，即在计算均值时是求每张特征图所有像素的均值，这里的加权结果也和特征图是一致的，然后对它进行上采样直到原始输入大小，叠加到原图上去就可以看到可视化效果。

CAM的效果其实是不错的，但因为要把全连接层替换为GAP层，模型要重新训练，所以代价较高，而Grad-CAM便是解决这个问题的。Grad-CAM求权重的方法不同，它是用梯度的全局平均来计算权重，并在论文“Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization”中给出了数学推导，证明了与CAM方法得到的结果是等价的。Grad-CAM还对最终的加权和做了一个ReLU操作，只关注对类别C

有正向影响的像素点，避免带入一些属于其他类别的像素，从而影响解释的效果。

如果每种模型的解释都需要非常清楚模型的结构，那么成本无疑是非常高的，我们当然希望能够用一些通用的方法来解释尽可能多的模型，这就是模型无关的解释方法。

3.2.4 模型无关的解释方法

这里介绍两个模型无关的解释方法，一个是 LIME，属于代理模型解释法，采用一个新的简单模型来拟合原来黑盒模型的局部结构，比如采用线性模型，用模型的系数权重来解释原模型；另一个是 SHAP，基于原模型的条件期望响应来计算特征贡献度。这是目前应用比较广泛的两种解释方法，一些其他的工具如 iml（R 语言）、Live（R 语言）、breakDown（R 语言）、ELI5、Skater 等也多与这两种方法有相关性或可比性。

1. LIME 方法

1）LIME 方法的原理

LIME（Local Interpretable Model-Agnostic Explanations，模型无关的局部解释器）由 2016 年的 KDD 论文“‘Why Should I Trust You?’Explaining the Predictions of Any Classifier”提出，认为用作解释的表征与原始特征可以不同，但在局部逼近。它把模型的原始特征映射到一个简单二进制向量空间，通过在局部抽样建立新的数据集，然后观察原模型预测结果，重新学习一个新的简单模型专门用来解释。如下公式：

$$\xi(x)=\underset{g\in G}{\operatorname{argmin}}\,\mathcal{L}(f,g,\pi_x)+\Omega(g)$$

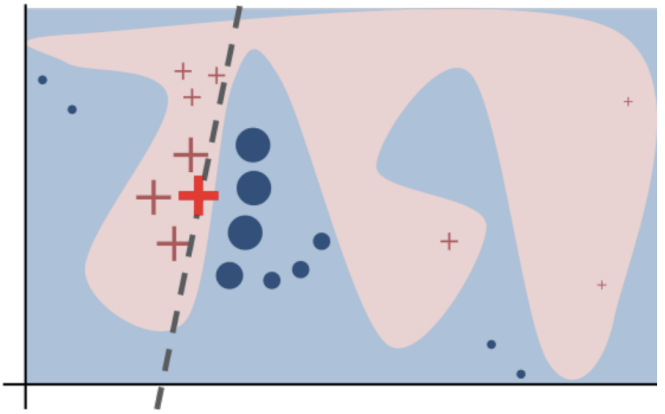
其中，

- 定义解释模型 $g\in G$，G 表示所有可能的解释模型集合。
- $\mathcal{L}$ 等同于损失函数，或者理解为解释模型与原始模型在局部的偏离程度。
- $\Omega(g)$ 表示 g 的复杂度，实际上 LIME 只优化损失部分，复杂度由使用者关注，比如控制使用的特征数 K。
- f 为原模型，$f:\mathbb{R}^d\to\mathbb{R}$，$x\in\mathbb{R}^d$，$x'\in\{0,1\}^{d'}$。
- $\pi_x(z)$ 用来度量 x 和 z 的相似性，即刻画 x 的局部。

文章中以稀疏线性解释为例，采用一个指数核函数实现 $\pi_x(z)=\exp(-D(x,z)^2/\sigma^2)$，$D$ 可以为余弦距离等度量方式；σ 为核函数宽度，越大表示距离远的抽样样本影响越大。

局部抽样的大体过程如图 3.13 所示，假设原始待解释模型的决策边界用蓝粉背景表示，显然是非线性的。图 3.13 中最显著的加号表示被解释的样本（称为 X）。LIME 方法会在 X 周围采样，也会在远离 X 的地方采样，按照它们到 X 的距离赋予权重。用原始

模型预测这些扰动过的样本，用这个新的样本集学习出一个线性模型，并假设其决策边界为图中的虚线。新模型在X附近可以很好地拟合原模型，所以这个解释只在X附近成立，对全局无效，这也是名字中 Local 的由来。



扫码看彩图

图 3.13　LIME 示意图

不过阅读 LIME 的 Python 实现发现，实际由一个变量（sample_around_instance 变量）决定采样方式，如果为 True，以 X 为中心，服从 (0,1) 正态分布抽样，否则，以整体训练数据的均值为中心抽样，但它的默认值却是 False。这也难怪 Christoph Molnar 在 *Interpretable Machine Learning* 中吐槽，LIME 并不是以被解释的实例为中心抽样，而是以训练数据的均值为中心抽样。

对于文本类或者图像类的问题，局部抽样过程明显不同，文本类的数据是通过随机去掉字词来抽样，以 1 和 0 分别表示字词的出现与否；图像类的数据则是通过超像素块分割或者关闭超像素块来采样，因为单个像素点的影响很小，所以按照临近相同颜色的像素块为单位抽样。

LIME 方法虽然缺乏理论支撑，也无法在高度非线性模型中有较好效果，但它对于大部分模型能够给出较为直观的解释。更为重要的是，LIME 方法提出的**通过扰动实例建模**和**用损失函数衡量差异**的思路具有很强的参考意义，掀起了模型解释性研究的热潮。后来很多可解释性方面的研究都与 LIME 有关，比如 Skater 方法，以及下面即将介绍的 SHAP 方法。

2）LIME 方法的使用

LIME 方法的使用是非常方便的，可以下载 Python 工具包，执行 pip install lime 即可安装。下面根据论文中提到的基于新闻数据集（20 newsgroups）来区分 atheism 和 christian 的例子，看看官网给出的具体用法参考。例子中的分类模型采用了随机森林算法，训练有 500 棵树的森林，测试集上准确率可以达到 92%，可以说非常高了。

```
    import lime
    import sklearn
    import numpy as np
    import sklearn.ensemble
    import sklearn.metrics
    from __future__ import print_function
    from sklearn.datasets import fetch_20newsgroups
    categories = ['alt.atheism', 'soc.religion.christian']
    newsgroups_train = fetch_20newsgroups(subset='train',
categories=categories)
    newsgroups_test = fetch_20newsgroups(subset='test',
categories=categories)
    class_names = ['atheism', 'christian']
    vectorizer = sklearn.feature_extraction.text.TfidfVectorizer(low
ercase=False)
    train_vectors = vectorizer.fit_transform(newsgroups_train.data)
    test_vectors = vectorizer.transform(newsgroups_test.data)
    rf = sklearn.ensemble.RandomForestClassifier(n_estimators=500)
    rf.fit(train_vectors, newsgroups_train.target)

    pred = rf.predict(test_vectors)
    sklearn.metrics.f1_score(newsgroups_test.target, pred,
average='binary')

    from lime import lime_text
    from sklearn.pipeline import make_pipeline
    c = make_pipeline(vectorizer, rf)
    print(c.predict_proba([newsgroups_test.data[0]]))

    from lime.lime_text import LimeTextExplainer
```

```
    explainer = LimeTextExplainer(class_names=class_names)


    idx = 83


    exp = explainer.explain_instance(newsgroups_test.data[idx],
c.predict_proba, num_features=6)
    print('Document id: %d' % idx)
    print('Probability(christian) =', c.predict_proba([newsgroups_
test.data[idx]])[0,1])
    print('True class: %s' % class_names[newsgroups_test.
target[idx]])


    exp.as_list()


    print('Original prediction:', rf.predict_proba(test_
vectors[idx])[0,1])
    tmp = test_vectors[idx].copy()
    tmp[0,vectorizer.vocabulary_['Posting']] = 0
    tmp[0,vectorizer.vocabulary_['Host']] = 0
    print('Prediction removing some features:', rf.predict_
proba(tmp)[0,1])
    print('Difference:', rf.predict_proba(tmp)[0,1] - rf.predict_
proba(test_vectors[idx])[0,1])


    %matplotlib inline
    fig = exp.as_pyplot_figure()


    #exp.show_in_notebook(text=False)
    #exp.save_to_file('/tmp/oi.html')
    exp.show_in_notebook(text=True)
```

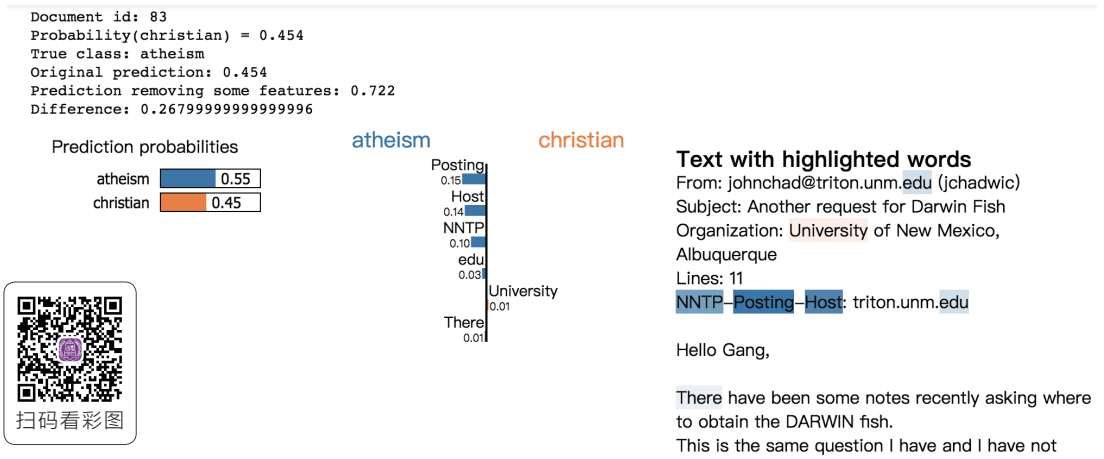


图 3.14　LIME 方法解释示例 1

这里主要是看一下 LIME 的用法，案例的更多细节可以参考官方网站。从示例代码可以看出，前半部分与正常建模并无区别，后半部分使用解释器时，需要传入待解释的个体、回调方法以及特征数（只用少数的特征即可），回调方法的作用是输出类别标签。这是一个只采用了 K=6 个特征的线性模型解释器，从给出的解释可以看出，之所以被分为 atheism 一类，图 3.14 有背景色的几个词起了很大作用，通过去掉影响最大的两个词，预测结果就会偏向对立方，而偏移的概率大小正是这两个维度的影响权重之和，这也从侧面反映了模型是线性的。但是实际上，去掉这两个词，划分为另一类是错误的，虽然原模型准确率很高，但是从这里可以看出它并不是很合理，这也说明 LIME 方法能够辅助发现建模合理性。

再来看一下如何用 LIME 方法解释 3.2.2 节的基于 Synthetic Financial Datasets For Fraud Detection 数据集训练的 XGBoost 分类模型。图 3.15 是用 LIME 解释一个具体样本的可视化示例，左边是模型预测的结果，右边是特征与取值的情况，中间是特征对每个类别的重要性，这里的排序是 oldbalanceOrig > newbalanceOrig > typeCategory > amount > oldbalanceDest > newbalanceDest，与前面 plot_importance 给出的总体重要性结果是不同的。这里给出的是针对这个具体样本的解释，单看 oldbalanceOrig 和 amount 可能有风险，但是考虑到其他特征，综合起来便没有风险。

```
    explainer = lime.lime_tabular.LimeTabularExplainer(X_train.
values,\ feature_names=X_train.columns.values.tolist(), class_
names=c_names, discretize_continuous=True)
    fn = X_train.columns.values.tolist()
    predict_fn = lambda x: clf.predict_proba(pd.DataFrame
(x,columns=fn))
```

```
    exp = explainer.explain_instance(d1.values, predict_fn, num_
features=6, top_labels=1)
    exp.show_in_notebook(show_table=True, show_all=False)
```

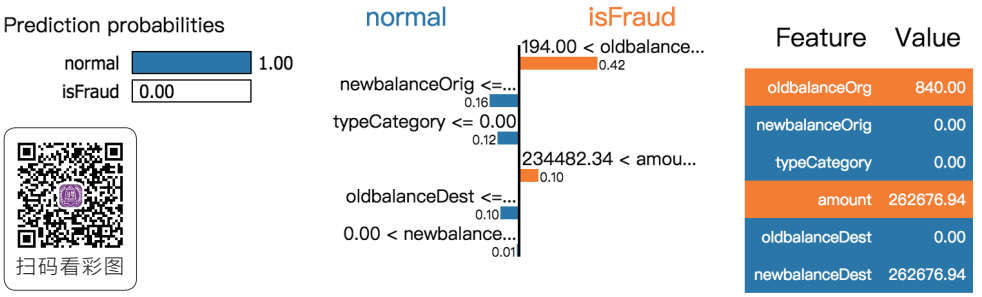


图 3.15　LIME 方法解释示例 2

从以上两个例子可以看到，LIME 方法的可视化效果是比较直观的，可以较为方便地看到每个个体被识别为相应类别的概率，以及对应的原因。2017 年 LIME 作者又提出了新的研究成果 Anchors，它与 LIME 不同的地方在于，LIME 是在局部建立一个可以理解的模型，而 Anchors 则是建立一套更精准的规则。虽然 LIME 论文中提到在各种模型和场景的数据上都表现良好，但笔者更建议在文本方面使用 LIME 方法。LIME 模型在实际应用中可能会存在不稳定现象，由于采用抽样的方法和依赖核函数的原因，原本两个非常相近的样本在解释结果上可能会出现较大差别。笔者更倾向于使用 LIME 解释图像、文本相关的模型，对于结构化的表格数据（tabular 数据），还是建议使用下面的 SHAP 方法。

当然除 LIME 外，图像方面的解释方法还有 Grad-CAM（Selvaraju et. al. 2017）、Loss Landscape（Li et. al. 2017）、Tree Regularization（Mike Wu, et. al. 2017）等，有兴趣的读者可以参考相应的文献。

2. SHAP 方法

1）SHAP 方法的原理

SHAP（SHapley Additive exPlantions）方法目前已经融合到 XGBoost 模型里，可以在源代码 python-package/xgboost/core.py 文件中看到 predict 方法的定义，其中就有 SHAP 相关内容：

def predict(self, data, output_margin=False, ntree_limit=0, pred_leaf=False,pred_contribs=False, approx_contribs=False,pred_interactions=False,

validate_features=True)

我们重点关注后面几个参数的注释，翻译为中文如下：

- 当 pred_leaf=True 时，会输出每个样本在所有树中的叶子节点，可以通过可视化每棵树看到一个样本在树中的决策路径，而叶子节点上的值累加求和后，再经过模型参数 objective 指定的函数进行转换，就得到最终的预测值。
- 当 pred_contribs=True 时，输出所有特征对于一个样本预测值的贡献度，即 SHAP 值，而所有贡献度的累加和等同于预测值。
- 当 approx_contribs=True 时，输出 pred_contribs 的近似版本，时效性上要比 pred_contribs 好一些。
- 当 pred_interactions=True 时，输出两两特征组合的 SHAP 值。

这里面提到的 SHAP 值是一个什么东西呢？在了解 SHAP 之前，我们先来看看什么是 Shapley Value。Shapley Value 其实是来自游戏理论中的一种价值分配算法，该算法由诺贝尔经济学奖获得者 Lloyd Stowell Shapley 发明，因此以其名字命名。

Shapley Value 要解决的问题是多人合作的价值分配问题（参考自 *A Course in Game Theory*，读者可参考了解细节）。记全集 $N=\{x_1,x_2,\cdots,x_n\}$ 有 n 个元素，代表 n 个人；任意 s 个元素组成的子集 $S\subseteq N$，称为一个合作联盟；v 为价值函数，$v(S)$ 表示子集 S 中所有元素共同合作产生的价值。那么最终每个人分配的价值表示为 $\varphi_i(N,v)$，此即 Shapley Value。价值分配满足下面几个原则：

- 有效性：每个人分得的价值之和等于 $v(N)$，即 $\sum_{i\in N}\varphi_i(N,v)=v(N)$。
- 对称性：如果 i 和 j 是可互换的，那么 $\varphi_i(N,v)=\varphi_j(N,v)$；对于任一不包含 i 和 j 的集合 S，都有 $v(S\cup\{i\})=v(S\cup\{j\})$。
- 冗员性：如果 i 未做贡献，那么 $\varphi_i(N,v)=0$，$\varphi_i(S)=v(S\cup\{i\})-v(S)=0$。
- 可加性：如果说把一个游戏分成两部分，那么获得的价值也可以拆分为两部分，即对于任何两项任务 v 和 w，$\varphi_i(N,v+w)=\varphi_i(N,v)+\varphi_i(N,w)$。

Shapley Value 的计算公式为：

$$\varphi_i(N,v)=\sum_{S\subseteq N\setminus\{i\}}P(S)\Delta_i(S)=\frac{1}{|N|!}\sum_{S\subseteq N\setminus\{i\}}|S|!(|N|-|S|-1)!\{v(S\cup\{i\})-v(S)\}$$

$$P(S)=\frac{|S|!(|N|-|S|-1)!}{|N|!}$$

$$\Delta_i(S)=v(S\cup\{i\})-v(S)$$

对于该公式的直观解释，可以认为 Shapley Value 就是求解 x_i 对每个参与的合作联盟的边际贡献期望值（边际贡献即 $\Delta_i(S)$），如果把所有元素的顺序看作排列组合，而所有排列中位于 x_i 前面的排列总共有 $|S|!(|N|-|S|-1)!$ 个。

为了便于理解，以一个具体例子演示一下计算过程。假设一个开发团队由 L、M、N 三人组成，目标是要开发一个 100 行代码的机器学习模型，三人必须一起才能完成该项目。合作的价值贡献如表 3-2 所示。

表 3-2　合作的价值贡献

合作联盟	代码行数
L	10
M	30
N	5
L,M	50
L,N	40
M,N	35
L,M,N	100

不同序列的边际贡献如表 3-3 所示。

表 3-3　边际贡献

序列	L 贡献	M 贡献	N 贡献
L,M,N	v(L)=10	v(L,M)−v(L)=50−10=40	v(L,M,N)−v(L,M)=100−0=50
L,N,M	v(L)=10	v(L,M,N)−v(L,N)=100−40=60	v(L,N)−v(L)=40−10=30
M,L,N	v(L,M)−v(M)=50−30=20	v(M)=30	v(L,M,N)−v(L,M)=100−50=50
M,N,L	v(L,M,N)−v(M,N)=100−35=65	v(M)=30	v(M,N)−v(M)=35−30=5
N,L,M	v(L,N)−v(L)=40−5=35	v(L,M,N)−v(L,N)=100−40=60	v(N)=5
N,M,L	v(L,M,N)−v(M,N)=100−35=65	v(M,N)−v(N)=35−5=30	v(N)=5

三个参与者共有 3!=6 种序列组合，那么按照公式，每个开发人员的 Shapley Value 为：

贡献者	计算公式	Shapley Value
L	1*(10+10+20+65+35+65)/6	34.17
M	1*(40+60+30+30+60+30)/6	41.7
N	1*(50+30+50+5+5+5)/6	24.17

那 Shapley Value 如何与机器学习关联起来呢？把所有特征视作游戏的参与者，训练模型进行预测便视为一场游戏，而样本 i 之特征 j 的 Shapley Value，即 φ_{ij} 的含义是：特征值 x_{ij} 对样本 x 的预测结果相对于整个数据集的平均预测结果的贡献度（或者叫偏离度，所以存在负值）。图 3.16 为 Christoph Molnar 在 *Interpretable Machine Learning* 中所举的例子。

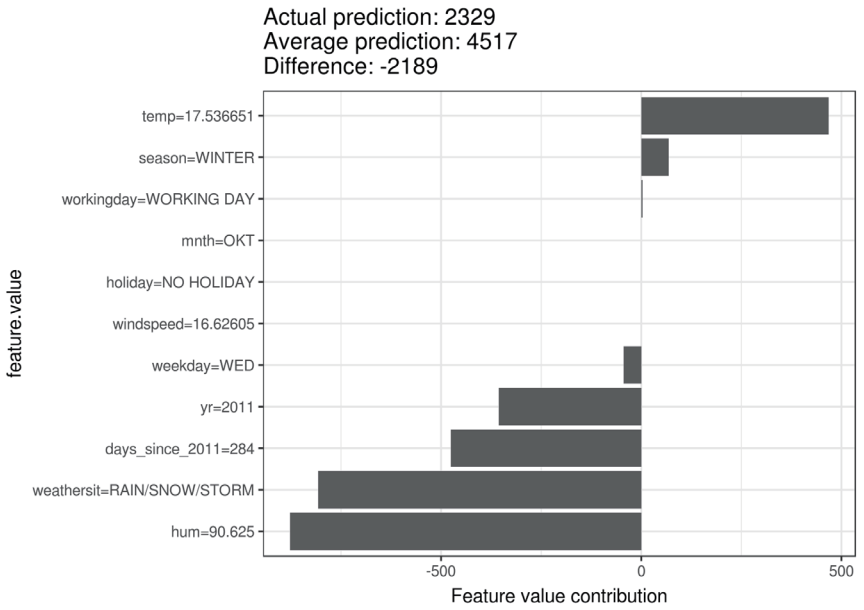


图 3.16　特征贡献示例图

该例子采用 RF 模型，利用天气和日期等信息预测每天的自行车租赁数量，图中横轴表示特征值对应的贡献度，纵轴为每个特征与特征值，自上而下分别表示温度（temp）、季节（season）、工作日（workingday）、月份（mnth）、假期（holiday）、风速（windspeed）、周末（weekday）、年份（yr）、时间（days_since_2011）、天气（weathersit）和湿度（hum）。针对图中的一个样本，实际预测值（Actual prediction）是 2329，当天的平均预测值（Average prediction）是 4517，比平均值小 2189。从图中可以看出，最有负向影响的是湿度（hum）、天气（weathersit）和时间（days_since_2011），而当天的温度是正向的影响。所有特征的贡献度加和等于 −2189，即一个样本的所有特征的贡献度加和，为该样本预测值与平均值的差：

$$\sum_{j=1}^{p}\varphi_{ij} = f(x_i) - E_X(f(X))$$

那么，每个特征的 Shapley Value 到底如何计算呢？根据前面的公式，要计算一个特征的 Shapley Value，需要计算出有该特征参与的一个集合的预测值，以及对应的只缺少该特征的集合的预测值，得出其边际贡献，然后重复计算，直到遍历了所有该特征参与的集合。最后所有的边际贡献的平均值即为该特征的 Shapley Value。这将是一个指数级的计算复杂度，因此 Strumbelj et al. 在 *Explaining Prediction Models and Individual Predictions with Feature Contributions* 中提出了一种近似的计算方法：

$$\varphi_{ij} = \frac{1}{M}\sum_{m=1}^{M}(f(x^{*+j}) - f(x^{*-j}))$$

其中，$f(x^{*+j})$ 是样本 x_i 的预测值，但其中一部分特征值被随机样本的特征值所取代，

x^{*-j} 与 x^{*+j} 类似，区别是少了 x_{ij}，换成了随机样本对应的特征值。这 M 个样本都是经过这种替换模式、由两个样本拼凑而成的。近似算法的计算过程如下：

step1：For all $j \in \{1,2,\cdots,p\}$：

step2：For all $m \in \{1,2,\cdots,M\}$：

step3：从数据集 X 中随机选择一个实例 z

step4：选择特征的一个随机排列 $o \in \pi(S)$

step5：实例 x：$x_o = (x_{o1},\cdots,x_{oj},\cdots,x_{op})$

step6：实例 z：$z_o = (z_{o1},\cdots,z_{oj},\cdots,z_{op})$

step7：构造新的实例：

$$x^{*+j} = (x_{o1},\cdots,x_{oj-1},x_{oj},z_{oj+1}\cdots,z_{op})$$

$$x^{*-j} = (x_{o1},\cdots,x_{oj-1},z_{oj},z_{oj+1}\cdots,z_{op})$$

step8：计算 $\varphi_{ij}^{(m)} = f(x^{*+j}) - f(x^{*-j})$

step9：计算 $\varphi_{ij}(x) = \frac{1}{M}\sum_{i=1}^{M}\varphi_{ij}^{(m)}$

根据算法描述，对于样本实例 i 和特征 j，$j \in \{1,2,\cdots,p\}$，固定抽样次数 M；每次从数据集中选择一个样本 z，样本的特征值顺序做一定扰动，并生成两个新的按该顺序排序的样本实例 x^{*+j} 和 x^{*-j}，前者由实例 i 的 j 个特征值与样本 z 的 p-j 个特征值组成；后者由实例 i 的 j–1 个特征值与样本 z 的 p–j+1 个特征值组成，相较于前者而言，仅仅换掉了特征 j。然后计算 M 个 x^{*+j} 和 x^{*-j} 的模型结果之差的均值，作为特征 j 对样本 i 的 Shapley Value。

了解了 Shapley Value，再来解密 SHAP。SHAP 是 SHapley Additive exPlantions 的缩写，由华盛顿大学的 Scott M. Lundberg 和 Su-In Lee 在论文“A Unified Approach to Interpreting Model Predictions”中提出，是一种可以解释任何模型的通用归因方法。而 Shapley Value 正是 SHAP 方法中的重要技术基础，SHAP value 的本质就是 Shapley Value，但它与原模型的边际期望有关。SHAP 被认为是同时满足 Local accuracy（局部准确性）、Missingness（缺失零贡献）和 Consistency（一致性）三个属性的加性特征归因方法唯一解。

（1）局部准确性是指解释模型 $g(x')$ 在局部等同于原始模型 $f(x)$，式子表示如下：

$$f(x) = g(x') = \varphi_0 + \sum_{i=1}^{M}\varphi_i x_i'$$

其中，$g(x')$ 为解释模型，x' 是简化的输入，可以通过映射函数 $x=h_x(x')$ 映射到原始输入。$\varphi_0=f(h_x(0))$ 表示没有任何简化输入时的模型输出，即偏置项。

（2）缺失零贡献是指某项特征缺失时，它的贡献度为 0，式子表示如下：

$$x_i'=0\Rightarrow\varphi_i=0$$

其中，x_i' 是第 i 个特征项，φ_i 是该特征项的贡献度。

（3）一致性是指，如果一个特征在一个模型中的缺失所导致的变化，大过其在另一个模型中缺失所导致的变化，那么该特征在第一个模型中的贡献也要大于其在另一个模型中的贡献，式子表示如下：

$$f_x'(z')-f_x'(z'\setminus i)\geqslant f_x(z')-f_x(z'\setminus i)\Rightarrow\varphi_i(f',x)\geqslant\varphi_i(f,x)$$

其中，f' 和 f 为两个模型，$z'\setminus i$ 表示 $z_i'=0$。

理论证明，Shapley Value 是满足三个条件的加性特征归因方法唯一解。因为 SHAP 基于 Shapley Value 做解释，自然也就是满足条件的唯一解。SHAP 定义每个特征的贡献为该特征在给定条件下（因此排序的集合也是给定的）模型结果的期望：$f_x(z')=f(h_x(z'))=E\left[f(z)\mid z_S\right]$，其中 S 是 z'（$z'\subseteq x'$）中非 0 下标的集合，z_S 为其中一种排序，$h_x(z')=z_S$。

SHAP 的作者曾在论文中指出，用 $E\left[f(z)\mid z_S\right]$ 来近似表示 $f(z_S)$ 主要是考虑大部分模型无法处理缺失值。这种定义形式，基本对齐了 Shapley regression（Stan Lipovetsky, et. al. 2001）、Shapley sampling（Erik Štrumbelj, et. al. 2014）等方法，同时还与 LIME、DeepLIFT（Avanti Shrikumar, et. al. 2017）等方法有一定联系。如图 3.17 所示，以一种顺序为例（实际上 φ_i 应该为所有排列顺序的平均值），SHAP 值解释了当前输出 $f(x)$ 是怎样偏离基准值 $E[f(x)]$ 的，其中 φ_0、φ_1、φ_2、φ_3 使得 $f(x)$ 往高于基准值方向走，而 φ_4 使其更接近基准值。

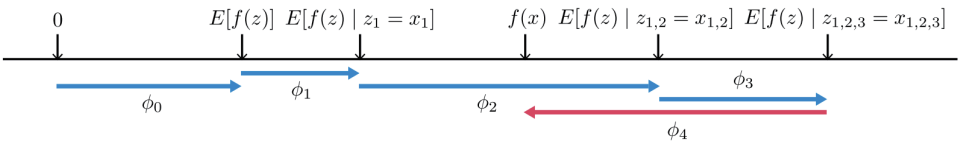


图 3.17　SHAP 的模型表示

鉴于精准计算 SHAP 值的计算量高（因为 Shapley Value 的计算代价高），SHAP 根据 LIME、DeepLIFT 等方法，又结合 Shapley Value 算法进行近似计算，并针对不同的模型提供了不同的解释器，如下：

- TreeExplainer：Tree SHAP 实现的解释器，适用于树模型如 XGBoost、LightGBM 等，速度快，准确度高。

- DeepExplainer：DEEP SHAP 实现的用于深度学习模型的解释器，基于 DeepLIFT 和 Shapley Value 算法，适用于 TensorFlow 和 Keras 框架上的深度模型，速度快，但只能做近似解释。
- GradientExplainer：也是应用于深度模型的解释器，不过是基于 SHAP 和集成的梯度算法，性能上比 DeepExplainer 差一些。
- KernelExplainer：Kernel SHAP 实现的解释器，适用于任何模型，主要方法是 LIME 和 Shapley Value。

也就是说，SHAP 方法集成了多种技术，并在其基础上改进优化，使得满足三条性质，包括 LIME、Shapley Value、DeepLIFT 等，所以 SHAP 又是一种综合框架，从其论文中前半部分对各种方法的抽象也能看出。至于 LIME 方法的推导如何与 Shapley Value 建立联系，感兴趣的读者可以参考论文。

传统的 Shapley Value 计算忽略了特征之间的相互作用，后来又提出了改进版支持特征交叉组合。传统的 SHAP 也未考虑特征组合问题，后来又对二阶特征组合进行了研究。目前最新的 XGBoost API 也支持了这种功能，即 pred_interactions 参数。

2）SHAP 方法的使用

如果感觉前面的原理介绍还是比较难懂，没关系，这并不影响我们的使用，下面通过一个具体的例子来看一下如何做解释。

```
import shap


# load JS visualization code to notebook
shap.initjs()
# explain the model's predictions using SHAP values
# (same syntax works for LightGBM, CatBoost, and scikit-learn models)
explainer = shap.TreeExplainer(clf)
shap_values = explainer.shap_values(X_test)
# visualize the first prediction's explanation
shap.force_plot(explainer.expected_value, shap_values[i], X_
test.iloc[i,:], link='logit')
```

使用是不是非常简单？不过对非技术人员来说，其可视化输出并不是特别直观，后面还会在第 8 章介绍可视化的相关方法。如图 3.18 所示，预测为欺诈类的概率约为 0，

在使用 SHAP 解释时，所有特征重要性排序为 oldbalanceOrg > amount > oldbalanceDest > newbalanceDest > newbalanceOrig > typeCategory。同 LIME 一样，不用太关注这个顺序与总体重要性排名是否相符，重要的是通过 SHAP 的可视化结果，可以看出来哪些特征对于判断这个具体的实例是否有欺诈风险有正向作用，比如 oldbalanceDest 和 newbalanceOrig 对判断风险有正向作用，但是其他特征起负向作用，综合起来认为是一个无风险的实例。

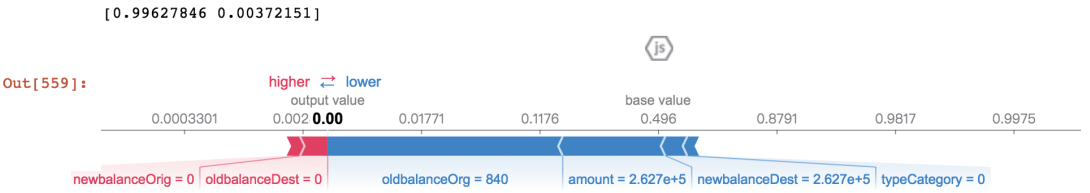


图 3.18　SHAP 可视化

以上内容就是对常用的两种模型无关解释方法的介绍。近年来，关于模型无关解释方法的研究也逐渐多了起来，其中还有一类方法是借助实例来做解释。比如反例方法（Counterfactual Explanation，参考论文“Counterfactual Explanations without Opening the Black Box: Automated Decisions and the GDPR”）。如果用在风控中，可以用一些已经确定事实的各种类型的作弊欺诈例子作为参考，反向套用公式，以特征的距离和模型预测结果的差异来衡量是否可用作解释。

还有一些基于实例的方法，比如 MMD-Critic 方法、贝叶斯实例模型（Bayesian Case Model，BCM）等，通过一些代表性的样本来解释聚类 / 分类结果。这类方法比较有探索性，可以辅助我们更好地了解数据的分布。MMD-Critic 方法能够帮助我们找到数据中一些具有代表性和特例的样本，是 2016 年提出的一种无监督学习方法，它的主要作者是谷歌大脑的一位科学家 Been Kim，主要思想是根据人的认知过程，通过一些有代表性的原型例子来做分类和决策，但这些例子毕竟有限，且不能反映一个事物的诸多方面，所以需要通过一些特例来完善认知。Critic 一词可以理解为“挑刺”，即发现不同于代表性群体的个例。

关于什么是好的解释，并没有一个严格的标准，不同的学者对可解释性的研究重点也有所不同。从使用角度看，当用户受众是产品运营人员时，我们更侧重让结果有更多的数据支撑，而不过分关注模型的决策原理，毕竟模型本身还存在准确性的问题，为错误的结果做解释只对研发调优模型有作用。而想要做到有数据支撑，就容易很多，借助图表将特征和原始指标可视化呈现出来，人们凭借经验和对信息的捕捉能力，结合上述模型的结果，便能够快速形成判断。

3.3 引导型风控 <<<

多年的经验发现，**理解业务**和**被业务理解**并不容易，模型的可解释性解决了“为什么要干掉某个用户”的问题，但很多时候你会遇到另外一种情况：识别准确率很高，理由很充分，却无法改善平台的长期作弊问题，刹不住虚假流量之风。这既是一个理解业务的问题，又是一个被理解的问题，需要双方就“什么样的作弊问题需要打击”达成一致。什么样的作弊需要打击，这并不是一个非常明确的问题，尤其是真人真机参与的作弊。

引导型风控是笔者自定义的一个概念，旨在灵活管控，引导黑产向着高成本方向走，引导业务向着低风险方向走。下面通过几个例子来理解其中的含义。

1）案例：获客

用户增长是任何产品的必经之路，然而获客越来越难，成本越来越高。成本高了便有人便盯上这块肥肉，这里面存在一条暗黑的获客产业链。无论线上获客还是线下获客，都存在大量的作弊，尤其是线上的渠道买量、广告买量。

我们知道，如果是按安装、激活、注册、次留不同深度的指标来衡量流量水分，渠道想要获得稳定收益，势必要达标才行。假如业务按激活付费，风控就要按照注册、次留、七日留等更深度指标去量化风险，并让渠道知晓这种评估方式，以此拉高作弊成本，缩小作弊虚假流量与真实流量的差距，当虚假流量在深度行为上足够逼真时，就不再适合当作虚假流量对待，相应的管控手段也可以有所区别。例如，业务按激活付费拉新时，某渠道带来的用户中，一部分用户在激活、注册、次留甚至七日留存上都表现良好，然而这批用户每天安装 App 数量超过几十个，可谓专业作弊者，试问此时如何对待这批用户？这时风控的目标就非常重要，如果当成作弊者，对于黑产来说，久而久之并不能形成一个通过提高深度行为来规避作弊的认知，这对于引导流量往提高深度转化效果方向发展并不利，对长期的反作弊也不利。

2）案例：推荐

针对一线销售业务的风控，一以贯之的方法是用各种规则和规章限制、约束其行为。如果约束过于宽松，则效果有限；如果约束过于严格，则会引发一线的反弹，不够灵活应对业务形势，严重的甚至会出现销售聚众上访闹事现象。

根据笔者多年的领悟，在处理与一线销售接触的事情上，尤其是如果想要算法在一线业务落地应用，要么自上而下地强推，要么以一种推荐的思路去做。而风控方面，这两种思路可以兼具：有些红线性质的行为，必须明确规章约束；而对于有探讨空间的业务打法，因线上算法难以全面获取线下数据，准确性难以保障，可以在相对有把握的领域以推荐的方式辅助一线，逐步改变销售习惯，以减少风险。

例如，外卖业务的销售可以跟商家合作在地面进行拉新活动，在商家门口或者远离商家的繁华地段进行活动宣传，吸引新用户当场下单。为了不影响商家正常的接单和配

送派单工作，出现了一种“虚拟店”，即在系统中复制原商家，除配送范围外，其他所有信息与原商家一模一样，但仅用于拉新，用户下单不需要走配送派单环节。很多漏洞往往都是因为临时方案的善后工作没做好导致的，这种“虚拟店”也是如此——没有完善的销毁管理手段，甚至需要借助研发写脚本手动删除。这给流动性很大的销售人员带来了可乘之机，他们便可以利用“虚拟店”进行虚假拉新和劣质拉新，进而这种现象又影响了正常使用“虚拟店”拉新的销售，线上数据看到的表象是拉新质量越来越差，甚至有作假成分。

要解决这个问题，初步方案便是一开始讲的通过一系列规则加以约束，但是不易找到合适的临界点，因为这种“虚拟店”思路本就是填不满的无底洞，在此基础上的任何修修补补都不能起到很好的改善作用，还要为此制订复杂的管理流程、付出监控成本。那么换用推荐思路该怎么做呢？即推荐适合用作拉新的店铺，甚至还可以基于大数据推荐适合拉新的地点、范围和菜品。如此一来，拉新采用推荐的店铺，风险更可控，转化率也会更高。

类似的理念还有很多。例如，在借贷业务中，根据还款能力预估模型推荐匹配的额度，这种思路一方面利于提升转化率，另一方面也是在变相控制风险。又如，基于推荐的信息流广告和搜索广告在风险上也是不一样的，后者可以通过关键词明确定位靶子，而前者则是系统推荐的，黑产无法锁定目标。再如，短视频中的公域流量和私域流量在作弊风险上也存在很大区别，都是因为推荐的方法让黑产很难锁定目标。

3）案例：刷销量、好评

刷销量、刷好评是在很多电商平台都存在的现象，但笔者认为关于这种作弊的识别并不是最重要的，关于它的定位和处罚才是更重要的。销量、好评是商户在搜索排名机制下的一种竞争手段，有刷量需求的商户破坏了这种竞争的公平性和平台的口碑，但它有生存的强需求。所以针对这种类型的作弊，平台要把它封掉下线吗？并不会。一方面要去识别刷销量和刷好评的商户，并尽可能删除这些数据，因为这会影响正常用户的购物体验和平台口碑；另一方面要设计更稳健的排名算法，弱化单纯的销量、好评对排名的权重影响，并加强刷量带来的负面影响。

4）小结

从这几个例子可以看出，引导型风控是基于对业务的理解，从风控角度提出的一种适合良性循环的理念。一方面是让黑产有“目标”，往深度行为进化，虽然更难识别，但更接近真流量，需要灵活管控，避免“嫉恶如仇”和“有异常就干掉”的粗暴手段；另一方面是推荐业务上的降维打法，从业务的根本源头上减少风险的发生。

理解业务路漫长，绝不仅仅是知道业务流程和产品交互。对目标和定位要有深刻理解，清楚风控要解决什么问题，了解风控手段对业务的影响，以及业务反过来对风控的影响。这说起来简单，一以贯之地做起来很难。