http://www.chioka.in/kaggle-competition-solutions/

http://mlwave.com/kaggle-ensembling-guide/

<http://blog.csdn.net/a353833082/article/details/50768140>

<http://www.ihref.com/read-16369.html>

从零开始的机器学习竞赛问答——Kesci交流群经验分享

我们都是从零开始接触机器学习的，在刚开始实践的过程中也会遇到许许多多问题。很多人都会从中遇到很多数据处理、模型、算法等等困惑。在这篇帖子中我把在Kesci官方群里交流过程中所做的问答汇总出来变成这样一个帖子。因为很多问题是我和大家当初学习遇到相同的困惑，然后我一点点遇到的解决办法，所以在解答的时候，我也尽量不给知识添什么门槛。我个人最多只是比完全零基础的朋友们先学一步，再加上一些有限经验的探索总结，所以这里的问答很多只是经验的交流探讨，为新入门的机器学习爱好者提供一些便于理解而可行的办法，让先学两步及以上的大牛们见笑了。因为是问答，所以话题比较散，很多地方解释得也不够系统和明确，有不准确和错误的地方还望纠正。在此也感谢主办方提供这样一个宝贵交流的平台，以及各位一同参加学习比赛的朋友们彼此的支持分享，愿主基督耶稣的恩惠和祝福与大家同在！

**I. 机器学习流程**

**Q: 各种机器学习的书和原理看了很多，仍不知道实战怎么做，兄弟能否带我入门一次，哪怕列个提纲都好：比如1. 用\*\*算法进行\*\*清洗，达到什么目的。2.用\*\*算法进行\*\*\*，达到什么目的**

A: 我把数据处理预测分为五步。这五步也可以很快速地通用到当前很多数据预测问题中

1. 数据摘要。在样本量非常大的前提下，我们可以通过变量的特征快速了解数据结构。由于大数据中变量通常成百上千，我们不妨通过循环或函数的应用，把数据中的各个变量统一做成一个表格。每行是一个变量，各列的内容有：
   1. 每个变量的类型（数值、类别、时间、……）
   2. 每个变量的非空值数量，该变量频数最高的几大类别和相应的频数统计（比如我用前5大类别）
   3. 每个数值变量的简单统计量，比如均值、方差、四分位数、最小值、最大值
2. 清洗变量。目的是把所有变量在保留尽量多的有效信息的前提下变成数值型用于建模。具体方法是：
   1. 保留所有数值变量。
   2. 把类别变量全部通过One-Hot Encoding变成0-1变量，此时所有变量都变成数值型的了。
   3. 删掉那些绝大多数值为NA或同一值（比如49990个0，10个其他值）的变量，目的是尽可能去除包含有效信息很少的变量冗余
   4. 用平均数（或中位数等）填充NA，如果需要的话再把数据标准化。这样就把数据洗好了，接下来可以拿来建模。
3. 拆分训练集和验证集。目的是为训练模型和交叉验证做准备。具体方法（以10-folds交叉验证为例）：
   1. 把数据按列分成x和y（预测目标列为y，其他的列为x）。
   2. 把样本行的index随机拆成10份保存起来，每次取1份做验证集index，另外9份粘起来做训练集index。这样通过index就可以取到x和y的训练集和验证集了
4. 训练模型。目的是得到预测所用的模型，并在本地得到比较准确的结果范围估计，避免对单个模型和数据集的过度拟合。具体方法：
   1. 先取一份训练集和预测集index，把取到的x和y的训练集在模型上训练，得到的模型保存起来，然后在验证集x上预测y，和真实的验证集y比较，得到分数。
   2. 照着之前拆分出来的index做10次，每次取不同的份做验证，剩下的9份粘起来做训练。这样就得到了10个模型和10个验证分数。十个验证分数平均大致就是预测分数的水平
5. 预测结果。最后把10个模型都预测那个预测集，得到了10个预测结果y，然后预测的结果取平均提交上去就OK了。其中通过交叉验证的模型来取平均主要是想通过模型的差异来抵消一部分模型本身的系统性误差，减小过拟合，增加总体稳健性。

**II. 分数估计**

**Q: 为什么比赛要用AUC而不是0-1正确率呢？AUC的意义是什么？**

A: 是这样，这个数据集上标签非常不平衡，如果做全0预测的话，训练集的0-1准确率可以达到92.7%，但是这样的预测没有任何实际意义，我们大概也不希望被这样的结果比下去。所以其实auc是在做一个权衡，目的是当从中一个阈值的时候，我们发现预测高于这个阈值的更有可能真值为1，而预测低于这个阈值真值更有可能为0，我们来算这个模型预测中真为1和假冒为1的概率，从此我们就能知道这个模型是不是真的有效，还是我说的全零碰运气的。AUC的目的就是不管取什么阈值，在尽量少地预测假冒为1的前提下，模型尽可能把真为1的找出来，这样这个模型就有高的AUC得分

**Q: 我做了样本平衡，结果验证的AUC很好，比排行榜上的高多了？**

A: 验证集上必须使用原样本而不是平衡后的样本。AUC的结果会受到样本标签分布的影响，本来不平衡的0-1，如果抽样平衡，变成一半是海水，一半是陆地，很容易调整海平面来把海洋和陆地分开。但是如果是十倍的海水把岛屿包围，原先的分界就变模糊了，一调海平面就容易把岛屿淹没或使岛屿变得很大，而这才是实际情况的困难所在，所以在验证集上平衡样本会把问题简化，造成比较严重的估计偏差。

**III. 模型选取**

**1. 关于XGBoost**

**Q: 在这个比赛做预测的话，用什么模型效果好？**

A: 针对非线性的数据预测问题，目前从预测精度、稳健性、适用范围和速度都首选XGBoost，原理上可以从机器学习中的决策树入门，然后参考Random Forest和Gradient Boosted Tree；技术上可以减小步长eta来更精细地逼近，减小每步分支选择变量的比例colsample\_bylevel来增加子模型的多样性

**Q: XGBoost为什么有优势，选别的模型比如logistic regression可以吗？**

A: XGBoost主要强项在于对这类独立样本、高维互不直接关联的变量的问题预测效果好，因为问题给的是这样、比分的标准设定在那里，所以就这样选了。但是logistic回归的简洁和可解释性，神经网络对于序列和图像的识别预测是它们各自的优势领域。此外神经网络在这个比赛中相比XGBoost虽然不容易达到最优，但是是可以用作辅助改善预测效果的

**Q: 能解释XGBoost一下步长的意义吗？**

A: XGB是一种梯度优化的决策树模型(Gradient Boosted Trees)，其中步长(eta, learning rate) 是决策树每次拟合的比例。相当于我们如果用石头刻一个雕像，每棵树相当于凿掉一些石头，然后继续雕刻剩下的石材，直到把雕像里的形象显露出来。步长(eta, learning rate)决定了大斧头砍还是小凿子凿，下刀过分了就是过拟合；大斧头雕刻起来速度快，只要很少几刀就能雕刻得差不多，但是结果也比较粗糙，小凿子凿的慢，要用很多刀才能雕好，但是可以做得很精细

**Q: 那么比赛过程中步长应该是越小越好吗？**

A: 步长提交最后一次再取到极限吧，步长太小除了最后一次更新leaderboard之外用途不大，容易拖慢其他调参的进度

**Q: 话说训练XGBoost需要对自变量的分布进行标准化吗？**

A: 话说xgboost好像是无视自变量的分布的，只要大小的排序不改变。我试了一下不管标准化，转成rank，甚至转成正态分布都不影响训练结果

**Q: 有XGBoost的调参经验可以分享吗？**

A: XGBoost的调参推荐一下这篇文章，介绍得还是比较全面的：<http://www.analyticsvidhya.com/blog/2016/03/complete-guide-parameter-tuning-XGBoost-with-codes-python/>

**2. 关于神经网络及Python的Keras包**

**Q: Keras参数太多了，该如何调参呢？**

A: 我也不是做神经网络的，所以认识比较浅，主要只能在学习和别人经验的基础上慢慢理解。Keras的参数和背后涉及的神经网络原理太多，自己理解还很难达到特别简明地讲清要点的程度。建议从了解神经网络的原理入手Keras，这两篇文集还是不错的。两位作者也是从学习者的角度写的，但是核心内容蛮全的。   
\* 关于深度学习：<http://blog.csdn.net/zouxy09/article/category/1387932>   
\* 关于Keras包：<http://blog.csdn.net/niuwei22007/article/category/5868745>

**Q: 用神经网络要怎样处理数据呢？**

A: 要注意用keras的一个关键点是除了模型的调参之外，需要把数据标准化为正态分布，而且要去除冗余特征，这样会非常明显有利于训练

**Q: 怎样把数据转成正态分布呢？用Box-cox吗？**

A: Box-cox是通过多项式和对数变换，来对整体数据的中心进行修正，如果要对数据彻底转换，那么可以直接做rank然后复合正态分布的百分位函数

**Q: 深度学习的效果怎么样？**

A: 深度学习的深度是依赖于数据量的，不能在小河里开轮船，这个数据量只能做浅度学习/呲牙

**3. 算法、模型融合和其他调参**

**Q: CV和Bootstrap的区别和联系是什么？**

A: 交叉验证(CV)是每次不重复地取样本的一部分做验证集，剩下的训练集，直到验证集把样本覆盖掉，比如对6个面的骰子取6-folds交叉验证，就是一次取1个面验证，5个面训练，直到验证集把6个面都取光，好处是验证集之间互不包含，而且完备地覆盖样本。Bootstrap，名字含义是用鞋拔子把自己提起来。。。原理是通过对旧有样本不断地放回抽样生成一组新样本，这个样本可以与原来大小一样，也可以是任意大小，比如同样一个6个面的骰子，我可以扔6次，然后把6次的结果做成一个新骰子（但是奇怪，这六个面怎么好像有重复的），进而可以扔100个6次，做成100个新骰子，或者甚至干脆扔100次，做成一个100面的骰子，这就是Bootstrap，好处是可以任意次数地抽取任意大小的数据集，坏处就是每组样本的独立性未必有交叉验证那么好，而且总觉得哪里怪怪的。

**Q: Gradient Boosted Trees的Random Forest之间的区别和联系是什么？为什么Random Forest可以和xgboost融合改进结果呢？**

A: Random Forest的原理每棵树都是在原数据的某个子变量集上做独立的决策，各树随机挑选的自变量集是为了增加树的多样性，相当于不同领域的专家各自投票，最后平均；Gradient Boosted Trees是每棵树在上一棵树的残差的基础上继续回归改进，相当于一个专家接着上个专家的成果上改进。xgb主要采用的原理是GBT的残差回归，但是也可以辅助利用RF的子变量集挑选的理念来增加树的多样性，那样就是让不同领域的专家轮流来改进上个专家的成果。但是总的来说xgboost还是属于改进式而不是投票式，所以与投票式方法的Random Forest结合起来是可以起到互补的。

**Q: XGBoost调参怎么做呢？手动调好一个参数再调下一个可以吗？**

A: XGBoost一个一个来应该问题不大，通用地找模型的超参数的话Python的hyperopt包也是个不错的选择，手动调参把模型熟悉了之后可以用这个来化简调参的过程

**Q: 我已经训练了两种不同的模型，该如何把它们合起来预测呢，求平均吗?**

A: 可以加权平均，设两种的模型的权重为w1和w2，通常来说取最优解的时候平均效果好的模型权重比较大，可以通过交叉验证取最优的。但是不用每次调整w1和w2都重新跑模型，因为交叉验证的y已经跑出来了，我们优化的时候只需要用到两个模型的交叉验证的分别的预测y，交叉验证的真实y，然后加权来算分数就好了。Python的hyperopt包是调模型超参数的，功能相当于搜索复杂函数的最优解，好处是效率比单纯的网格搜索或者随机搜索要高，不妨拿它来优化一下加权权重。