机器学习规则 (Rules of Machine Learning)：关于机器学习工程的最佳实践

马丁·辛克维奇

本文档旨在帮助已掌握机器学习基础知识的人员从 Google 机器学习的最佳实践中受益。它介绍了一种机器学习样式，类似于 Google C++ 样式指南和其他常用的实用编程指南。如果您学习过机器学习方面的课程，或者拥有机器学习模型的构建或开发经验，则具备阅读本文档所必需的背景知识。

术语

在我们讨论有效的机器学习的过程中，会反复提到下列术语：

- **实例**：要对其进行预测的事物。例如，实例可以是一个网页，您希望将其分类为“与猫相关”或“与猫无关”。
- **标签**：预测任务的答案，它可以是由机器学习系统生成的答案，也可以是训练数据中提供的正确答案。例如，某个网页的标签可能是“与猫相关”。
- **特征**：预测任务中使用的实例的属性。例如，某个网页可能具有“包含字词‘猫’”这一特征。
- **特征列**：一组相关特征，例如用户可能居住的所有国家/地区的集合。样本的特征列中可能包含一个或多个特征。“特征列”是 Google 专用的术语。特征列在 Yahoo/Microsoft 使用的 VM 系统中被称为“命名空间”或场 (https://www.csie.ntu.edu.tw/~cjlin/libffm/)。
- **样本**：一个实例（及其特征）和一个标签。
- **模型**：预测任务的统计表示法。您使用样本训练一个模型，然后使用该模型进行预测。
- **指标**：您关心的一个数值。也许（但不一定）可以直接得到优化。
- **目标**：算法尝试优化的一种指标。
- **管道**：机器学习算法的基础架构。管道包括从前端收集数据、将数据放入训练数据文件、训练一个或多个模型以及将模型运用到生产环境。
- **点击率**：点击广告中的链接的网页访问者所占的百分比。

概览

要打造优质的产品：

请把自己看成是一位出色的工程师，而不是一位机器学习专家。

实际上，您将面临的大部分问题都是工程问题。即使在使用出色的机器学习专家掌握的所有资源的情况下，大多数收获也是由合适的特征（而非精确的机器学习算法）带来的。所以，进行机器学习的基本方法是：

1. 确保管道从头到尾都稳固可靠。
2. 从制定合理的目标开始。
3. 以简单的方式添加常识性特征。
4. 确保管道始终稳固可靠。

上述方法将在长时间内取得很好的效果。只要您仍然可以通过某种简单的技巧取得进展，就不应该偏离上述方法。增加复杂性会减缓未来版本的发布。

当您充分利用了所有的简单技巧，或许就到了探索机器学习最前沿技术的时候了。请参阅第三阶段 (#ml_phase_iii_slowed_growth_optimization_refinement_and_complex_models)的“机器学习项目”部分。

本文档结构如下：

1. 第一部分 (#before_machine_learning)可帮助您了解构建机器学习系统的时机是否已经成熟。
2. 第二部分 (#ml_phase_i_your_first_pipeline)介绍了如何部署第一个管道。
3. 第三部分 (#ml_phase_ii_feature_engineering)介绍了在向管道添加新特征时如何进行发布和迭代、如何评估模型，以及如何应对训练-应用偏差。
4. 最后一部分 (#ml_phase_iii_slowed_growth_optimization_refinement_and_complex_models)介绍了当您达到稳定阶段时该怎么做。
5. 之后是相关资源 (#related_work)列表和附录 (#appendix)，附录针对多次作为示例在本文档中提及的系统，提供了一些背景信息。

在进行机器学习之前

第 1 条规则：不要害怕发布未采用机器学习技术的产品。

机器学习技术很酷，但它需要数据。从理论上讲，您可以采用来自其他问题的数据，然后针对新产品调整模型，但其效果很可能不如基本的启发式算法。如果您认为机器学习技术能为您带来 100% 的提升，那么启发式算法可为您带来 50% 的提升。

例如，如果您要对应用市场中的应用进行排名，则可以将安装率或安装次数作为启发式算法指标。如果您要检测垃圾邮件，则可以滤除以前发送过垃圾邮件的发布商。此外，也不要害怕手动修改。如果您需要对联系人进行排名，可以按使用联系人的时间顺序由近及远对其排序（或按字母顺序排序）。如果您的产品并非必须使用机器学习技术，则在获得足够的数据之前，请勿使用该技术。

第 2 条规则：首先设计并实现指标。

在正式确定机器学习系统的功能之前，尽可能在当前系统中跟踪指标的值。这样做的原因如下：

1. 提前行动有助于更轻松地从系统的用户获得授权。
2. 如果您认为将来可能需要考虑某个方面，最好立即开始收集相关历史数据。
3. 如果您在设计系统时考虑到指标测量，将来会省下很多力气。具体而言，您不希望自己以后在日志中苦苦查找字符串以测量指标！
4. 您将发现哪些内容发生了变化以及哪些内容始终未变。例如，假设您希望直接优化单日活跃用户数。但是，在早期操纵系统的过程中，您可能会发现用户体验的显著改变并没有使该指标发生明显变化。

Google+ (#google_plus_overview) 团队会衡量每次阅读的展开次数、 转发次数、+1 次数、评论次数，以及每位用户的评论次数、转发次数等，然后在应用模型时利用这些数据来衡量帖子的质量。**另请注意，实验框架非常重要，您必须在实验框架中将用户分组为多个分桶，并按实验汇总统计信息。** 请参阅第 12 条规则 (#rule_12_don_t_overthink_which_objective_you_choose_to_directly_optimize)。

通过以更加自由的方式收集指标，您可以更加全面地了解您的系统。发现问题了？添加指标对其进行跟踪！对上个版本中发生的一些量变激动不已？添加指标对其进行跟踪！

第 3 条规则：选择机器学习技术而非复杂的启发式算法。

简单的启发式算法有利于推出产品。但复杂的启发式算法难以维护。当您获得足够的数据并基本确定自己要尝试实现的目标后，请考虑使用机器学习技术。与大多数软件工程任务一样，您需要不断更新方法（无论是启发式算法还是机器学习模型），而且您会发现机器学习模型更易于更新和维护（请参阅第 16 条规则 (#rule_16_plan_to_launch_and_iterate)）。

机器学习第一阶段：您的第一个管道

重点关注第一个管道的系统基础架构。虽然展望您将要进行的创新性机器学习的方方面面是一件很有趣的事，但如果您不先确认管道的可靠性，则很难弄清楚所发生的情况。

第 **4** 条规则：确保第一个模型简单易用，并正确实施基础架构。

第一个模型可以最有效地提升您的产品质量，因此不需要花哨，简单易用即可。但是，您会遇到很多预料之外的基础架构问题。在公开推出您精心构建的新机器学习系统之前，您必须确定以下几点：

- 如何为您的学习算法获取样本。
- 初步确定对于您的系统来说，“好”和“坏”的定义是什么。
- 如何将模型整合到应用中。您可以在线应用模型，也可以离线使用样本对模型进行预计算，并将结果存储在表格中。例如，您可能需要对网页进行预分类并将结果存储在表格中，但也可能需要在线对聊天消息进行分类。

选择简单的特征可以更轻松地确保：

- 将这些特征正确应用于您的学习算法。
- 模型学习出合理的权重。
- 将这些特征正确应用于服务器端。

当您有了能可靠做到上述三点的系统时，则表示您已完成大部分工作。简单的模型可为您提供基准指标和基准行为，您可以利用这些指标和行为测试更复杂的模型。某些团队以“中性”作为首次发布的目标 - 在首次发布时明确淡化机器学习成果，以避免分心。

第 **5** 条规则：撇开机器学习，单独测试基础架构。

确保基础架构可测试，且对系统的学习部分进行封装，以便测试这些部分之外的方方面面。具体而言：

1. 测试数据导入算法的效果。检查应填充的特征列是否已填充。在隐私权许可的情况下，手动检查输入到训练算法的数据。如果可能的话，查看管道中的统计信息，并与在其他地方处理的相同数据的统计信息进行比较。
2. 测试从训练算法得出模型的效果。确保训练环境中的模型与应用环境中的模型给出的分数相同（请参阅第 37 条规则 (#rule_37_measure_training_serving_skew)）。

机器学习具有不可预测性，因此要有用于训练环境和应用环境中创建样本的代码的测试；并确保您可以在应用期间加载和使用固定模型。此外，了解您的数据至关重要：请参阅分析大

型复杂数据集的实用建议
(http://www.unofficialgoogledatascience.com/2016/10/practical-advice-for-analysis-of-large.html)。

第 6 条规则：复制管道时注意丢弃的数据。

通常，我们通过复制现有管道来创建新管道（即货物崇拜编程
(https://en.wikipedia.org/wiki/Cargo_cult_programming)），且旧管道会丢弃一些新管道需要的数据。例如，Google+ (#google_plus_overview) 热门信息的管道会丢弃时间较早的帖子（因为它会不断尝试对最新的帖子进行排名）。此管道被复制用于 Google+ (#google_plus_overview) 信息流，在信息流中，时间较早的帖子仍然有意义，但旧管道仍会丢弃它们。另一种常见模式是仅记录用户看到的数据。因此，如果我们想要对用户看不到特定帖子的原因进行建模，此类数据就毫无用处，因为管道已丢弃所有负分类样本。Play 中也曾出现过类似的问题。在处理 Play 应用首页时，创建了一个新管道，其中还包含来自 Play 游戏着陆页的样本，但无任何特征可区分各个样本的来源。

第 7 条规则：将启发式算法转变为特征或在外部处理它们。

通常，机器学习尝试解决的问题并不是全新的问题。有一个现有的系统，它可用于排名、分类，或解决您正尝试解决的任何问题。这意味着有多种规则和启发式算法。**使用机器学习进行调整后，此类启发式算法可为您提供便利。** 您应该挖掘自己的启发式算法，了解它们所包含的任何信息，原因有以下两点。首先，向机器学习系统的过渡会更平稳。其次，这些规则通常包含大量您不愿意丢弃的关于系统的直觉信息。您可以通过以下四种方法使用现有启发式算法：

- 使用启发式算法进行预处理。如果特征非常好，则可以选择执行此操作。例如，在垃圾邮件过滤器中，如果发件人已被列入黑名单，则不要试图重新学习“已列入黑名单”的含义。屏蔽该邮件即可。这种方法最适合在二元分类任务中使用。
- 创建特征。直接通过启发式算法创建特征是一种很好的做法。例如，如果您使用启发式算法来计算查询结果的相关性分数，则可以将此分数纳为一个特征的值。您日后可能想要使用机器学习技术调整该值（例如，将该值转换为一个有限离散值组中的一个，或与其他特征相组合），但是首先请使用启发式算法生成的原始值。
- 挖掘启发式算法的原始输入。如果某个应用启发式算法结合了安装次数、文本中的字符数以及星期值，考虑将这些内容拆分开来，并作为输入单独提供给学习算法。部分适用于集成学习的技巧也适用于此（请参阅第 40 条规则 (#rule_40_keep_ensembles_simple)）。
- 修改标签。当您感觉启发式算法会获取当前标签中未包含的信息时，可以选择进行此操作。例如，如果您正在尝试最大程度地增加下载次数，但同时也想要优质的内容，则可能的解决方案是用标签乘以应用获得的平均星数。您可以非常灵活地修改标签。请参阅“您的第一个目标” (#your_first_objective)。

在机器学习系统中使用启发式算法时，请务必留意是否会带来额外的复杂性。在新的机器学习算法中使用旧启发式算法有助于实现平稳过渡，但思考下是否有可以达到相同效果的更简单的方法。

监控

在一般情况下，请实行良好的警报安全机制，例如设计解决警报的步骤以及提供“信息中心”页面。

第 8 条规则：了解您的系统对新鲜程度的要求。

如果您使用一天前的模型，效果会降低多少？一周前的模型呢？一个季度前的模型呢？此类消息有助于您了解需要优先监控哪些方面。如果一天不更新模型会对您的产品质量产生严重影响，则最好让工程师持续观察相关情况。大多数广告投放系统每天都有新广告要处理，并且必须每天更新。例如，如果不更新 Google Play 搜索 (#google_play_overview)的机器学习模型，则不到一个月便会产生负面影响。Google+ (#google_plus_overview) 热门信息的某些模型中没有帖子标识符，因此无需经常导出这些模型。其他具有帖子标识符的模型的更新频率要高得多。另请注意，新鲜程度会随着时间而改变，尤其是在向模型中添加特征列或从中移除特征列时。

第 9 条规则：先检测问题，然后再导出模型。

很多机器学习系统都会经历导出模型以应用模型的阶段。如果导出的模型存在问题，则是面向用户的问题。

在导出模型之前，请进行健全性检查。具体而言，确保模型在处理预留数据方面表现合理。或者说，如果您一直认为数据存在问题，请不要导出模型。很多经常部署模型的团队在导出模型之前，会先检查 ROC 曲线下面积 (https://en.wikipedia.org/wiki/Receiver_operating_characteristic)（简称 AUC）。**尚未导出的模型存在问题时，需要发送电子邮件提醒；但面向用户的模型出现问题时，可能需要通过一个页面进行宣布。** 因此，最好先等待检查完毕并确保万无一失后再导出模型，以免对用户造成影响。

第 10 条规则：注意隐藏的问题。

相比其他类型的系统，这种问题更常见于机器学习系统。假设关联的特定表格不再更新，那么，机器学习系统会进行相应调整，其行为仍然会相当好，但会逐渐变糟。有时，您会发现有些表格已有几个月未更新，只需刷新一下，就可以获得比相应季度做出的所有其他改进都更有效的效果提升！特征的覆盖率可能会因实现变化而发生改变：例如，某个特征列可能在 90% 的样本中得到填充，但该比率突然下降到 60%。Google Play 曾有一个过时 6 个月的表

格，但仅刷新了一下该表格，安装率就提升了 2%。如果您对数据的统计信息进行跟踪，并不时地手动检查数据，就可以减少此类失败。

第 11 条规则：提供特征列的所有者及相关文档。

如果系统很大，且有很多特征列，则需要知道每个特征列的创建者或维护者。如果您发现了解某个特征列的人要离职，请确保有人知道相关信息。尽管很多特征列都有说明性名称，但针对特征的含义、来源以及预计提供帮助的方式提供更详细的说明，是一种不错的做法。

您的第一个目标

您会关注很多有关系统的指标或测量结果，但通常只能为您的机器学习算法指定一个**目标，即您的算法“尝试”优化的数值。** 在这里，我介绍一下目标和指标有何区别：指标是指您的系统报告的任意数字，可能重要，也可能不重要。另请参阅第 2 条规则 (#rule_2_first_design_and_implement_metrics)。

第 12 条规则：选择直接优化哪个目标时，不要想太多。

您想赚钱，想让用户满意，想让世界变得更美好。您关注的指标有很多，而且您应该对所有这些指标进行测量（请参阅第 2 条规则 (#rule_2_first_design_and_implement_metrics)）。不过，在早期的机器学习过程中，您会发现这些指标都呈上升趋势，甚至那些您没有选择直接优化的指标也是如此。例如，假设您关注点击次数和用户在网站上停留的时间。如果您优化点击次数，则用户在网站上停留的时间很可能也会增加。

所以，当您仍然可以轻松增加所有指标时，保持简单，不要过多考虑如何在不同的指标间实现平衡。但不要过度使用此规则：不要将您的目标与系统最终的运行状况相混淆（请参阅第 39 条规则 (#rule_39_launch_decisions_are_a_proxy_for_long_term_product_goals)）。此外，**如果您发现自己增大了直接优化的指标，但决定不发布系统，则可能需要修改某些目标。**

第 13 条规则：为您的第一个目标选择一个可观察且可归因的简单指标。

您往往并不知道真正的目标是什么。您以为自己知道，但当您盯着数据，对旧系统和新的机器学习系统进行对比分析时，您发现自己想调整目标。此外，团队的不同成员通常无法就什么是真正的目标达成一致意见。**机器学习目标应是满足以下条件的某种目标：易于测量且是“真正的”目标的代理。**实际上，通常没有“真正的”目标（请参阅第 39 条规则 (#rule_39_launch_decisions_are_a_proxy_for_long_term_product_goals)）。因此，请对简单的机器学习目标进行训练，并考虑在顶部添加一个“策略层”，以便您能够添加其他逻辑（最好是非常简单的逻辑）来进行最终排名。

要进行建模，最简单的指标是可直接观察到且可归因到系统操作的用户行为：

- 用户是否点击了此已排名链接？
- 用户是否下载了此已排名对象？
- 用户是否转发/回复/使用电子邮件发送了此已排名对象？
- 用户是否评价了此已排名对象？
- 用户是否将此显示的对象标记为了垃圾邮件/色情内容/攻击性内容？

避免一开始对间接影响进行建模：

- 用户第二天访问网站了吗？
- 用户在网站上停留了多长时间？
- 每日活跃用户数有多少？

其实，间接影响可成为出色的指标，可以在 A/B 测试和发布决策期间使用。

最后，不要试图让机器学习系统弄清楚以下问题：

- 用户在使用产品时是否感到满意？
- 用户是否对使用体验感到满意？
- 产品是否提升了用户的整体满意度？
- 这会对公司的整体运行状况产生什么样的影响？

所有这些都很重要，但也极难衡量。请改为使用代理指标：如果用户感到满意，他们会在网站上停留更长时间。如果用户感到满意，他们明天会再次访问网站。就满意度和公司运行状况而言，需要进行人为判断，以便将任意机器学习目标与您销售的产品的性质和业务计划关联起来。

第 **14** 条规则：从可解释的模型着手可更轻松地进行调试。

线性回归、逻辑回归和泊松回归均由概率模型直接推动。每个预测都可看作是一个概率或预期值。这样一来，相较于使用目标（0-1 损失、各种合页损失函数等）以尝试直接优化分类准确度或对效果进行排名的模型，这种模型更易于进行调试。例如，如果在训练中得出的概率与采用并排分析方式或通过检查生产系统的方式预测的概率之间存在偏差，则表明存在问题。

例如，在线性回归、逻辑回归或泊松回归中，**有一部分平均预测期望值等于平均标签值（一阶矩校准，或只是校准）的数据**。假设您没有正则化且算法已收敛，那么理论上即是如此，实际上也是差不多这种情形。如果您有一个特征，对于每个样本来说，其值要么是 0，要么是 1，则会校准 3 个特征值为 1 的样本集。此外，如果您有一个特征，对于每个样本来说，其值均为 1，则会校准所有样本集。

借助简单的模型，您可以更轻松地处理反馈环（请参阅第 36 条规则(#rule_36_avoid_feedback_loops_with_positional_features)）。通常情况下，我们会根据这些概率预测来做出决策；例如，以期望值（点击概率/下载概率等）为标准，按降序对帖子进行排名。 **但是，请注意，当选择要使用的模型时，您的决定比模型给出的数据概率更为重要（请参阅第 27 条规则(#rule_27_try_to_quantify_observed_undesirable_behavior)）。**

第 **15** 条规则：在策略层中区分垃圾内容过滤和质量排名。

质量排名是一门艺术，但垃圾内容过滤就像一场战争。对于使用您系统的用户来说，您使用哪些信号来确定高质量帖子将变得显而易见，而且这些用户会调整自己的帖子，使其具有高质量帖子的属性。因此，您的质量排名应侧重于对诚实发布的内容进行排名。您不应该因为质量排名学习器将垃圾内容排在前列而对其应用折扣。**同样，“少儿不宜”的内容也不应该在质量排名中进行处理。** 垃圾内容过滤则另当别论。您必须明白，需要生成的特征会不断变化。通常情况下，您会在系统中设置一些明显的规则（如果一个帖子收到三次以上的垃圾内容举报，请勿检索该帖子等等）。所有学习模型都必须至少每天更新。内容创作者的声誉会发挥很大作用。

在某个层级，必须将这两个系统的输出整合在一起。请注意，与过滤电子邮件中的垃圾邮件相比，在过滤搜索结果中的垃圾内容时，可能应该更加主动。这种说法的前提是您没有正则化且算法已收敛。一般来说大致是这样。此外，从质量分类器的训练数据中移除垃圾内容是一种标准做法。

机器学习第二阶段：特征工程

在机器学习系统生命周期的第一阶段，重要的问题涉及以下三个方面：将训练数据导入学习系统、对任何感兴趣的指标进行测量，以及构建应用基础架构。**当您构建了一个端到端的可稳定运行的系统，并且制定了系统测试和单元测试后，就可以进入第二阶段了。**

第二阶段的很多目标很容易实现，且有很多明显的特征可导入系统。因此，机器学习的第二阶段涉及导入尽可能多的特征，并以直观的方式将它们组合起来。在这一阶段，所有的指标应该仍然呈上升趋势，您将会多次发布系统，并且非常适合安排多名工程师，以便整合创建真正出色的学习系统所需的所有数据。

第 **16** 条规则：制定发布和迭代模型计划。

不要指望您现在正在构建的模型会是您将要发布的最后一个模型，也不要指望您会停止发布模型。因此，请考虑此次发布中增加的复杂性是否会减缓未来版本的发布。很多团队多年来每季度都会发布一个或多个模型。发布新模型的三个基本原因如下所示：

- 您将要添加新特征。

- 您将要调整正则化并以新方式组合旧特征。
- 您将要调整目标。

无论如何，构建模型时多考虑考虑并没有什么坏处：查看提供到样本中的数据有助于发现新信号、旧信号以及损坏的信号。因此，在构建模型时，请考虑添加、移除或重新组合特征的难易程度。考虑创建管道的全新副本以及验证其正确性的难易程度。考虑是否可以同时运行两个或三个副本。最后，不必担心此版本的管道有没有纳入第 16 个特征（共 35 个），下个季度会将其纳入。

第 17 条规则：从可直接观察和报告的特征（而不是经过学习的特征）着手。

这一点可能存在争议，但可以避免许多问题。首先，我们来介绍一下什么是学习的特征。学习的特征是由外部系统（例如非监督式集群系统）或学习器本身（例如通过因子模型或深度学习）生成的特征。这两种方式生成的特征都非常有用，但会导致很多问题，因此不应在第一个模型中使用。

如果您使用外部系统创建特征，请注意，外部系统有其自己的目标。外部系统的目标与您当前的目标之间可能仅存在一点点关联。如果您获取外部系统的某个瞬间状态，它可能就会过期。如果您从外部系统更新特征，则特征的含义可能会发生变化。如果您使用外部系统提供特征，请注意，采用这种方法需要非常小心。

因子模型和深度模型的主要问题是，它们是非凸模型。因此，无法保证能够模拟或找到最优解决方案，且每次迭代时找到的局部最小值可能不同。这种变化导致难以判断系统发生的某次变化的影响是有意义的还是随机的。通过创建没有深度特征的模型，您可以获得出色的基准效果。达到此基准后，您可以尝试更深奥的方法。

第 18 条规则：探索可跨情境泛化的内容的特征。

机器学习系统通常只是更大系统中的一小部分。例如，想象热门信息中可能会使用的帖子，在其显示到热门信息之前，很多用户已经对其进行 +1、转发或评论了。如果您将这些统计信息提供给学习器，它就会对在正在优化的情景中没有数据的新帖子进行推广。 YouTube (#youtube_overview) 的“接下来观看”可以使用来自 YouTube (#youtube_overview) 搜索的观看次数或连看次数（观看完一个视频后观看另一个视频的次数）或明确的用户评分来推荐内容。最后，如果您将一个用户操作用作标签，在其他情境中看到用户对文档执行该操作可以是很好的特征。借助所有这些特征，您可以向该情境中引入新内容。请注意，这与个性化无关：先弄清楚是否有人喜欢此情境中的内容，然后再弄清楚喜欢程度。

第 19 条规则：尽可能使用非常具体的特征。

对于海量数据，学习数百万个简单的特征比学习几个复杂的特征更简单。正在被检索的文档的标识符以及规范化的查询不会提供很多泛化作用，但可以让您的排名与频率靠前的查询的

标签保持一致。因此，请不要害怕具有以下特点的特征组：每个特征适用于您的一小部分数据但总体覆盖率在 90% 以上。您可以使用正则化来消除适用样本过少的特征。

第 20 条规则：组合和修改现有特征，以便以简单易懂的方式创建新特征。

有多种方式可以组合和修改特征。借助 TensorFlow 等机器学习系统，您可以通过转换(https://www.tensorflow.org/tutorials/linear#feature-columns-and-transformations)对数据进行预处理。最标准的两种方法是“离散化”和“组合”。

“离散化”是指提取一个连续特征，并从中创建许多离散特征。以年龄这一连续特征为例。您可以创建一个年龄不满 18 周岁时其值为 1 的特征，并创建年龄在 18-35 周岁之间时其值为 1 的另一个特征，等等。不要过多考虑这些直方图的边界：基本分位数给您带来的影响最大。

“组合”方法是指组合两个或更多特征列。在 TensorFlow 中，特征列指的是同类特征集（例如，{男性, 女性}、{美国, 加拿大, 墨西哥} 等等）。组合指的是其中包含特征的新特征列，例如，{男性, 女性} × {美国, 加拿大, 墨西哥}。此新特征列将包含特征（男性, 加拿大）。如果您使用的是 TensorFlow，并让 TensorFlow 为您创建此组合，则此（男性, 加拿大）特征将存在于表示加拿大男性的样本中。请注意，您需要拥有大量数据，才能使用具有三个、四个或更多基准特征列的组合学习模型。

生成非常大的特征列的组合可能会过拟合。例如，假设您正在执行某种搜索，您的某个特征列包含查询中的字词，另一个特征列包含文档中的字词。这时，您可以使用“组合”方法将这些特征列组合起来，但最终会得到很多特征（请参阅第 21 条规则(#rule_21_the_number_of_feature_weights_you_can_learn_in_a_linear_model_is_roughly_proportional_to_the_amount_of_data_you_have)）。

处理文本时，有两种备用方法。最严苛的方法是点积。点积方法采用最简单的形式时，仅会计算查询和文档间共有字词的数量。然后将此特征离散化。另一种方法是交集：如果使用交集方法，当且仅当文档和查询中都包含“pony”一词时，才会出现一个特征；当且仅当文档和查询中都包含“the”一词时，才会出现另一个特征。

第 21 条规则：您可以在线性模型中学习的特征权重数目与您拥有的数据量大致成正比。

关于模型的合适复杂度方面，有各种出色的统计学习理论成果，但您基本上只需要了解这条规则。在某次谈话中，曾有人表达过这样的疑虑：从一千个样本中是否能够学到任何东西，或者是否需要超过一百万个样本，他们之所以有这样的疑虑，是因为局限在了一种特定学习方式中。关键在于根据数据规模调整您的学习模型：

1. 如果您正在构建搜索排名系统，文档和查询中有数百万个不同的字词，且您有 1000 个有标签样本，那么您应该在文档和查询特征、TF-IDF

(https://en.wikipedia.org/wiki/Tf%E2%80%93idf) 和多个其他高度手动工程化的特征之间得出点积。您会有 1000 个样本，十多个特征。

2. 如果您有一百万个样本，则使用正则化和特征选择（可能）使文档特征列和查询特征列相交。这样一来，您将获得数百万个特征；但如果使用正则化，则您获得的特征会有所减少。您会有千万个样本，可能会产生十万个特征。
3. 如果您有数十亿或数千亿个样本，您可以使用特征选择和正则化，通过文档和查询标记组合特征列。您会有十亿个样本，一千万个特征。统计学习理论很少设定严格的限制，但能够提供很好的起点引导。

最后，请根据第 28 条规则
(#rule_28_be_aware_that_identical_short_term_behavior_does_not_imply_identical_long_term_behavior)
决定要使用哪些特征。

第 22 条规则：清理不再使用的特征。

未使用的特征会产生技术负债。如果您发现自己没有使用某个特征，而且将其与其他特征组合在一起不起作用，则将其从您的基础架构中删除。您需要让自己的基础架构保持简洁，以便尽可能快地尝试最有可能带来良好效果的特征。如有必要，他人可以随时将您的特征添加回来。

在决定要添加或保留哪些特征时，要考虑到覆盖率。即相应特征覆盖了多少个样本？例如，如果您有一些个性化特征，但只有 8% 的用户有个性化特征，那效果就不会很好。

同时，有些特征可能会超出其权重。例如，如果您的某个特征只覆盖 1% 的数据，但 90% 具有该特征的样本都是正分类样本，那么这是一个可以添加的好特征。

对系统的人工分析

在继续探讨机器学习的第三阶段之前，请务必重点了解一下在任何机器学习课程中都无法学到的内容：如何检查现有模型并加以改善。这更像是一门艺术而非科学，但是有几个有必要避免的反模式。

第 23 条规则：您不是典型的最终用户。

这也许是让团队陷入困境的最简单的方法。虽然 fishfood（在团队内部使用原型）和 dogfood（在公司内部使用原型）有许多优点，但员工应该看看是否符合性能要求。虽然应避免应用明显比较糟糕的更改，但在临近生产时，应对任何看起来比较合理的更改进行进一步测试，具体方法有两种：请非专业人员在众包平台上回答有偿问题，或对真实用户进行在线实验。

这样做的原因有如下两点。首先，您与代码的关系太密切了。您关注的可能是帖子的某个特定方面，或者您只是投入了太多感情（例如确认偏差）。其次，您的时间很宝贵。考虑一下九名工程师开一个小时会议所花的费用可以在众包平台上购买多少签约的人工标签。

如果您确实想获得用户反馈，请**使用用户体验方法**。在流程的早期阶段创建用户角色（请参阅比尔·布克斯顿的 Sketching User Experiences (https://play.google.com/store/books/details/Bill_Buxton_Sketching_User_Experiences_Getting_the?id=2vfPxocmLh0C) 一书中的描述），然后进行可用性测试（请参阅史蒂夫·克鲁格的 Don't Make Me Think (https://play.google.com/store/books/details/Steve_Krug_Don_t_Make_Me_Think_Revisited?id=QlduAgAAQBAJ) 一书中的描述）。用户角色是指创建假想用户。例如，如果您的团队成员都是男性，则有必要设计一个 35 岁的女性用户角色（使用用户特征完成），并查看其生成的结果，而不是只查看 10 位 25-40 岁男性的结果。在可用性测试中请真实用户体验您的网站（通过本地或远程方式）并观察他们的反应也可以让您以全新的视角看待问题。

第 **24** 条规则：衡量模型间的差异。

在向任何用户展示您的新模型之前，您可以进行的最简单（有时也是最有用）的一项衡量是，评估新模型的结果与生产有多大差别。例如，如果您有一项排名任务，则在整个系统中针对一批示例查询运行这两个模型，并查看结果的对称差分有多大（按排名位置加权）。如果差分非常小，那么您无需运行实验，就可以判断不会出现很大变化。如果差分很大，那么您需要确保这种更改可以带来好的结果。查看对称差分较大的查询有助于您了解更改的性质。不过，请确保您的系统是稳定的。确保模型与自身之间的对称差分较低（理想情况下为零）。

第 **25** 条规则：选择模型时，实用效果比预测能力更重要。

您的模型可能会尝试预测点击率。但归根到底，关键问题在于您用这种预测做什么。如果您使用该预测对文档进行排名，那么最终排名的质量比预测本身更重要。如果您要预测一个文档是垃圾内容的概率，然后选择一个取舍点来确定要阻断的内容，那么允许的内容的精确率更为重要。大多数情况下，这两项应该是一致的：当它们不一致时，带来的优势可能会非常小。因此，如果某种更改可以改善对数损失，但会降低系统的性能，则查找其他特征。当这种情况开始频繁发生时，说明您该重新审视模型的目标了。

第 **26** 条规则：在衡量的错误中寻找规律，并创建新特征。

假设您看到模型"弄错"了一个训练样本。在分类任务中，这种错误可能是假正例，也可能是假负例。在排名任务中，这种错误可能是假正例和假负例，其中正例的排名比负例的排名低。最重要的是，机器学习系统知道自己弄错了该样本，如果有机会，它会修复该错误。如果您向该模型提供一个允许其修正错误的特征，该模型会尝试使用它。

另一方面，如果您尝试根据系统不会视为错误的样本创建一个特征，该特征将会被系统忽略。例如，假设某人在 Play 应用搜索中搜索“免费游戏”。假设排名靠前的搜索结果中有一个是相关性较低的搞笑应用。因此，您为“搞笑应用”创建了一个特征。但是，如果您要最大限度地增加安装次数，并且用户在搜索免费游戏时安装了搞笑应用，那么“搞笑应用”特征不会达到您想要的效果。

如果模型弄错了您的某些样本，请在当前特征集之外寻找规律。例如，如果系统似乎在降低内容较长的帖子的排名，那么添加帖子长度。不要添加过于具体的特征。如果您要添加帖子长度，请不要试图猜测长度的具体含义，只需添加十多个特征，然后让模型自行处理（请参阅第 21 条规则
(#rule_21_the_number_of_feature_weights_you_can_learn_in_a_linear_model_is_roughly_proportional_to_the_amount_of_data_you_have)
）。这是实现目标最简单的方式。

第 27 条规则：尝试量化观察到的异常行为。

当现有的损失函数没有捕获您团队中的部分成员不喜欢的某些系统属性时，他们会开始有挫败感。此时，他们应该竭尽所能将抱怨转换成具体的数字。例如，如果他们认为 Play 搜索中显示的“搞笑应用”过多，则可以通过人工评分识别搞笑应用。（在这种情况下，您可以使用人工标记的数据，因为相对较少的一部分查询占了很大一部分流量。）如果您的问题是可衡量的，那么您可以开始将它们用作特征、目标或指标。一般规则是“**先量化，再优化**”。

第 28 条规则：请注意，短期行为相同并不意味着长期行为也相同。

假设您的新系统会查看每个 doc_id 和 exact_query，然后计算每个查询的每个文档的点击概率。您发现在并排分析和 A/B 测试中，其行为与您当前系统的行为几乎完全相同，考虑到它的简单性，您发布了它。不过，您发现它没有显示任何新应用。为什么？那是因为您的系统仅根据自己的查询历史记录显示文档，所以不知道应该显示新文档。

了解这种系统长期行为的唯一方法是，仅使用模型在线时获得的数据对其进行训练。这一点非常难。

训练-应用偏差

训练-应用偏差是指训练效果与应用效果之间的差异。出现这种偏差的原因可能是：

- 训练管道和应用管道中数据的处理方式有差异。
- 训练时和应用时所用数据有变化。
- 模型和算法之间有反馈环。

我们注意到 Google 的生产机器学习系统也存在训练-应用偏差，这种偏差对性能产生了负面影响。最好的解决方案是明确进行监控，以避免在系统和数据改变时引入容易被忽视的偏差。

第 29 条规则：确保训练效果和应用效果一样的最佳方法是，保存在应用时使用的特征集，然后将这些特征通过管道传输到日志，以便在训练时使用。

即使您不能对每个样本都这样做，也对一小部分样本这样做，以便验证应用和训练之间的一致性（请参阅第 37 条规则 (#rule_37_measure_training_serving_skew)）。采取了这项措施的 Google 团队有时会对结果感到惊讶。 YouTube (#youtube_overview) 首页改用这种在应用时记录特征的做法后，不仅大大提高了质量，而且减少了代码复杂度。目前有许多团队都已经在其基础设施上采用了这种方法。

第 30 条规则：按重要性对采样数据加权，不要随意丢弃它们！

数据过多时，总会忍不住采用前面的文件而忽略后面的文件。这是错误的做法。尽管可以丢弃从未向用户展示过的数据，但对于其他数据来说，按重要性加权是最佳选择。按重要性加权意味着，如果您决定以 30% 的概率对样本 X 进行抽样，那么向其赋予 10/3 的权重。**按重要性加权时，您仍然可以使用第 14 条规则 (#rule_14_starting_with_an_interpretable_model_makes_debugging_easier)中讨论的所有校准属性。**

第 31 条规则：如果您在训练和应用期间关联表格中的数据，请注意，表格中的数据可能会变化。

假设您将文档 ID 与包含这些文档的特征（例如评论次数或点击次数）的表格相关联。表格中的特征在训练时和应用时可能有所不同。那么，您的模型在训练时和应用时对同一文档的预测就可能会不同。要避免这类问题，最简单的方法是在应用时记录特征（请参阅第 32 条规则 (#rule_32_re_use_code_between_your_training_pipeline_and_your_serving_pipeline_whenever_possible)）。如果表格只是缓慢发生变化，那么您还可以每小时或每天创建表格快照，以获得非常接近的数据。请注意，这仍不能完全解决问题。

第 32 条规则：尽可能在训练管道和应用管道间重复使用代码。

批处理不同于在线处理。进行在线处理时，您必须在每个请求到达时对其进行处理（例如，您必须为每个查询单独进行查找），而进行批处理时，您可以组合任务（例如进行关联）。应用时，您进行的是在线处理，而训练时，您进行的是批处理。不过，您可以通过一些方法来重复使用代码。例如，您可以专门为自己的系统创建一个对象，其中所有查询结果和关联都能以非常易于人类读取的方式进行存储，且错误也可以轻松进行测试。然后，收集了所有信息后，您可以在应用和训练期间使用一种共同的方法，在人类可读对象（特定于您的系

统）和机器学习需要的任何格式之间架起一座桥梁。**这样可以消除训练-应用偏差的一个根源**。由此推知，在训练和应用时，尽量不要使用两种不同的编程语言。如果这样做，就几乎不可能共享代码了。

第 33 条规则：如果您根据 1 月 5 日之前的数据生成模型，则根据 1 月 6 日及之后的数据测试模型。

一般来说，要衡量模型的效果，应使用在训练模型所有数据对应的日期之后的日期收集的数据，因为这样能更好地反映系统应用到生产时的行为。如果您根据 1 月 5 日之前的数据生成模型，则根据 1 月 6 日及之后的数据测试模型。您一般会发现，使用新数据时模型的效果不如原来好，但应该不会太糟。由于可能存在的一些日常影响，您可能没有预测到平均点击率或转化率，但曲线下面积（表示正分类样本的分数高于负分类样本的概率）应该非常接近。

第 34 条规则：在有关过滤的二元分类（例如，垃圾邮件检测或确定有趣的电子邮件）中，在短期内小小牺牲一下效果，以获得非常纯净的数据。

在过滤任务中，标记为负分类的样本不会向用户显示。假设您的过滤器在应用时可屏蔽 75% 的负分类样本。您可能会希望从向用户显示的实例中提取额外的训练数据。例如，如果用户将您的过滤器未屏蔽的电子邮件标记为垃圾邮件，那么您可能想要从中学习规律。

但这种方法会引入采样偏差。如果您改为在应用期间将所有流量的 1% 标记为“预留”，并向用户发送所有预留样本，则您可以收集更纯净的数据。现在，过滤器屏蔽了至少 74% 的负分类样本。这些预留样本可以成为训练数据。

请注意，如果过滤器屏蔽了 95% 或以上的负分类样本，则此方法的可行性会降低。即便如此，如果您希望衡量应用效果，可以进行更低比例的采样（比如 0.1% 或 0.001%）。一万个样本足以非常准确地评估效果。

第 35 条规则：注意排名问题中存在的固有偏差。

当您彻底改变排名算法，导致出现不同的排名结果时，实际上改变了您的算法以后会处理的数据。这时，就会出现固有偏差，您应该围绕这种偏差来设计模型。具体方法有多种。以下是让您的模型青睐已见过的数据的方法。

1. 对覆盖更多查询的特征（而不是仅覆盖一个查询的特征）进行更高的正则化。通过这种方式，模型将青睐专门针对一个或几个查询的特征，而不是泛化到所有查询的特征。这种方法有助于防止十分热门的查询结果显示到不相关的查询中。请注意，这与以下更为传统的建议相左：对具有更多唯一值的特征列进行更高的正则化。
2. 仅允许特征具有正权重。这样一来，就可确保任何好特征都比“未知”特征合适。
3. 不选择只处理文档数据的特征。这是第一条规则的极端版本。例如，即使指定应用是热门下载应用（无论查询是什么），您也不想在所有地方都展示它。如果不选择只处理文

档数据的特征，这一点很容易做到。您之所以不想在所有地方展示某个特定的热门应用，是因为让用户可以找到所有所需应用至关重要。例如，如果一位用户搜索“赏鸟应用”，他/她可能会下载“愤怒的小鸟”，但那绝对不是他/她想要的应用。展示此类应用可能会提高下载率，但最终却未能满足用户的需求。

第 36 条规则：通过位置特征避免出现反馈环。

内容的位置会极大地影响用户与其互动的可能性。如果您将应用放在首位，则应用获得的点击率更高，导致您认为用户更有可能点击该应用。处理此类问题的一种方法是添加位置特征，即关于内容在网页中的位置的特征。您可以使用位置特征训练模型，使模型学习（例如）对特征“1stposition”赋予较高的权重。因此，对于具有“1stposition=true”特征的样本的其他因素，模型会赋予较低的权重。然后，在应用时，您不向任何实例提供位置特征，或为所有实例提供相同的默认特征，因为在决定以怎样的顺序显示候选实例之前，您就对其进行了打分。

请注意，因为训练和测试之间的这种不对称性，请务必在位置特征与模型的其余特征之间保持一定的分离性。让模型成为位置特征函数和其余特征函数之和是理想的状态。例如，不要将位置特征与任何文档特征组合在一起。

第 37 条规则：测量训练/应用偏差。

一般来说，很多情况都会引起偏差。此外，您可以将其分为以下几个部分：

- 训练数据和预留数据的效果之间的差异。一般来说，这种情况始终存在，而且并非总是坏事。
- 预留数据和“次日”数据的效果之间的差异。同样，这种情况始终存在。您应该调整正则化，以最大程度地提升次日数据的效果。不过，如果与预留数据相比，次日数据效果下降明显，则可能表明某些特征具有时效性，而且可能会降低模型的效果。
- “次日”数据和实时数据的效果之间的差异。如果您将模型应用于训练数据中的某个样本，并在应用时使用同一样本，那么您得到的结果应该完全相同（请参阅第 5 条规则 (https://developers.google.com/machine-learning/guides/rules-of-ml/rule_5_test_the_infrastructure_independently_from_the_machine_learning) ）。因此，此处的差异很可能表示出现了工程错误。

机器学习第三阶段：缓慢增长、优化细化和复杂模型

第二阶段即将结束时会出现一些信号。首先，月增长开始减弱。您将开始在指标之间做出取舍：在部分试验中，您会看到一些指标上升了，而另一些指标下降了。情况变得有趣起来。由于越来越难实现增长，因此机器学习系统必须变得更加复杂。注意：相比之前两个部分，

本部分中会有较多的纯理论性规则。我们见过许多团队在机器学习的第一阶段和第二阶段非常满意。但到了第三阶段后，他们必须找到自己的道路。

第 38 条规则：如果目标不协调，并成为问题，就不要在新特征上浪费时间。

当您的衡量结果稳定时，您的团队会开始关注当前机器学习系统的目标范围之外的问题。如前所述，如果现有算法目标未涵盖产品目标，则您需要修改算法目标或产品目标。例如，您可以优化点击次数、+1 次数或下载次数，但让发布决策部分依赖于人工评分者。

第 39 条规则：发布决策代表的是长期产品目标。

Alice 有一个关于减少预测安装次数的逻辑损失的想法。她添加了一个特征。逻辑损失降低了。当她运行在线实验时，看到安装率增加了。但是，在发布评审会上，有人指出，每日活跃用户数减少了 5%。于是，团队决定不发布该模型。Alice 很失望，但现在她意识到发布决策取决于多个条件，只有一部分条件可以通过机器学习直接得到优化。

事实上，现实世界并不是网游世界：没有“生命值”来确定产品的运行状况。团队必须使用自己收集的统计信息来尝试有效地预测系统未来的表现会如何。他们需要关注互动度、日活跃用户数 (DAU)、30 日 DAU、收入以及广告主的投资回报率。这些可在 A/B 测试中衡量的指标本身仅代表了以下更长期目标：让用户满意、增加用户数量、让合作伙伴满意以及实现盈利，进一步，您还可以认为它们代表了发布优质且实用的产品，以及五年后公司繁荣发展。

唯一可以轻松做出发布决策的情况是，所有指标都在变好（或至少没有变差）。 如果团队能够在复杂的机器学习算法和简单的启发式算法之间做出选择，而对所有这些指标来说，简单的启发式算法可以提供更好的效果，那么应该选择启发式算法。此外，并未对所有可能的指标值进行明确排名。具体而言，请考虑以下两种情形：

实验	每日活跃用户数	收入/日
A	100 万	400 万美元
B	200 万	200 万美元

如果当前系统是 A，那么团队不太可能会改用 B。如果当前系统是 B，那么团队不太可能会改用 A。这似乎与理性行为背道而驰；但是，对更改指标的预测可能会成功也可能不会，因此这两种改变都蕴含着巨大的风险。每个指标都涵盖了团队所担心的一些风险。

此外，没有一个指标涵盖团队最关心的问题，即“五年后我的产品将何去何从”？

另一方面，个人更倾向于选择可以直接优化的目标。 大多数机器学习工具也都青睐这样的环境。在这样的环境下，快速创建新特征的工程师能稳定地进行一系列发布。一种称为“多目标学习”的机器学习已开始解决此问题。例如，您可以提出约束满足问题，对每个指标设定下

限，并优化指标的一些线性组合。不过，即使如此，也并不是所有指标都可以轻松框定为机器学习目标：如果用户点击了文档或安装了应用，那是因为相应内容展示出来了。但要弄清楚用户为什么访问您的网站就难得多。如何预测整个网站未来的成功状况属于 AI 完备问题 (https://en.wikipedia.org/wiki/AI-complete)：与计算机视觉或自然语言处理一样难。

第 **40** 条规则：保证集成学习简单化。

采用原始特征并直接对内容进行排名的统一模型是最易于进行调试和理解的模型。但是，集成学习模型（将其他模型的分数结合到一起的模型）可以实现更好的效果。**为了简单起见，每个模型应该要么是仅接受其他模型的输入的集成学习模型，要么是接受多个特征的基本模型，但不能两者皆是。** 如果在单独训练的模型之上还有其他模型，则组合它们会导致不良行为。

使用简单的模型进行集成学习（仅将“基本”模型的输出作为输入）。此外，您还需要将属性强加到这些集成学习模型上。例如，基本模型生成的分数的升高不应使集成学习模型的分数有所降低。另外，如果传入的模型在语义上可解释（例如，经过校准），则最理想，因为这样一来，即使基本模型发生改变，也不会扰乱集成学习模型。另外，**强制要求：如果基本分类器的预测概率增大，不会使集成学习模型的预测概率降低。**

第 **41** 条规则：效果达到平稳后，寻找与现有信号有质的差别的新信息源并添加进来，而不是优化现有信号。

您添加了一些有关用户的受众特征信息，也添加了一些有关文档中字词的信息。您探索了模板，并调整了正则化。但在几个季度的发布中，关键指标的提升幅度从来没有超过 1%。现在该怎么办？

是时候开始为截然不同的特征（例如，用户在过去一天内、一周内或一年内访问的文档的历史记录，或者其他属性的数据）构建基础架构了。您可以使用维基数据 (https://en.wikipedia.org/wiki/Wikidata)条目或公司内部信息（例如，Google 的知识图谱 (https://en.wikipedia.org/wiki/Knowledge_Graph)）。利用深度学习。开始调整您对投资回报的预期，并付出相应的努力。与在任何工程项目中一样，您必须对添加新特征的好处与增加复杂性的成本进行一番权衡。

第 **42** 条规则：不要期望多样性、个性化或相关性与热门程度之间的联系有您认为的那样密切。

一组内容中的多样性可以有多种含义，其中内容来源的多样性是最常见的一种。个性化意味着每个用户获得贴合其个人需求的结果。相关性意味着某个特定查询的结果更适合该查询，而非其他任何查询。因此，这三个属性均具有不同于常态的定义。

但常态往往很难被打败。

请注意，如果您的系统在测量点击次数、访问时间、观看次数、+1 次数、转发次数等数据，那么您测量的是内容的**热门程度**。团队有时会尝试学习具备多样性的个性化模型。为实现个性化，他们会添加支持系统进行个性化（代表用户兴趣的部分特征）或多样化（表明相应文档是否与其他返回的文档有任何相同特征的特征，例如作者或内容）的特征，然后发现这些特征的权重比预期低（或者有时是不同的信号）。

这并不意味着多样性、个性化或相关性不重要。正如上一条规则中所指出的那样，您可以进行后期处理来增加多样性或相关性。如果您看到更长期的目标有所增长，您可以声明除了热门程度外，多样性/相关性也很有价值。然后，您可以继续采用后期处理方法，也可以根据多样性或相关性直接修改目标。

第 **43** 条规则：在不同的产品中，您的好友基本保持不变，但您的兴趣并非如此。

Google 的团队通过以下做法取得了大量进展：采用一个预测产品中某种联系的紧密程度的模型，并使用该模型对其他产品进行准确预测。您的好友保持不变。另一方面，我曾见过几个团队在应对多个产品间的个性化特征时捉襟见肘。是的，当时看起来应该可以奏效的。但现在看来并没有。有时可以奏效的方法是，使用一个属性的原始数据来预测另一个属性的行为。此外，请注意，仅仅是知道用户有其他属性的历史记录也会有帮助。例如，两个产品上出现了用户活动或许本身就可以说明该问题。

相关资源

Google 内部和外部有许多关于机器学习的文档。

- 机器学习速成 (https://developers.google.com/machine-learning/guides/crash-course/)课程：应用机器学习简介。
- 机器学习：概率法 (https://www.cs.ubc.ca/~murphyk/MLbook/)，凯文·墨菲著，帮助了解机器学习领域。
- 分析大型复杂数据集的实用建议 (http://www.unofficialgoogledatascience.com/2016/10/practical-advice-for-analysis-of-large.html)：一种考虑数据集的数据科学方法。
- 深度学习 (http://www.iro.umontreal.ca/~bengioy/dlbook/)，伊恩·古德费洛等著，帮助学习非线性模型。
- 关于技术负债 (http://research.google.com/pubs/pub43146.html)的 Google 论文，其中提供了许多一般性建议。
- Tensorflow 文档 (https://www.tensorflow.org/)。

致谢

感谢 David Westbrook、Peter Brandt、Samuel Ieong、Chenyu Zhao、Li Wei、Michalis Potamias、Evan Rosen、Barry Rosenberg、Christine Robson、James Pine、Tal Shaked、Tushar Chandra、Mustafa Ispir、Jeremiah Harmsen、Konstantinos Katsiapis、Glen Anderson、Dan Duckworth、Shishir Birmiwal、Gal Elidan、Su Lin Wu、Jaihui Liu、Fernando Pereira 和 Hrishikesh Aradhye 对本文档进行多处更正、提供建议和有用示例。此外，还要感谢 Kristen Lefevre、Suddha Basu 和 Chris Berg 对早期版本提供的帮助。任何错误、遗漏或（喘气声！）不受欢迎的看法均由本人承担责任。

附录

本文档中多处提到了一些 Google 产品。为了提供更多背景信息，我将在下面对几个最常见的示例进行简单说明。

YouTube 概览

YouTube 是流式视频服务。YouTube 的“接下来观看”和 YouTube 首页团队均使用机器学习模型对推荐视频进行排名。“接下来观看”会推荐在当前视频播放完后观看的视频，而首页向浏览首页的用户推荐视频。

Google Play 概览

Google Play 有许多解决各种问题的模型。Play 搜索、Play 首页个性化推荐和“用户还安装了以下应用”都采用了机器学习技术。

Google+ 概览

Google+ 在各种情况下采用了机器学习技术，例如对用户可以看见的帖子“信息流”中的帖子进行排名时、对“热门信息”中的帖子（目前非常热门的帖子）进行排名时、对您认识的人进行排名时，等等。

上次更新日期：八月 23, 2018