

1.Kaggle 基本介绍

Kaggle 于 2010 年创立，专注数据科学，机器学习竞赛的举办，是全球最大的数据科学社区和数据竞赛平台。在 Kaggle 上，企业或者研究机构发布商业和科研难题，悬赏吸引全球的数据科学家，通过众包的方式解决建模问题。而参赛者可以接触到丰富的真实数据，解决实际问题，角逐名次，赢取奖金。诸如 Google，Facebook，Microsoft 等知名科技公司均在 Kaggle 上面举办过数据挖掘比赛。2017年3月，Kaggle 被 Google CloudNext 收购。

1.1 参赛方式

可以以个人或者组队的形式参加比赛。组队人数一般没有限制，但需要在 Merger Deadline 前完成组队。为了能参与到比赛中，需要在 Entry Deadline 前进行至少一次有效提交。最简单地，可以直接提交官方提供的 Sample Submission。关于组队，建议先单独个人进行数据探索和模型构建，以个人身份进行比赛，在比赛后期（譬如离比赛结束还有 2~3 周）再进行组队，以充分发挥组队的效果（类似于模型集成，模型差异性越大，越有可能有助于效果的提升，超越单模型的效果）。当然也可以一开始就组好队，方便分工协作，讨论问题和碰撞火花。

Kaggle 对比赛的公正性相当重视。在比赛中，每个人只允许使用一个账号进行提交。在比赛结束后 1~2 周内，Kaggle 会对使用多账号提交的 Cheater 进行剔除（一般会对 Top 100 的队伍进行 Cheater Detection）。在被剔除者的 Kaggle 个人页面上，该比赛的成绩也会被删除，相当于该选手从没参加过这个比赛。此外，队伍之间也不能私自分享代码或者数据，除非在论坛上面公开发布。

比赛一般只提交测试集的预测结果，无需提交代码。每人（或每个队伍）每天有提交次数的限制，一般为2次或者5次，在 Submission 页面会有提示。

1.2 比赛获奖

Kaggle 比赛奖金丰厚，一般前三名均可以获得奖金。在最近落幕的第二届 National Data Science Bowl 中，总奖金池高达 100W 美刀，其中第一名可以获得 50W 美刀的奖励，即使是第十名也能收获 2.5W 美刀的奖金。

获奖的队伍需要在比赛结束后 1~2 周内，准备好可执行的代码以及 README，算法说明文档等提交给 Kaggle 来进行获奖资格的审核。Kaggle 会邀请获奖队伍在 Kaggle Blog 中发表 Interview，来分享比赛故事和经验心得。对于某些比赛，Kaggle 或者主办方会邀请获奖队伍进行电话/视频会议，获奖队伍进行 Presentation，并与主办方团队进行交流。

1.3 比赛类型

从 Kaggle 提供的官方分类来看，可以划分为以下类型（如下图1所示）：

- ◆ Featured：商业或科研难题，奖金一般较为丰厚；
- ◆ Recruitment：比赛的奖励为面试机会；
- ◆ Research：科研和学术性较强的比赛，也会有一定的奖金，一般需要较强的领域和专业知识；
- ◆ Playground：提供一些公开的数据集用于尝试模型和算法；
- ◆ Getting Started：提供一些简单的任务用于熟悉平台和比赛；

◆ In Class：用于课堂项目作业或者考试。







	Google Cloud & YouTube-8M Video Understanding Challenge Can you produce the best video tag predictions? <i>Featured</i> · a month to go	\$100,000 522 teams
	Facebook V: Predicting Check Ins Identify the correct place for check ins <i>Recruitment</i> · 10 months ago	Jobs 1,212 teams
	Melbourne University AES/MathWorks/NIH Seizure Prediction Predict seizures in long-term human intracranial EEG recordings <i>Research</i> · 5 months ago	\$20,000 478 teams
	Sentiment Analysis on Movie Reviews Classify the sentiment of sentences from the Rotten Tomatoes dataset <i>Playground</i> · 2 years ago	861 teams
	Titanic: Machine Learning from Disaster Start here! Predict survival on the Titanic and get familiar with ML basics <i>Getting Started</i> · 3 years to go	6,905 teams
	Large-scale classification-SYSU-2017 Limited This is a in-class competition of classification. <i>In-Class</i> · 3 months to go	13 teams

图1. Kaggle 比赛类型

从领域归属划分：包含搜索相关性，广告点击率预估，销量预估，贷款违约判定，癌症检测等。

从任务目标划分：包含回归，分类（二分类，多分类，多标签），排序，混合体（分类+回归）等。

从数据载体划分：包含文本，语音，图像和时序序列等。

从特征形式划分：包含原始数据，明文特征，脱敏特征（特征的含义不清楚）等。

1.4 比赛流程

一个数据挖掘比赛的基本流程如下图2所示，具体的模块我将在下一章进行展开陈述。

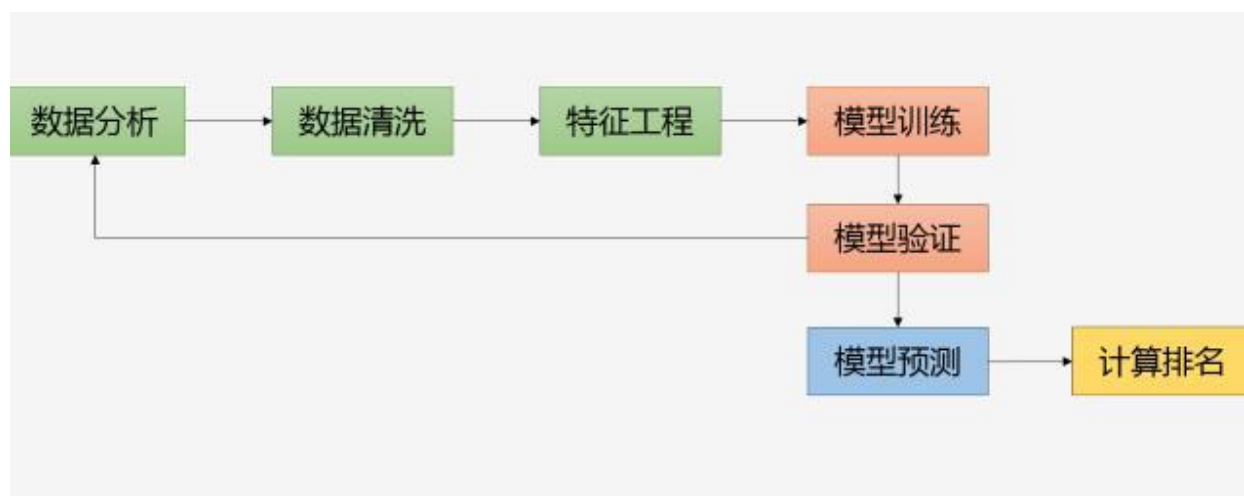


图2. 数据挖掘比赛基本流程

这里想特别强调的一点是，Kaggle 在计算得分的时候，有Public Leaderboard (LB)和 Private LB 之分。具体而言，参赛选手提交整个测试集的预测结果，Kaggle 使用测试集的一部分计算得分和排名，实时显示在 Public LB上，用于给选手提供及时的反馈和动态展示比赛的进行情况；测试集的剩余部分用于计算参赛选手的最终得分和排名，此即为 Private LB，在比赛结束后会揭晓。用于计算 Public LB 和 Private LB 的数据有不同的划分方式，具体视比赛和数据类型而定，一般有随机划分，按时间划分或者按一定规则划分。

这个过程可以概括如下图3所示，其目的是避免模型过拟合，以得到泛化能力好的模型。如果不设置 Private LB（即所有的测试数据都用于计算 Public LB），选手不断地从 Public LB（即测试集）中获得反馈，进而调整或筛选模型。这种情况下，测试集实际上是作为验证集参与到模型的构建和调优中来。Public LB上面的效果并非是在真实未知数据上面的效果，不能可靠地反映模型的效果。划分 Public LB 和 Private LB 这样的设置，也在提醒参赛者，我们建模的目标是要获得一个在未知数据上表现良好的模型，而并非仅仅是在已知数据上效果好。



图3. 划分 Public LB 和 Private LB的目的

（图参考 Owenzhang 的分享 [1]）

2.数据挖掘比赛基本流程

从上面图2可以看到，做一个数据挖掘比赛，主要包含了数据分析，数据清洗，特征工程，模型训练和验证等四个大的模块，以下来一一对其进行介绍。

2.1 数据分析

数据分析可能涉及以下方面：

- ◆ 分析特征变量的分布
 - ◇ 特征变量为连续值：如果为长尾分布并且考虑使用线性模型，可以对变量进行幂变换或者对数变换。
 - ◇ 特征变量为离散值：观察每个离散值的频率分布，对于频次较低的特征，可以考虑统一编码为“其他”类别。
- ◆ 分析目标变量的分布
 - ◇ 目标变量为连续值：查看其值域范围是否较大，如果较大，可以考虑对其进行对数变换，并以变换后的值作为新的目标变量进行建模（在这种情况下，需要对预测结果进行逆变换）。一般情况下，可以对连续变量进行Box-Cox变换。通过变换可以使得模型更好的优化，通常也会带来效果上的提升。

◇ 目标变量为离散值：如果数据分布不平衡，考虑是否需要上采样/下采样；如果目标变量在某个ID上面分布不平衡，在划分本地训练集和验证集的时候，需要考虑分层采样（Stratified Sampling）。

◆ 分析**变量之间两两的分布和相关度

◇ 可以用于发现高相关和共线性的特征。

通过对数据进行探索性分析（甚至有些情况下需要肉眼观察样本），还可以有助于启发数据清洗和特征抽取，譬如缺失值和异常值的处理，文本数据是否需要拼写纠正等。

2.2 数据清洗

数据清洗是指对提供的原始数据进行一定的加工，使得其方便后续的特征抽取。其与特征抽取的界限有时也没有那么明确。常用的数据清洗一般包括：

◆ 数据的拼接

◇ 提供的数据散落在多个文件，需要根据相应的键值进行数据的拼接。

◆ 特征缺失值的处理

◇ 特征值为连续值：按不同的分布类型对缺失值进行补全：偏正态分布，使用均值代替，可以保持数据的均值；偏长尾分布，使用中值代替，避免受 outlier 的影响；

◇ 特征值为离散值：使用众数代替。

◆ 文本数据的清洗

◇ 在比赛当中，如果数据包含文本，往往需要进行大量的数据清洗工作。如去除HTML 标签，分词，拼写纠正, 同义词替换，去除停词，抽词干，数字和单位格式统一等。

2.3 特征工程

有一种说法是，特征决定了效果的上限，而不同模型只是以不同的方式或不同的程度来逼近这个上限。这样来看，好的特征输入对于模型的效果至关重要，正所谓“Garbage in, garbage out”。要做好特征工程，往往跟领域知识和对问题的理解程度有很大的关系，也跟一个人的经验相关。特征工程的做法也是Case by Case，下面就一些点，谈谈自己的一些看法。

2.3.1 特征变换

主要针对一些长尾分布的特征，需要进行幂变换或者对数变换，使得模型（LR或者DNN）能更好的优化。需要注意的是，Random Forest 和 GBDT 等模型对单调的函数变换不敏感。其原因在于树模型在求解分裂点的时候，只考虑排序分位点。

2.3.2 特征编码

对于离散的类别特征，往往需要进行必要的特征转换/编码才能将其作为特征输入到模型中。常用的编码方式有 LabelEncoder, OneHotEncoder（sklearn里面的接口）。譬如对于“性别”这个特征（取值为男性和女性），使用这两种方式可以分别编码为{0,1}和[[1,0], [0,1]]。

对于取值较多（如几十万）的类别特征（ID特征），直接进行OneHotEncoder编码会导致特征矩阵非常巨大，影响模型效果。可以使用如下的方式进行处理：

- ◆ 统计每个取值在样本中出现的频率，取 Top N 的取值进行 One-hot 编码，剩下的类别分到“其他”类目下，其中 N 需要根据模型效果进行调优；
- ◆ 统计每个 ID 特征的一些统计量（譬如历史平均点击率，历史平均浏览率）等代替该 ID 取值作为特征，具体可以参考 Avazu 点击率预估比赛第二名的获奖方案；
- ◆ 参考 word2vec 的方式，将每个类别特征的取值映射到一个连续的向量，对这个向量进行初始化，跟模型一起训练。训练结束后，可以同时得到每个ID的Embedding。具体的使用方式，可以参考 Rossmann 销量预估竞赛第三名的获奖方案，<https://github.com/entron/entity-embedding-rossmann>。

对于 Random Forest 和 GBDT 等模型，如果类别特征存在较多的取值，可以直接使用 LabelEncoder 后的结果作为特征。

2.4 模型训练和验证

2.4.1 模型选择

在处理好特征后，我们可以进行模型的训练和验证。

- ◆ 对于稀疏型特征（如文本特征，One-hot的ID类特征），我们一般使用线性模型，譬如 Linear Regression 或者 Logistic Regression。Random Forest 和 GBDT 等树模型不太适用于稀疏的特征，但可以先对特征进行降维（如PCA，SVD/LSA等），再使用这些特征。稀疏特征直接输入 DNN 会导致网络 weight 较多，不利于优化，也可以考虑先降维，或者对 ID 类特征使用 Embedding 的方式；
- ◆ 对于稠密型特征，推荐使用 XGBoost 进行建模，简单易用效果好；
- ◆ 数据中既有稀疏特征，又有稠密特征，可以考虑使用线性模型对稀疏特征进行建模，将其输出与稠密特征一起再输入 XGBoost/DNN 建模，具体可以参考2.5.2节 Stacking 部分。

2.4.2 调参和模型验证

对于选定的特征和模型，我们往往还需要对模型进行超参数的调优，才能获得比较理想的效果。调参一般可以概括为以下三个步骤：

1. 训练集和验证集的划分。根据比赛提供的训练集和测试集，模拟其划分方式对训练集进行划分为本地训练集和本地验证集。划分的方式视具体比赛和数据而定，常用的方式有：
 - a) 随机划分：譬如随机采样 70% 作为训练集，剩余的 30% 作为测试集。在这种情况下，本地可以采用 KFold 或者 Stratified KFold 的方法来构造训练集和验证集。
 - b) 按时间划分：一般对应于时序序列数据，譬如取前 7 天数据作为训练集，后 1 天数据作为测试集。这种情况下，划分本地训练集和验证集也需要按时间先后划分。常见的错误方式是随机划分，这种划分方式可能会导致模型效果被高估。
 - c) 按某些规则划分：在 HomeDepot 搜索相关性比赛中，训练集和测试集中的 Query 集合并非完全重合，两者只有部分交集。而在另外一个相似的比赛（CrowdFlower 搜索相关性比赛），训练集和测试集具有完全一致的 Query 集合。对于 HomeDepot 这个比赛中，训练集和验证集数据的划分，需要考虑 Query 集合并非完全重合这个情况，其中的一种方法可以参考第三名的获奖方案，https://github.com/ChenglongChen/Kaggle_HomeDepot。
1. 指定参数空间。在指定参数空间的时候，需要对模型参数以及其如何影响模型的效果有一定的了解，才能指定出合理的参数空间。譬如DNN或者XGBoost中学习率这个参数，一般就选 0.01 左右就 OK 了（太大可能会导致优化算法错过最优化点，太小导致优化收敛过

慢)。再如 Random Forest, 一般设定树的棵数范围为 100~200 就能有不错的效果, 当然也有人固定数棵数为 500, 然后只调整其他的超参数。

2. 按照一定的方法进行参数搜索。常用的参数搜索方法有, Grid Search, Random Search 以及一些自动化的方法 (如 Hyperopt)。其中, Hyperopt 的方法, 根据历史已经评估过的参数组合的效果, 来推测本次评估使用哪个参数组合更有可能获得更好的效果。有关这些方法的介绍和对比, 可以参考文献 [2]。

2.4.3 适当利用 Public LB 的反馈

在2.4.2节中我们提到本地验证 (Local Validation) 结果, 当将预测结果提交到 Kaggle 上时, 我们还会接收到 Public LB 的反馈结果。如果这两个结果的变化趋势是一致的, 如 Local Validation 有提升, Public LB 也有提升, 我们可以借助 Local Validation 的变化来感知模型的演进情况, 而无需靠大量的 Submission。如果两者的变化趋势不一致, 需要考虑2.4.2节中提及的本地训练集和验证集的划分方式, 是否跟训练集和测试集的划分方式一致。

另外, 在以下一些情况下, 往往 Public LB 反馈亦会提供有用信息, 适当地使用这些反馈也许会给你带来优势。如图4所示, (a)和(b)表示数据与时间没有明显的关系 (如图像分类), (c)和(d)表示数据随时间变化 (如销量预估中的时序序列)。(a)和(b)的区别在于, 训练集样本数相对于 Public LB 的量级大小, 其中(a)中训练集样本数远超于 Public LB 的样本数, 这种情况下基于训练集的 Local Validation 更可靠; 而(b)中, 训练集数目与 Public LB 相当, 这种情况下, 可以结合 Public LB 的反馈来指导模型的选择。一种融合的方式是根据 Local Validation 和 Public LB 的样本数目, 按比例进行加权。譬如评估标准为正确率, Local Validation 的样本数为 N_l , 正确率为 A_l ; Public LB 的样本数为 N_p , 正确率为 A_p 。则可以使用融合后的指标: $(N_l * A_l + N_p * A_p) / (N_l + N_p)$, 来进行模型的筛选。对于(c)和(d), 由于数据分布跟时间相关, 很有必要使用 Public LB 的反馈来进行模型的选择, 尤其对于(c)图所示的情况。

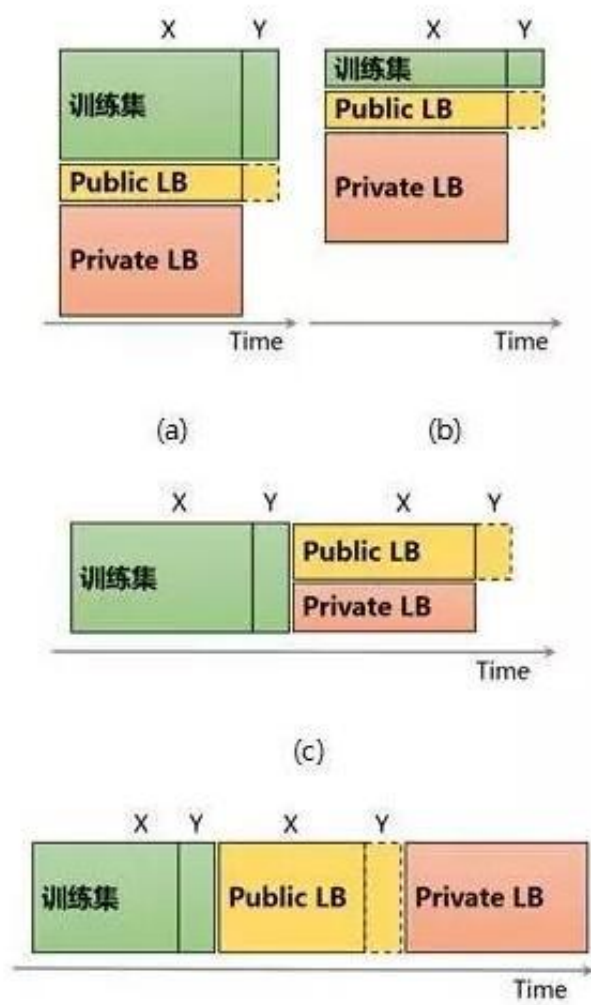


图4. 适当利用 Public LB 的反馈

(图参考 Owenzhang 的分享 [1])

2.5 模型集成

如果想在比赛中获得名次，几乎都要进行模型集成（组队也是一种模型集成）。关于模型集成的介绍，已经有比较好的博文了，可以参考 [3]。在这里，我简单介绍下常用的方法，以及个人的一些经验。

2.5.1 Averaging 和 Voting

直接对多个模型的预测结果求平均或者投票。对于目标变量为连续值的任务，使用平均；对于目标变量为离散值的任务，使用投票的方式。

2.5.2 Stacking

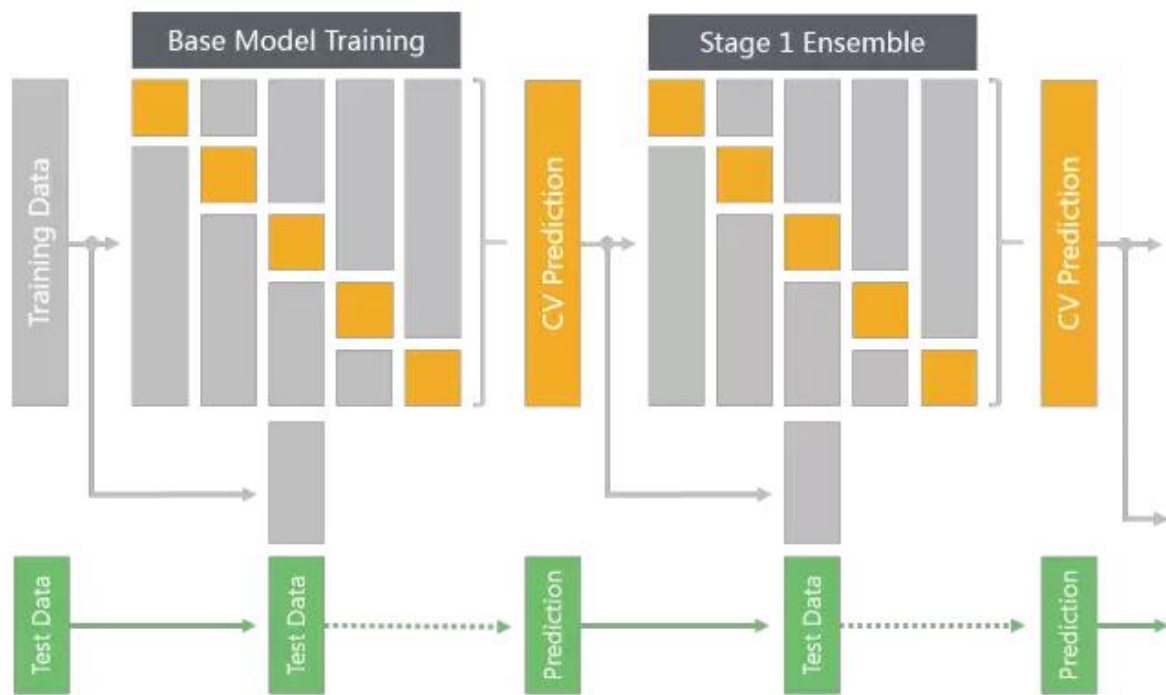


图5. 5-Fold Stacking

(图参考 Jeong-Yoon Lee 的分享 [4])

图5展示了使用 5-Fold 进行一次 Stacking 的过程（当然在其上可以再叠加 Stage 2, Stage 3 等）。其主要的步骤如下：

1. 数据集划分。将训练数据按照5-Fold进行划分（如果数据跟时间有关，需要按时间划分，更一般的划分方式请参考3.4.2节，这里不再赘述）；
 1. 基础模型训练 I（如图5第一行左半部分所示）。按照交叉验证（Cross Validation）的方法，在训练集（Training Fold）上面训练模型（如图灰色部分所示），并在验证集（Validation Fold）上面做预测，得到预测结果（如图黄色部分所示）。最后综合得到整个训练集上面的预测结果（如图第一个黄色部分的CV Prediction所示）。
 2. 基础模型训练 II（如图5第二和三行左半部分所示）。在全量的训练集上训练模型（如图第二行灰色部分所示），并在测试集上面做预测，得到预测结果（如图第三行虚线后绿色部分所示）。
 3. Stage 1 模型集成训练 I（如图5第一行右半部分所示）。将步骤 2 中得到的 CV Prediction 当作新的训练集，按照步骤 2 可以得到 Stage 1模型集成的 CV Prediction。
 4. Stage 1 模型集成训练 II（如图5第二和三行右半部分所示）。将步骤 2 中得到的 CV Prediction 当作新的训练集和步骤 3 中得到的 Prediction 当作新的测试集，按照步骤 3 可以得到 Stage 1 模型集成的测试集 Prediction。此为 Stage 1 的输出，可以提交至 Kaggle 验证其效果。

在图5中，基础模型只展示了一个，而实际应用中，基础模型可以多种多样，如SVM，DNN，XGBoost 等。也可以相同的模型，不同的参数，或者不同的样本权重。重复4和5两个步骤，可以相继叠加 Stage 2, Stage 3 等模型。

2.5.3 Blending

Blending 与 Stacking 类似，但单独留出一部分数据（如 20%）用于训练 Stage X 模型。

2.5.4 Bagging Ensemble Selection

Bagging Ensemble Selection [5] 是我在 CrowdFlower 搜索相关性比赛中使用的方法，其主要的优点在于可以以优化任意的指标来进行模型集成。这些指标可以是可导的（如 LogLoss 等）和不可导的（如正确率，AUC，Quadratic Weighted Kappa等）。它是一个前向贪婪算法，存在过拟合的可能性，作者在文献 [5] 中提出了一系列的方法（如 Bagging）来降低这种风险，稳定集成模型的性能。使用这个方法，需要有成百上千的基础模型。为此，在 CrowdFlower 的比赛中，我把在调参过程中所有的中间模型以及相应的预测结果保留下来，作为基础模型。这样做的好处是，不仅仅能够找到最优的单模型（Best Single Model），而且所有的中间模型还可以参与模型集成，进一步提升效果。

2.6 自动化框架

从上面的介绍可以看到，做一个数据挖掘比赛涉及到的模块非常多，若有一个较自动化的框架会使得整个过程更加的高效。在 CrowdFlower 比赛较前期，我对整个项目的代码架构进行了重构，抽象出来特征工程，模型调参和验证，以及模型集成等三大模块，极大的提高了尝试新特征，新模型的效率，也是我最终能斩获名次的一个有利因素。这份代码开源在 Github 上面，目前是 Github 有关 Kaggle 竞赛解决方案的 Most Stars，地址：https://github.com/ChenglongChen/Kaggle_CrowdFlower。

其主要包含以下部分：

\1. 模块化特征工程

- a) 接口统一，只需写少量的代码就能够生成新的特征；
- b) 自动将单独的特征拼接成特征矩阵。

\2. 自动化模型调参和验证

- a) 自定义训练集和验证集的划分方法；
- b) 使用 Grid Search / Hyperopt 等方法，对特定的模型在指定的参数空间进行调优，并记录最佳的模型参数以及相应的性能。

\3. 自动化模型集成

- a) 对于指定的基础模型，按照一定的方法（如Averaging/Stacking/Blending 等）生成集成模型。

3. Kaggle竞赛方案盘点

到目前为止，Kaggle 平台上面已经举办了大大小小不同的赛事，覆盖图像分类，销量预估，搜索相关性，点击率预估等应用场景。在不少的比赛中，获胜者都会把自己的方案开源出来，并且非常乐于分享比赛经验和技巧心得。这些开源方案和经验分享对于广大的新手和老手来说，是入门和进阶非常好的参考资料。以下笔者结合自身的背景和兴趣，对不同场景的竞赛开源方案作一个简单的盘点，总结其常用的方法和工具，以期启发思路。

3.1 图像分类

3.1.1 任务名称

National Data Science Bowl

3.1.2 任务详情

随着深度学习在视觉图像领域获得巨大成功，Kaggle 上面出现了越来越多跟视觉图像相关的比赛。这些比赛的发布吸引了众多参赛选手，探索基于深度学习的方法来解决垂直领域的图像问题。NDSB就是其中一个比较早期的图像分类相关的比赛。这个比赛的目标是利用提供的大量的海洋浮游生物的二值图像，通过构建模型，从而实现自动分类。

3.1.3 获奖方案

- 1st place: Cyclic Pooling + Rolling Feature Maps + Unsupervised and Semi-Supervised Approaches。值得一提的是，这个队伍的主力队员也是Galaxy Zoo行星图像分类比赛的第一名，其也是Theano中基于FFT的Fast Conv的开发者。在两次比赛中，使用的都是 Theano，而且用的非常溜。方案链接：<http://benanne.github.io/2015/03/17/plankton.html>
- 2nd place: Deep CNN designing theory + VGG-like model + RReLU。这个队伍阵容也相当强大，有前MSRA 的研究员Xudong Cao，还有大神Tianqi Chen，Naiyan Wang，Bing XU等。Tianqi 等大神当时使用的是 CXXNet（MXNet 的前身），也在这个比赛中进行了推广。Tianqi 大神另外一个大名鼎鼎的作品就是 XGBoost，现在 Kaggle 上面几乎每场比赛的 Top 10 队伍都会使用。方案链接：<https://www.kaggle.com/c/datasciencebowl/discussion/13166>
- 17th place: Realtime data augmentation + BN + PReLU。方案链接：<https://github.com/ChenglongChen/caffe-windows>

3.1.4 常用工具

- ▲ Theano: <http://deeplearning.net/software/theano/>
- ▲ Keras: <https://keras.io/>
- ▲ Cuda-convnet2: <https://github.com/akrizhevsky/cuda-convnet2>
- ▲ Caffe: <http://caffe.berkeleyvision.org/>
- ▲ CXXNET: <https://github.com/dmlc/cxxnet>
- ▲ MXNet: <https://github.com/dmlc/mxnet>
- ▲ PaddlePaddle: <http://www.paddlepaddle.org/cn/index.html>

3.2 销量预估

3.2.1 任务名称

Walmart Recruiting - Store Sales Forecasting

3.2.2 任务详情

Walmart 提供 2010-02-05 到 2012-11-01 期间的周销售记录作为训练数据，需要参赛选手建立模型预测 2012-11-02 到 2013-07-26 周销售量。比赛提供的特征数据包含：Store ID, Department ID, CPI, 气温, 汽油价格, 失业率, 是否节假日等。

3.2.3 获奖方案

- 1st place: Time series forecasting method: stlf + arima + ets。主要是基于时序序列的统计方法，大量使用了 Rob J Hyndman 的 forecast R 包。方案链接：<https://www.kaggle.com/c/walmart-recruiting-store-sales-forecasting/discussion/8125>
- 2nd place: Time series forecasting + ML: arima + RF + LR + PCR。时序序列的统计方法+传统机器学习方法的混合；方案链接：<https://www.kaggle.com/c/walmart-recruiting-store-sales-forecasting/discussion/8023>
- 16th place: Feature engineering + GBM。方案链接：https://github.com/ChenglongChen/Kaggle_Walmart-Recruiting-Store-Sales-Forecasting

3.2.4 常用工具

- ▲ R forecast package: <https://cran.r-project.org/web/packages/forecast/index.html>
- ▲ R GBM package: <https://cran.r-project.org/web/packages/gbm/index.html>

3.3 搜索相关性

3.3.1 任务名称

CrowdFlower Search Results Relevance

3.3.2 任务详情

比赛要求选手利用约几万个 (query, title, description) 元组的数据作为训练样本，构建模型预测其相关性打分 {1, 2, 3, 4}。比赛提供了 query, title和description的原始文本数据。比赛使用 Quadratic Weighted Kappa 作为评估标准，使得该任务有别于常见的回归和分类任务。

3.3.3 获奖方案

- 1st place: Data Cleaning + Feature Engineering + Base Model + Ensemble。对原始文本数据进行清洗后，提取了属性特征，距离特征和基于分组的统计特征等大量的特征，使用了不同的目标函数训练不同的模型（回归，分类，排序等），最后使用模型集成的方法对不同模型的预测结果进行融合。方案链接：https://github.com/ChenglongChen/Kaggle_CrowdFlower
- 2nd place: A Similar Workflow
- 3rd place: A Similar Workflow

3.3.4 常用工具

- ▲ NLTK: <http://www.nltk.org/>
- ▲ Gensim: <https://radimrehurek.com/gensim/>
- ▲ XGBoost: <https://github.com/dmlc/xgboost>
- ▲ RGF: https://github.com/baidu/fast_rgf

3.4 点击率预估

3.4.1 任务名称

3.4.2 任务详情

经典的点击率预估比赛。该比赛中提供了7天的训练数据，1 天的测试数据。其中有13 个整数特征，26 个类别特征，均脱敏，因此无法知道具体特征含义。

3.4.3 获奖方案

- 1st place: GBDT 特征编码 + FFM。台大的队伍，借鉴了Facebook的方案 [6]，使用 GBDT 对特征进行编码，然后将编码后的特征以及其他特征输入到 Field-aware Factorization Machine (FFM) 中进行建模。方案链接：<https://www.kaggle.com/c/criteo-display-ad-challenge/discussion/10555>
- 3rd place: Quadratic Feature Generation + FTRL。传统特征工程和 FTRL 线性模型的结合。方案链接：<https://www.kaggle.com/c/criteo-display-ad-challenge/discussion/10534>
- 4th place: Feature Engineering + Sparse DNN

3.4.4 常用工具

- ▲ Vowpal Wabbit: https://github.com/JohnLangford/vowpal_wabbit
- ▲ XGBoost: <https://github.com/dmlc/xgboost>
- ▲ LIBFFM: <http://www.csie.ntu.edu.tw/~r01922136/libffm/>

3.5 点击率预估 II

3.5.1 任务名称

Avazu Click-Through Rate Prediction

3.5.2 任务详情

点击率预估比赛。提供了 10 天的训练数据，1 天的测试数据，并且提供时间，banner 位置，site, app, device 特征等，8个脱敏类别特征。

3.5.3 获奖方案

- 1st place: Feature Engineering + FFM + Ensemble。还是台大的队伍，这次比赛，他们大量使用了 FFM，并只基于 FFM 进行集成。方案链接：<https://www.kaggle.com/c/avazu-ctr-prediction/discussion/12608>
- 2nd place: Feature Engineering + GBDT 特征编码 + FFM + Blending。Owenzhang（曾经长时间雄霸 Kaggle 排行榜第一）的竞赛方案。Owenzhang 的特征工程做得非常有参考价值。方案链接：<https://github.com/owenzhang/kaggle-avazu>

3.5.4 常用工具

- ▲ LIBFFM: <http://www.csie.ntu.edu.tw/~r01922136/libffm/>
- ▲ XGBoost: <https://github.com/dmlc/xgboost>

4. 参考资料

-
- [1] Owenzhang 的分享: Tips for Data Science Competitions
 - [2] Algorithms for Hyper-Parameter Optimization
 - [3] MLWave博客: Kaggle Ensembling Guide
 - [4] Jeong-Yoon Lee 的分享: Winning Data Science Competitions
 - [5] Ensemble Selection from Libraries of Models
 - [6] Practical Lessons from Predicting Clicks on Ads at Facebook