# 我希望我们在开始第一个机器学习项目之前就知道的事情



任何新事物都会带来许多我们随着时间而发现的未知。在花了一段时间使用任何新技术之后，我们有了一个清单，上面列出了我们希望能够寄回我们的旧自我的东西。这是我希望在开始构建我们的第一个机器学习管道之前，我可以发送回我们团队的列表之一。有了这个可以节省我们很多时间，希望通过分享这个可以节省你的时间。

正如我在我的一篇文章中所说的，我们正在与Apache Spark合作。因此，本文将包含一些构建机器学习管道和Apache Spark特有的其他东西的一般建议。

### 难以估计

像任何大而未知的事情一样，很难估计时间。我们有一个想法，我们需要做什么来实现我们的目标。但随着时间的推移，这些想法一个接一个地被证明是错误的。我们必须接受这一点，并确保我们能够快速迭代。

会有很大的未知数。确保您的团队能够迭代。

### 在开始之前验证数据是否整洁

当我们开始使用ML管道时，我们已经收集了大约3年的原始数据。我们将这些原始数据推送到我们的分析商店，将其转化为聚合数据。我们没有使用原始数据做任何事情，而是保留它，以防一切都坏了，我们需要重建我们的分析存储。原始数据是CSV文件的形式。我们没有注意到数据有问题。出现这些问题是因为编写文件的代码随着时间的推移而发生了更改，因此一些错误悄悄地出现，而这些错误没有被注意到。在构建机器学习管道时，我们确实修复了在数据中产生问题的错误。最后我们还在Apache spark中编写代码来清理历史数据。问题是，我们发现它是在所有事情的中间，而不是开始，这增加了我们团队的难度。

在开始之前，请确保数据正常。

### 对数据进行一次预处理，对模型进行一百万次的训练

为了训练机器学习模型，我们最初尝试加载所有数据。我们数据的大小是TBs。我们发现每次都试着把所有的东西都装进去，结果训练很慢。它还使得改进模型的迭代速度变慢。我们意识到不需要每次都加载所有数据。我们没有使用所有列的数据。我们做了一些预处理，并创建了一个新的较小的数据集，其中包含我们训练模型所需的列。我们还确保我们没有删除原始数据源，这将作为备份，以防我们随着时间的推移弄糟了一些东西。

不要把ETL和模型训练混为一谈。如果你正在训练1000个模型，你不想做1000次预处理。做一次预处理，保存到某个地方，然后使用它来满足您的所有模型培训需求。

### 为不同的团队成员提供易于探索的访问权限

如前所述，我们在AWS S3中存储原始数据以备备份从数据科学的角度来看，我们并没有很容易进行探索。当我们开始在S3中转储数据时，这并不是我们的目标。但是当我们开始研究ML时，我们发现向每个人提供易于探索的访问是至关重要的。

仅仅提供读取权限是不够的。人们不能只在笔记本上下载TBs的数据，不是吗？假设有人下载了TBs的数据，那么他们会在笔记本上做什么呢？人们不会随身携带32核笔记本电脑，通常是为了在合理的时间内处理TBs的数据。只是在浪费大家的时间。

我们发现使用Apache Spark支持的类似笔记本的环境可以达到这个目的。笔记本的例子有朱庇特，齐柏林飞艇。当我们有持久集群时，发现jupyter工作得很好。由于内置集成，齐柏林飞艇赢得了AWS EMR（AWS的托管星火集群）的冠军。

让人们阅读TBs，并希望他们能够理解出来，这显然是荒谬的。你必须给他们正确的工具，让他们能够理解。像jupyter，zeppelin这样的笔记本电脑在云端星火集群的支持下为我们工作。

### 大数据必须监控

当你在研究物理变化的大数据定律时。只是开玩笑，但是传统的软件工程方法根本不起作用。正常的程序需要几分钟，大数据可能需要几小时到几天，这取决于你在做什么和如何做。但我们并不是生活在不得不等待成批工作完成的日子里。也许是十年前，但现在不是了。

与传统编程相比，在大数据情况下减少批处理作业的完成时间更为复杂。使用云计算，我们可以水平缩放正在使用的机器并减少时间。但我们应该增加机器的数量还是彻底改变机器的类型呢？我们是CPU绑定、RAM绑定、网络绑定还是磁盘绑定？在这个分布式环境中，我们的瓶颈在哪里？这是我们需要回答的问题，以减少执行时间。

对于Apache Spark来说，很难找出所需的机器类型。Amazon EMR附带ganglia，让我们可以一眼就监视集群内存/CPU。但有时我们不得不去检查底层的EC2实例监控，因为ganglia并不完美。两者一起使用有帮助。我们发现我们的ETL作业与正在训练机器学习模型的作业具有不同的执行配置文件。ETL需要大量的网络和内存，ML训练需要更多的计算量。我们为这两种类型的作业选择了不同的实例类型。

需要CPU/内存/网络/IO监控来优化成本。我们发现不同的作业（ETL，ML）有不同的机器需求。

### 需要在一开始就对机器学习预测进行基准测试

您对机器学习模型的预测有延迟要求吗？如果是，请记住，找出框架的训练模型是否能够满足您的延迟需求。很容易从理论上对模型所涉及的数学有一个基本的了解，并认为它会很快。结果发现，还有其他事情可能导致预测不如理论上预期的那么快。正如聪明人所说，如果不进行测试，它就不会起作用。

建立一个简单的模型并对其进行基准测试。如果你在建立管道后发现这一点，可能会浪费你很多时间。当我们发现spark不能满足我们的延迟需求时，我们发现spark很难做到这一点。我们使用一个名为mleap的库来改进中共享的预测延迟。

如果有延迟要求，请从要使用的框架中创建一个简单的模型。准确度、精密度或任何其他指标都无关紧要。只需将它作为预测延迟的基准。

### 不管AWS如何设计，S3都不是一个文件系统

使用AWS的GUI或CLI很容易忘记S3不是一个文件系统。它是一个对象存储。如果您不知道这是什么，那么对象存储就是一个键值存储，其中的值是一个对象。对象可以是json、image等。

这一区别很重要，因为在S3中重命名对象的速度不如在真正的文件系统中那样快。如果您在文件系统中移动一个对象，它可能会非常快，这取决于您正在使用的是什么，但在S3中却不希望如此。

为什么这很重要？因为当通过Apache spark向S3写入数据时，apachespark会写入临时文件，然后将它们移动到新的密钥。如果您使用的是一个文件系统，而不是S3，这可能是可以的，因为上面解释了原因。Apache spark中有一个设置，告诉它不要写临时文件，而要写最终输出。我们使用它，节省了Apache spark编写AWS S3的大量时间。到目前为止我们还没有遇到问题。

### Apache spark主要基于Scala

如果您使用的是Apache spark，您应该知道它主要是基于Scala的。Java和PythonAPI确实可以工作，但internet上的示例大多是基于Scala的。如果你在他们的邮件列表上寻求帮助，人们会很乐意提供帮助，但主要是在Scala中，因为这是他们正在使用的API。

我们使用Java，因为我们的技术栈在Java上。我们刚起步时，既没有机器学习方面的专业知识，也没有Scala。我们认为机器学习对我们的业务是必不可少的Scala不是。我们不能让我们的团队处理Scala的学习曲线和机器学习。这些只是实际考虑，以确保整个项目不会成为一场灾难。

这很好地确保了我们能够得到最终产品。但是当面临Apache Spark的问题时，它就成了一个恼人的问题。我们遇到了问题，找到了解决办法。但解决方案是Scala的。将Scala转换成Java并不困难。将Spark Scala转换为Spark Java很困难。因为api在Java中更难使用。

如果您不了解Scala并想使用Spark Mllib，那么您可能需要考虑在选择的语言方面进行折衷。这不是一个理想的工程解决方案，而是一个实用的解决方案。记住软件工程是在迭代中发生的。让它工作然后让它更好。在生产中投入使用要比尝试一个从未投入生产的完美解决方案更令人满意。

### 如果你和一个团队一起工作，知识共享就变得很重要

如果你将机器学习与已经存在的系统集成在一起，你就必须和其他开发人员打交道。此外，你们还需要和业务人员、运营人员、营销人员等进行交流，除非你们正在开发一款人工智能产品，否则这些人大多对机器学习不太了解。因为机器学习是更大业务解决方案的一部分，而不是整个解决方案的一部分，所以他们也必须花时间处理其他事情。他们不能坐下来做学习机器学习的课程。

做一些关于机器学习的知识分享。你不必教他们算法和其他东西，但你需要消除机器学习的神秘感。在高层次上解释一些外行术语 - 培训/测试/验证集、模型、算法等中涉及的常用术语。

很容易忘记机器学习充满了术语。你可能完全熟悉他们，但其他人却不熟悉。团队中的其他人可能会觉得这是一种陌生的语言，他们可能会感到困惑。不是每个人都上过ML课程。

### 对数据进行版本控制可能是个好主意

您可能希望为数据构建一个版本控制方案，并允许在不重新部署整个软件的情况下切换不同的模型培训代码以使用不同的数据集。我们创建了一些模型，用一些数据对它们进行了尝试，发现这些数据还不够。这些模型很有效，但还不够好。

因此，我们为我们的数据位置构建了一个版本控制方案，这样我们就可以在v1上训练模型并继续生成下一个版本，在新版本中有足够的数据之后，我们就可以切换模型训练代码来使用新的数据。我们还制作了一个UI，它允许我们控制机器学习算法的参数，允许我们对某些特定参数进行基本过滤，并且能够指定我们希望用于训练的数据量。基本上，可以通过UI轻松地配置一些东西，以确保对用于培训的数据进行更改不需要重新部署。

If you enjoyed this you might also like this where I share how [we choose a data warehouse](https://medium.com/infinity-aka-aseem/why-we-chose-snowflake-as-our-data-warehouse-c5964a00802a) for easy to explore access.

### Thanks for reading :) If you enjoyed it, hit that clap button below as many times as possible!