参考链接:

<https://tianchi.aliyun.com/notebook-ai/detail?spm=5176.12281978.0.0.68021b43woz7CA&postId=95535>

**模型融合**

对于多种调参完成的模型进行模型融合

融合的分类往往有:

1. 简单加权融合:

回归（分类概率）：算术平均融合（Arithmetic mean），几何平均融合（Geometric mean）；

分类：投票（Voting)

综合：排序融合(Rank averaging)，log融合

1. stacking/blending:

构建多层模型，并利用预测结果再拟合预测。注意stack模型运用初始模型训练的标签再利用真实标签进行再训练，会造成模型的过拟合。这样或许模型在测试集上的泛化能力或者说效果会有一定的下降，因此现在的问题变成了如何降低再训练的过拟合性。因此运用两种方法：

1. 次级模型尽量选择简单的线性模型
2. 利用K折交叉验证
3. boosting/bagging（在xgboost，Adaboost,GBDT中已经用到）:

多树的提升方法

回归\分类概率-融合

1. 简单的加权平均，就是结果的直接融合
2. Mean均值融合，median平均
3. Stacking融合

分类模型的融合：

1. Voting投票机制
2. Stacking\Blending融合

优点：Blending比stacking简单（因为不用进行k次的交叉验证来获得stacker feature）; 避开了一个信息泄露问题：generlizers和stacker使用了不一样的数据集

缺点：使用了很少的数据（第二阶段的blender只使用training set10%的量）

blender可能会过拟合

stacking使用多次的交叉验证会比较稳健

（3）分类的Stacking融合(利用mlxtend)

#### 其它方法：将特征放进模型中预测，并将预测结果变换并作为新的特征加入原有特征中再经过模型预测结果 （Stacking变化）

模型的融合是提分和提升模型鲁棒性的一种重要方法：

1）结果层面的融合，这种是最常见的融合方法，其可行的融合方法也有很多，比如根据结果的得分进行加权融合，还可以做Log，exp处理等。在做结果融合的时候，有一个很重要的条件是模型结果的得分要比较近似，然后结果的差异要比较大，这样的结果融合往往有比较好的效果提升。

2）特征层面的融合，这个层面其实感觉不叫融合，准确说可以叫分割，很多时候如果我们用同种模型训练，可以把特征进行切分给不同的模型，然后在后面进行模型或者结果融合有时也能产生比较好的效果。

3）模型层面的融合，模型层面的融合可能就涉及模型的堆叠和设计，比如加Staking层，部分模型的结果作为特征输入等，这些就需要多实验和思考了，基于模型层面的融合最好不同模型类型要有一定的差异，用同种模型不同的参数的收益一般是比较小的。