**特征工程**

**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 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。

**模型集成**

**假设是五折的stacking，我们有一个train数据集和一个test数据集，那么一个基本的stacking框架会进行如下几个操作：**

1、选择基模型。我们可以有xgboost，lightGBM，RandomForest，SVM，ANN，KNN，LR等等你能想到的各种基本算法模型。

2、把训练集分为不交叉的五份。我们标记为train1到train5。

3、从train1开始作为预测集，使用train2到train5建模，然后预测train1，并保留结果； 然后，以train2作为预测集，使用train1，train3到train5建模，预测train2，并保留结果；如此进行下去，直到把train1到train5各预测一遍；

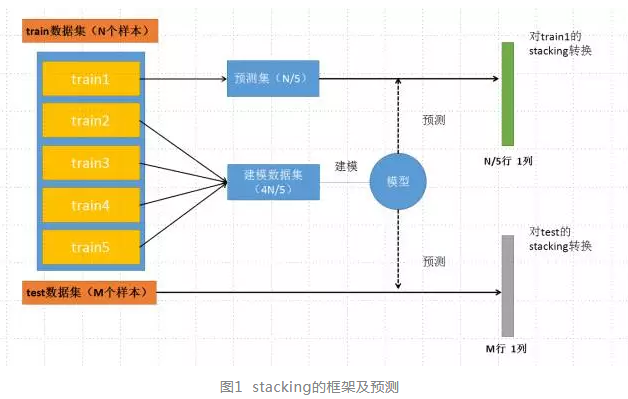
4、把预测的结果按照train1到trian5的位置对应填补上，得到对train整个数据集在第一个基模型的一个stacking转换。

5、在上述建立的五个模型过程中，每个模型分别对test数据集进行预测，并最终保留这五列结果，然后对这五列取平均，作为第一个基模型对test数据的一个stacking转换。

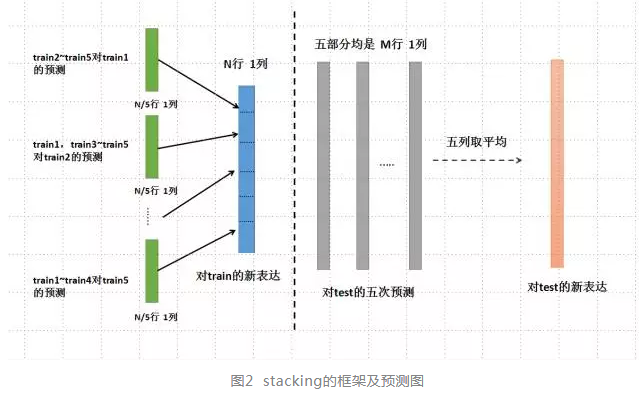
6、选择第二个基模型，重复以上2-5操作，再次得到train整个数据集在第二个基模型的一个stacking转换。

7、以此类推。有几个基模型，就会对整个train数据集生成几列新的特征表达。同样，也会对test有几列新的特征表达。

8、一般使用LR作为第二层的模型进行建模预测。



上面这个框架说明的是：对训练数据进行**无重复**的五次划分之后，分别对其中每一部分进行一次预测，而预测的模型就是由其余四部分训练的；并且在预测了预测集之后，还需要对我们的test数据集也进行一次预测，这这样就会得到5个N/5行、1列的对train数据集的特征转换，和5个M行、1列的对test数据集的特征转换，由此进入下一个图。



这个图说明的是对五部分的train分别预测之后再组成对train的新的表达，由5个5/N行一列的预测组成一个N行1列预测结果，从而得到对train的新表达；同时，对test的结果也进行处理，由于是5个模型对test的预测，所以要取平均，使得五列平均为一列，从而得到对test的新表达。