

FDE用AI重构电信网络数据分析

电信行业

网络数据分析

AI Agent

数据分析自动化

生成式AI

Simon

AI Native Consulting Manager at top Tech Consulting Firm、InnoBaker 专家顾问

以全局视野解构场景，用复合认知驾驭AI落地。



案例简介

对于大型电信运营商来说，网络数据分析是一项高频且复杂的工作。过去这类工作往往需要一到数名数据分析人员花费数小时甚至数天完成，而且随着数据和问题不断变化，结果的稳定性和准确性也很难始终得到保障。

本案例中，Simon把客户真实的数据分析流程拆开，重新设计输入、处理和输出，把AI做成一个能够稳定运行、反复使用、可以进入生产环境的Agent系统，让原本按天计算的数据分析工作缩短到分钟或小时级，同时保留人工对结果质量的确认环节。

适用人群

- 电信运营商及网络行业从业者
- 数据分析、数据科学、企业数字化团队
- FDE、AI产品经理、企业AI应用负责人

Q1: 作为FDE，你应用AI的具体场景是什么？在这个场景下遇到的具体问题长什么样？

我做的这个案例，客户是一家美国前三大的电信运营商。我们面对的是他们非常日常的一类工作：分析网络数据，回答网络到底发生了什么。

比如，一个很典型的问题就是：现在到底哪个基站坏了？为什么会坏？它的影响有多大？

客户手里有大量网络数据，包括基站、网络等基础数据。过去，这些数据主要还是由数据分析师、数据科学家人工处理。他们可能需要自己写SQL，甚至进一步建立机器学习模型，从数据里寻找问题，再把分析结果整理出来，给经理或者管理层看。

这件事情真正麻烦的地方，在于分析这件事本身太依赖人工，而且问题永远在变化。

一方面，它非常耗时间。一个新的分析需求出来之后，可能需要一两个人，甚至两三个人一起讨论、写SQL、分析数据，最后才能得到结果。

另一方面，它也存在人为错误。SQL一旦写到上百行甚至几百行，中间就可能出现问题。我们在后来的项目中，就发现过一些看起来正确、实际上取错了数据的情况。

还有一个很重要的问题，是传统仪表盘很难真正满足这种需求。

因为领导或者分析师拿到一组数据之后，很自然会继续问：“那如果我把这批人筛出来呢？”“如果换一个条件呢？”“这组数据到底意味着什么？”

仪表盘更适合回答预先定义好的问题，但现实中的数据分析往往是不断追问的。客户真正需要的，其实是一种可以围绕数据持续追问的交互式分析方式，一个静态报表远远不够。

所以从FDE的角度来看，我看到的问题在于：客户已经有数据、有分析师，也有很多AI工具，但这些东西还没有真正组合成一个可靠的业务流程。

这也是为什么我会认为，Agent好不好做不是关键。真正难的是，你能不能把它做成一个业务真正敢用、愿意持续用的东西。

Q2：在你参与之前，这个问题过去是怎么解决的？

平心而论，过去这套人工分析的办法是走得通的，也不能简单说传统方法“不好”。

最核心的方式就是人工分析。分析师拿到数据以后，自己写SQL、跑模型、出报告。对于相对固定的问题，这种方式其实很有效，而且分析师能够结合自己的业务经验判断结果。

后来客户也尝试过一些自动化方式。比如，他们尝试通过仪表盘自动生成报告，希望减少人工工作；也尝试过直接用一些Agent框架去解决问题。甚至在我们介入之前，他们自己已经搭过一个Agent。

但问题很快就暴露出来了。“能跑起来”和“可以生产使用”，是完全两回事。

客户自己做的Agent，可能就是给模型一个提示词：“你现在是一个数据分析师，我给你这些数据，请帮我分析。”

看起来非常合理，但真正跑起来以后，结果往往并不令人满意。即使底层用的是很好的模型，也不能保证最终输出符合业务要求。

更麻烦的是，有些方案把问题归结成“大模型本来就是不确定性的，所以答案不一样很正常”。

我并不认同这种说法。如果我是一个企业老板，我问一个Agent两遍“我们公司哪个班的成绩最好”，我希望它在数据没有变化的情况下告诉我同一个人。模型本身具有随机性，不代表我们做出来的业务系统就必须是不稳定的。

所以我们真正需要解决的，是怎么设计一个系统，让模型能够在业务流程里被可靠地使用。

Q3：后来你使用AI解决这个问题的整体思路 and 关键步骤是什么？

我做这个项目时，先做的第一件事是搞清楚三个问题：客户到底要什么输入？最后需要什么输出？中间到底发生了什么？

因为如果这三个东西没有搞清楚，Agent写得再漂亮也没有意义。

第一步，重新梳理原来的分析流程。

我们把原来的数据分析过程拆成比较细的不同组件，再重新考虑每一个环节应该使用什么方法、什么结构，最后再把这些组件重新组合起来，让整个Agent的输出尽可能达到客户要求的结构和质量。

第二步，把Agent定位成一个可以反复运行的业务系统。

我们没有把它做成一个“会聊天的分析师”，而是把它当作一套业务系统来设计。

比如说，客户问一个网络问题，系统需要能够理解问题，然后处理对应的网络数据，再进行分析，最后给出结构化结果。

这中间每一步都需要考虑：数据是不是对的？分析过程是不是可靠？输出格式是不是符合要求？如果同一个问题重复执行，结果是不是稳定？

所以我们最终交付的，从客户角度看可以理解成“一个Agent”；但从我们的角度，它实际上是一个可以被反复使用的Agent系统。

第三步，考虑如何让它可扩展。

客户往往有几十甚至上百个分析师。如果每个人都单独装一个Agent、单独维护一套东西，这个方案很快就会失控。

所以我们还需要考虑如何把Agent放到云端，让它能够被统一控制、监督和使用，同时让不同用户共同产生的数据和经验能够反过来提升系统，包括记忆、语义层等能力。

第四步，尽量不改变客户原来的工作习惯。

我并不认为AI落地一定意味着员工必须重新学习一套复杂的系统。

如果架构做得好，它完全可以嵌入原来的使用方式。客户原来怎么使用，就继续怎么使用。可以通过网页应用，也可以通过Teams、Slack，或者其他已有的框架去调用。

在这个案例里，我们最后让客户继续以聊天的方式使用它。这其实是我理解的FDE很重要的一点：想办法让AI进入客户已经存在的工作流。

Q4：相比传统方式，使用AI后的实际效果如何？

变化首先体现在时间尺度上。

以前一个新的数据分析问题出来以后，可能需要一个人甚至两三个人一起工作，而且是按天计算。

现在很多分析可以在分钟到小时级别内跑完，先得到一份比较完整的数据分析结果，再由人工做最后的质量确认。

我觉得这里不能简单理解成“AI替代了几个分析师”。

真正的价值在于，原来大量时间被消耗在重复的数据处理和分析过程中，现在这些工作可以先由Agent完成。分析师不需要从零开始，可以站在AI已经完成的结果上继续判断。

同时，AI系统还可以减少人工分析中一些容易出现的错误。

当然，我们并没有把人工审核彻底拿掉。因为企业真正需要的，是把人的时间从大量重复劳动中释放出来，集中到更需要判断的地方。

另外一个效果，是交互方式发生了变化。

过去仪表盘更像是“我提前把问题定义好，然后给你看答案”；现在则可以围绕数据继续追问。这对于电信网络这种数据量大、问题变化快的业务来说，其实比单纯生成一张报表更有价值。

而且从架构角度，我们交付的是一个可以长期运转的Agent系统，所以它的价值在于逐渐进入客户日常的数据分析工作。

Q5：在部署AI过程中，是否遇到了新问题或挑战？你是怎么解决的？

这个项目里最大的挑战，恰恰是很多AI演示不会暴露出来的东西：稳定性、准确性、成本，以及规模化。

很多人做Agent，第一阶段很容易成功。你给它一个问题，它能回答；换一个问题，它也能回答。于是大家会觉得这个东西已经做好了。

但一旦进入生产环境，问题就来了。比如，同一个问题反复问，答案是不是一致？数据分析过程有没有出现错误？SQL有没有写错？如果用户数量从一个人变成几十上百个人，成本还能不能接受？

这些问题，才是真正决定一个Agent能不能落地的地方。

第一个挑战，是稳定性与准确性。

稳定性尤其明显。前面说过，很多人把模型随机性当成“结果不稳定很正常”的借口。

但我们的做法，是通过系统设计，把这种随机性放进一个相对可控的业务流程里。就像汽油本身不稳定，但汽车工程的意义，就是让汽油可以安全、稳定地驱动车辆。

第二个挑战，是成本。

成本是一个很现实的问题，包含多个层面，“调用模型多少钱”只是其中之一。

如果一个团队不知道怎么做Agent，可能为了调试一个东西花几千美元；但一个真正有经验的人，可能用十几、二十美元级别的成本就能完成同样的调试。

而进入实际使用以后，还有长期运行成本。如果架构设计得不好，就像一辆漏油的车，你根本不用谈省油，因为油一直在往下掉。

第三个挑战，是对客户业务的理解。

如果没有真正理解客户的业务问题，你可能一开始方向就错了。再加上如果对整个AI生态、数据能力、云能力以及软件工程缺乏理解，就容易出现“手里只有一把锤子，所以什么问题都拿锤子解决”的情况。

所以我一直认为，真正能够做好这类工作的FDE，不应该只是懂某一个模型或者某一个工具的人。

Q6：根据你的经验，我们怎样才能更好地把AI部署到实际业务中？

关于怎么把AI部署到实际业务中，结合这次项目，我想强调几点。

第一，理解客户。

客户到底在解决什么问题？真正的痛点是什么？哪些东西值得自动化？哪些东西不应该自动化？如果这些问题没有搞清楚，后面的技术选择都没有意义。

第二，不能只懂一个AI工具。

我经常会说，如果你只知道自行车，那么你看到一个很远的地方，就只能想办法骑自行车过去。但如果你同时了解汽车、飞机以及其他交通工具，你才能真正判断什么场景应该用什么工具。

对于企业AI也是一样。你需要同时理解AI、数据、云，以及基本的软件工程能力。尤其是当你面对的是一个具体业务问题时，你不能因为“我最熟悉这个工具”，就强行把问

题往这个工具上套。

第三，不要把“能跑”当成“能落地”。

现在做一个Agent并不难，难的是把它做成生产就绪。

它要稳定，要可重复，要控制成本，还要能够进入客户原来的工作流。如果客户有几十、上百个用户，还要考虑怎么统一管理、监督和迭代。

所以FDE的价值，在“把工具部署进去”之外。

如果是比较严格意义上的FDE，它既需要有扎实的技术能力，也需要对客户那些没有被明确表达出来的需求保持敏感。技术能力和软能力，两方面缺一不可。

最后，不要迷信FDE这个名词。

现在国内把FDE这个概念讲得很宽，从AI咨询到方案设计，再到真正的部署，都有人称自己为FDE。但无论叫什么，最终还是要回到一件事情：你有没有真正用过这些工具，有没有在自己的场景里解决过问题。

以前Excel也有类似的“客户成功”或者技术支持角色，有人教你怎么用VBA、怎么把复杂功能用起来。今天换成了Claude Code、Codex、Agent等新工具，本质上并没有发生特别神秘的变化。

真正的区别，只是工具越来越复杂，而一个普通用户往往只会使用它最表层的功能。

一个真正懂这些工具的人，价值就在于把那些普通用户不知道的“下一层、再下一层”能力带进实际工作里。

所以，用一句话来概括这次经验，我会说：FDE的使命，是把AI真正变成企业正在使用的工作方式。