1. 数据处理

本项目中共有多个数据文件，数据时间的的有效时间为2015年，包括参考类文件：调查问卷原卷happiness\_survey\_cgss2015.pdf、问卷数据的数据目录索引happiness\_index.csv；数据类文件：精简版测试、训练数据集happiness\_test\_abbr.csv、happiness\_train\_abbr.csv；完整版测试、训练数据集：happiness\_test\_complete.csv、happiness\_train\_complete.csv。

参考文件当中一共涉及六部分：核心模块（家庭情况、社会人口属性、健康、移居、生活方式、社会观点、阶层认同、政治态度、认知和劳动、社会保障），十年回顾，EASS（东亚联合社会调查），ISSP（国际社会调查项目），能源使用，法制观点，联系方式。通过对比参考类文件和数据文件可以发现，所给的完整数据集选取了核心模块和部分的十年回顾内容。

数据集本身不规整，有空缺数据和不合理输入。通过pandas、numpy库进行预处理，把所有为空的数据归置为-1，如果该问题列当中很少有人回答则考虑删除该列。

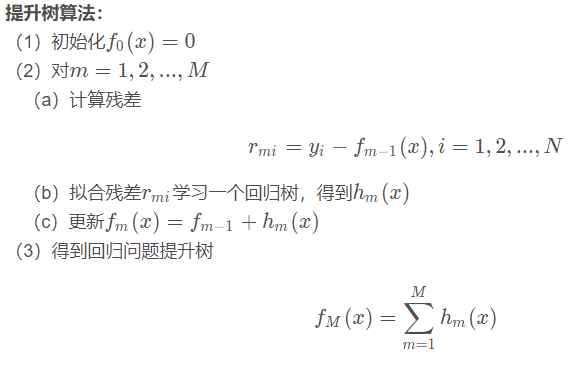
通过直观观察问题可以发现，整体可以划分为以下几个内容：年代、信仰、受教育程度、政治面貌、房产、业余生活、财富等十余个方面。根据该情况，考虑把同一类的重复问题合并，例如把有关于家庭投资的多个问题（股票、基金、活期）等通过问题顺序编制成二进制编码，例如：参与股票，基金不参与活期的样例改为110，再转换为十进制整数6，缩减数据规模，如果有特殊情况的则使用全1或者全1往上的数字进行编码。这样的合并一共有两组，一组是房产问题，一组是投资问题。同时把日期更改为可以进行分析的数据类型

1. Xgb模型
2. GBDT梯度提升树

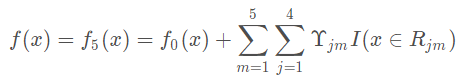
Xgb模型实际上是对GBDT的工业化实现和进一步的改进。全称是Gradient Boosting Decision Tree，其中Gradient Boosting指的是梯度上升算法，用来对分类器进行筛选的方法，Decision Tree决策树指的是CART回归树弱分类器。

GBDT是传统机器学习算法当中对真实分布拟合做好的算法之一，其主要思想是通过基函数的线性组合以及不断减少残差来得到回归预测。【3】通过不断地迭代生成一系列的树，然后对这一系列的树进行加权求和，目的是找到一个CART回归树模型的弱学习器，是的损失函数最小。其中的数学方法不多做介绍。

梯度提升树是提升树的改进算法，主要思想是在迭代中不断地接近拟合目标，减少拟合损失，例如目标是拟合一个距离是50，第一次拟合到30，下一次迭代用10来拟合差距20，这样继续往下，最后每次迭代的结果30+1…之和就是最终的模型输出结果。



对于这样的模型需要的输入为学习率，迭代次数和树的深度，首先初始化弱学习器，把数据残差也就是负梯度——上一轮的学习器差值作为标签值，遍历所有特征可能，找到最佳划分节点，找到总平方损失最小的作为划分。迭代次数结束之后将多棵树进行加总成为最后的强学习器。



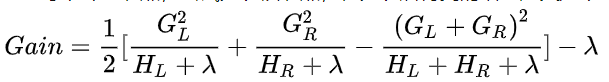
1. CART回归树

CART算法既可以用于创建分类树，也可以用于创建回归树、模型树，两者在建树的过程稍有差异。【1，2】

在建树的过程当中使用基尼系数作为判定标准选择一个作为类别的单位，在划分树的过程当中不断地重复寻找能够得到最好值的特征值作为切分树的标准，一直循环执行直到无法切分。

1. XGB模型【4】

XGB模型的逻辑和GBDT模型是一样的，而XGB使用贪心算法求解树结构，定义了一个增益值，对于每一层都用线性扫描寻找到最佳分裂。



总结算法流程：a.迭代生成树；b.迭代初始对每个样本计算损失函数，这个计算和已经迭代好的树有关；c.使用目标函数采用贪心算法求当前树；d.新增树后，模型更新；e.使用收缩率改变新增树的函数，防止过拟合。

相比GBDT支持线性分类和正则化逻辑回归，通过列抽样有效降低了模型的方差

1. 实验

本次实验当中采用sklearn给出的方法进行实验，首先对于数据进行读取，将训练集和测试集合并处理，然后进行训练，输入参数包括学习率，树数量，树高（GBDT项目），权重系数，惩罚系数，正则化参数，随即特征等等，并且设定随机数种子保证每次的实验结果都是一致的。

参考内容：

1. CART源码：<https://blog.csdn.net/weixin_40479663/article/details/84781500>
2. CART源码解析：<https://zhuanlan.zhihu.com/p/32164933>
3. GBDT算法解析：<https://blog.csdn.net/horizonheart/article/details/78782622>
4. XGB：<https://www.cnblogs.com/bnuvincent/p/9693190.html>