ReadMe

操作流程

先运行DataProcess.ipynb文件，直接点击全部运行即可，随后再打开Test.ipynb文件，同样点击全部运行即可。在数据处理的缺失值预测这一板块中，我导入了sklearn库中的knn模型进行预测，在提交时已经注释掉，可以通过运行knn而不运行本来的填充方法来进行测试（注意：运行KNN部分时请先重新运行原本的预测模块之上的代码，否则因为数据已被更改会报错）（两种不同的缺失值处理方法的差异会在最后模型预测准确率上反应，初始方法正确率约81%，KNN进行缺失值填补的正确率约为83%）

一、决策树的构建思路

1、决策树将所有的特征看作一个个结点，通过对数据不断进行分类使得每个结点下的样本数据纯度不断提高的模型(这里尤指分类树，不涉及回归树)；这里对于决策树的构建采用字典的键值对型进行构建，以键作为父结点，以值作为子结点；

2、决策树的构建过程中，首先需要对数据进行处理，将只有少量缺失值的特征补全，删除与分类相关性不大或者缺失十分严重的特征，还需将连续型数据或者其他字符型数据转换为用0、1等表示的离散型数据；

3、随后遍历每一种特征分割方式，通过如下的评判标准进行特征的选择：
首先定义信息熵: $$ H(X) = -\sum_{x\in X}P(X) \log P(X) $$ (1) 信息增益 设在分类之前的信息熵为 IntrinsicVal，按照特征D分类后的信息熵为 Info(D) 则： $$ max \quad \text{InfoGain} = \text{IntrinsicVal} - \text{Info(D)} $$

即通过选取能使原数据集熵减最大的特征来划分原数据集

(2) 信息增益率

由于信息增益不好处理分布极其分散的特征，会导致出现很多分支，一定程度上可能让决策树过拟合，且分类效果不好，所以考虑用信息增益率来衡量： $$ \text{InfoRatio} = \frac{\text{InfoGain}}{\text{IntrinsicVal}} $$

(3) 基尼指数
$$ Gini(D) = 1 - \sum_{k=1}^{K} p_{k}^{2} $$

4、根据所选取的特征对原数据集进行划分，并以此特征在决策树上构建新的结点；

5、再对已划分的数据集按照上述流程进行递归建树，注意当达到所规定的树最大高度、所剩余的样本个数未达到继续分支的条件以及所有特征都已被遍历时，按照少数服从多数的原则为该结点定下标签。

二、随机森林的构建思路

1、随机森林通过综合多棵决策树的分类结果进行投票达到更加准确的分类结果；

2、首先通过BootStrap方法创建建立单颗决策树的数据集，BootStrap方法通过从原数据集中进行抽样，抽样过程中允许样本重复出现(有放回的抽取)；

3、再利用创建的BootStrap数据集利用已构建好的决策树建立过程进行单棵决策树的创建；

4、重复上述步骤建立起多棵决策树构建随机森林；

5、利用随机森林预测时通过各个决策树的预测结果进行投票，按照少数服从多数的原则得到预测结果；

三、Adaboost的构建思路

1、基于已经创建的决策树模型，为每个样本点设置初始权重$\frac{1}{N}$(这里设置初始权重是为了后面对权重的迭代更新)，用已知数据训练此分类器，得到一个弱分类器；

2、计算弱分类器在上述权重的数据集下的分类误差(误分类样本的权重之和),记为L_{w}： $$ L_{w} =\sum_{i=1}^{N} P(G_m(x_{i})\not=y_{i}) =\sum_{G_m(x)\not=y_{i}}w_{mi} $$

3、根据上述所计算出的分类误差计算分类器的权重系数： $$ \alpha = \frac{1}{2} \log \frac{1-L_w}{L_w} $$

4、再更新训练数据的权值分布，对分类错误的样本加大其权重，分类正确的不变： $$ L_{w}^{'} = L_{w} \cdot e^{\alpha} $$

5、根据各个弱分类器的权重，将所训练好的弱分类器组合在一起共同决定最终的预测结果。