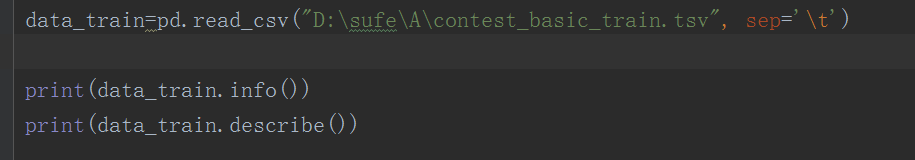
**数据处理与建模部分**

**李帅**

1. **数据预处理**

在对训练集数据处理之前，需要对训练集数据的缺失值和异常值情况进行了解。Python内有describe()函数，可以了解数据集的缺失值、均值和中位数等。



结果显示如下：

|  |  |  |  |
| --- | --- | --- | --- |
| REPORT\_ID | 30000 | non-null | int64 |
| ID\_CARD | 30000 | non-null | object |
| LOAN\_DATE | 30000 | non-null | object |
| AGENT | 8952 | non-null | object |
| IS\_LOCAL | 30000 | non-null | object |
| WORK\_PROVINCE | 27742 | non-null | float64 |
| EDU\_LEVEL | 26942 | non-null | object |
| MARRY\_STATUS | 30000 | non-null | object |
| SALARY | 8864 | non-null | float64 |
| HAS\_FUND | 29998 | non-null | float64 |
| Y | 30000 | non-null | int64 |

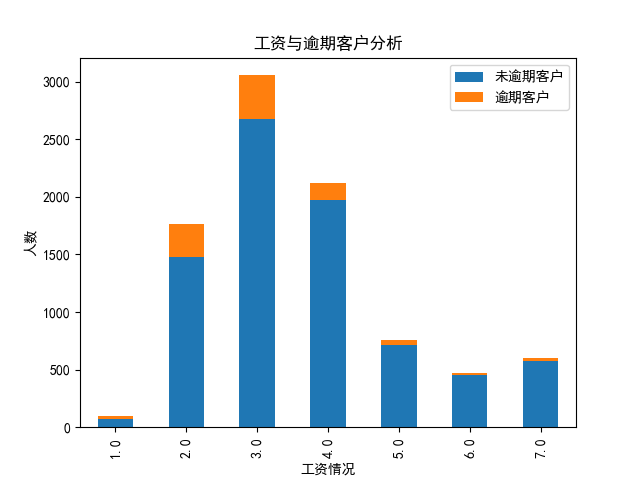
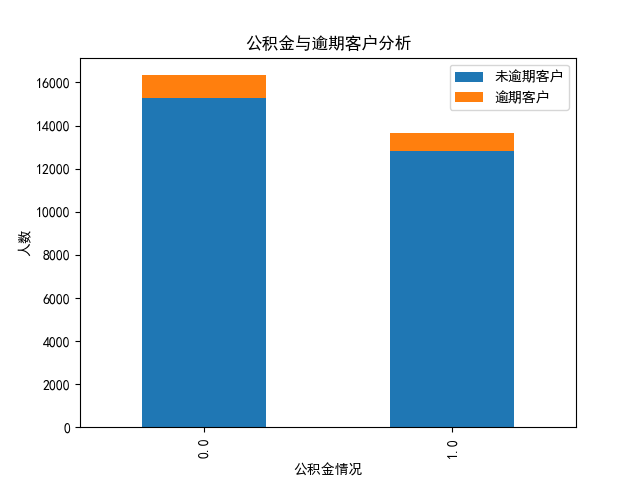
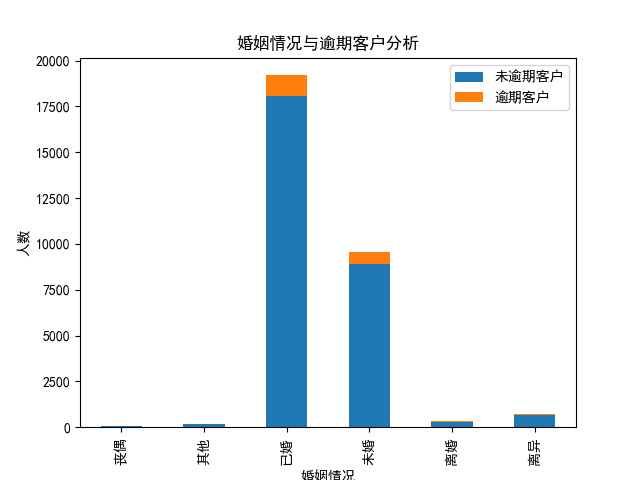
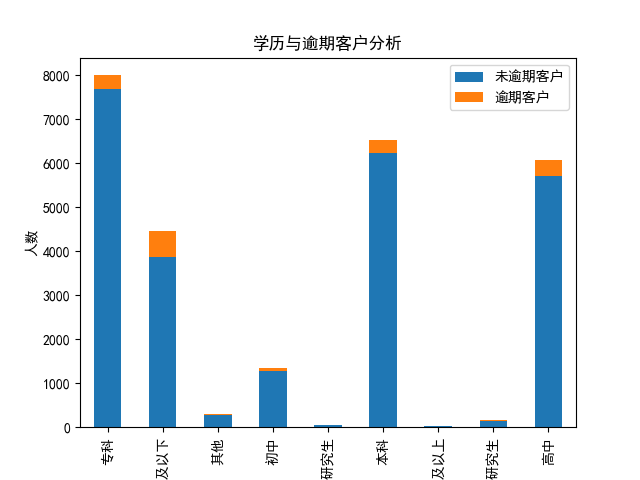
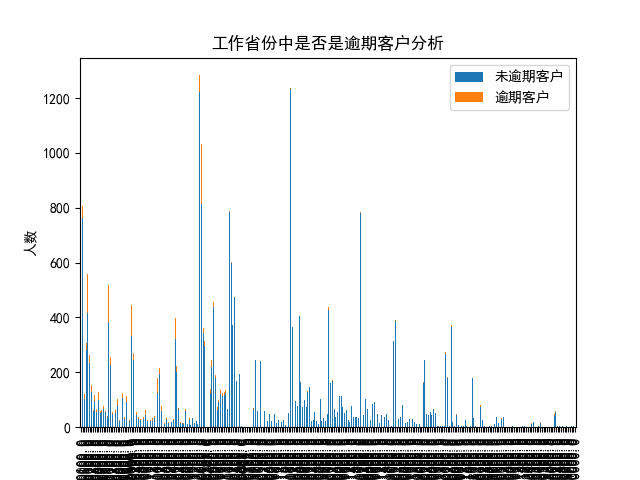
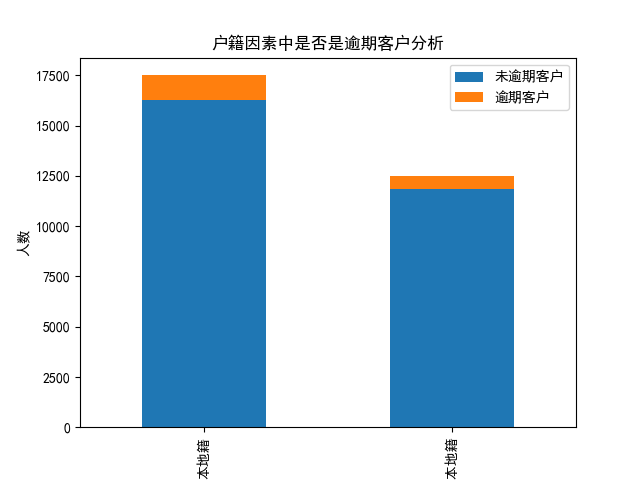
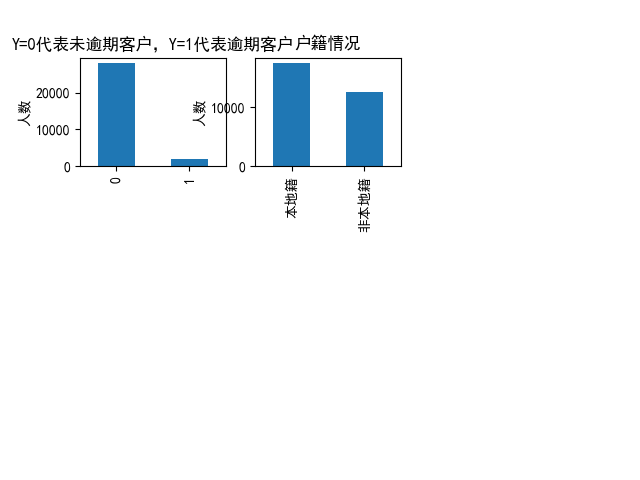
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | REPORT\_ID | WORK\_PROVINCE | SALARY | HAS\_FUND | Y |
| count | 3.000000e+04 | 27742.000000 | 8864.000000 | 29998.000000 | 30000.000000 |
| mean | 3.258910e+06 | 308176.414137 | 3.618682 | 0.455364 | 0.062500 |
| std | 1.563711e+06 | 101940.879912 | 1.415137 | 0.498012 | 0.242065 |
| min | 8.787000e+03 | 110000.000000 | 1.000000 | 0.000000 | 0.000000 |
| 25% | 2.021495e+06 | 230000.000000 | 3.000000 | 0.000000 | 0.000000 |
| 50% | 3.436337e+06 | 330100.000000 | 3.000000 | 0.000000 | 0.000000 |
| 75% | 4.634978e+06 | 370000.000000 | 4.000000 | 1.000000 | 0.000000 |
| max | 5.631528e+06 | 650000.000000 | 7.000000 | 1.000000 | 1.000000 |

由上图可知，train集一共30000行数据，其中AGENT,WORK\_PROVINCE,EDU\_LEVEL,SALAR，HAS\_FUND出现缺失，同时，将表格中的字符串替换成相应的数字，如IS\_LOCAL属性中本地籍设为1，非本地籍设为0，对于学历， 其他0 初中1 高中2 专科及以下3 专科 4 本科5 硕士研究生6 博士研究生7 硕士及以上8，等等如此处理。

1. **数据可视化**

用matplotlib画出训练集各属性与Y值之间的关系，用于初步数据分类。

（压缩包内有图）



1. **缺失值处理**

（随机森林算法（Random forest algorithm）是对 bagging 算法的扩展。除了仍然根据从训练数据样本建立复合模型之外，随机森林对用做构建树（tree）的数据特征做了一定限制，使得生成的决策树之间没有关联，从而提升算法效果。

随机森林算法

决策树运行的每一步都涉及到对数据集中的最优分裂点（best split point）进行贪婪选择（greedy selection）。

这个机制使得决策树在没有被剪枝的情况下易产生较高的方差。整合通过提取训练数据库中不同样本（某一问题的不同表现形式）构建的复合树及其生成的预测值能够稳定并降低这样的高方差。这种方法被称作引导聚集算法（bootstrap aggregating），其简称 bagging 正好是装进口袋，袋子的意思，所以被称为「装袋算法」。该算法的局限在于，由于生成每一棵树的贪婪算法是相同的，那么有可能造成每棵树选取的分裂点（split point）相同或者极其相似，最终导致不同树之间的趋同（树与树相关联）。相应地，反过来说，这也使得其会产生相似的预测值，降低原本要求的方差。

我们可以采用限制特征的方法来创建不一样的决策树，使贪婪算法能够在建树的同时评估每一个分裂点。这就是随机森林算法（Random Forest algorithm）。

与装袋算法一样，随机森林算法从训练集里撷取复合样本并训练。其不同之处在于，数据在每个分裂点处完全分裂并添加到相应的那棵决策树当中，且可以只考虑用于存储属性的某一固定子集。

）

缺失值这种情况在现实问题中非常普遍，这会导致一些不能处理缺失值的分析方法无法应用，因此，在信用风险评级模型开发的第一步我们就要进行缺失值处理。缺失值处理的方法，包括如下几种。

（1） 直接删除含有缺失值的样本。

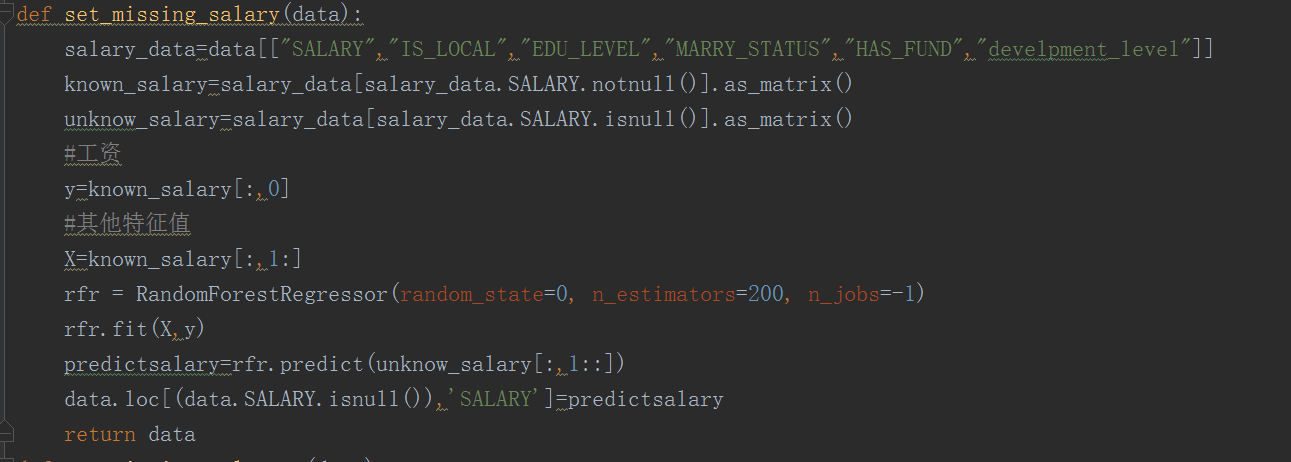
（2） 根据样本之间的相似性填补缺失值。

（3） 根据变量之间的相关关系填补缺失值。

在本案例中，对于AGENT,由第二步的数据可视化发现与Y值关系程度很低，并且缺失值太多，考虑舍弃。参考其他表格的数据，填补出HAS\_FUND和EDU\_LEVEL的数据（别人做的，我不清楚），WORK\_PRIVINCE无法从其他表格获取，且用LASSO原理进行特征关联度分析时，发现work-privince与Y值得coef系数为0，故舍弃，从其他表格提取出development\_level属性。

随后对于salary 用随机森林算法进行填补。

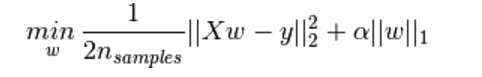
代码如下：



1. **特征选择**
2. **LassoCV初步特征选择**
   1. **LassoCV原理**

LASSO（ least absolute shrinkage and selection operator，最小绝对值收缩和选择算子）方法与岭回归和LARS（least angle regression，最小角回归）很类似。与岭回归类似，它也是通过增加惩罚函数来判断、消除特征间的共线性。与LARS相似的是它也可以用作参数选择，通常得出一个相关系数的稀疏向量**。**

**L**asso 是一种估计稀疏线性模型的方法.由于它倾向具有少量参数值的情况，对于给定解决方案是相关情况下，有效的减少了变量数量。 因此，Lasso及其变种是压缩感知(压缩采样)的基础。在约束条件下，它可以回复一组非零精确的权重系数。用数学形式表达，Lasso 包含一个使用 ell\_1 先验作为正则化因子的线性模型。其目标函数是最小化:



lasso 解决带 \alpha ||w||_1 惩罚项的最小平方和，其中 \alpha 是一个常量，||w||_1 是参数向量的 \ell_1-norm。Lasso 类实现使用了坐标下降法(一种非梯度优化算法)来拟合系数

* 1. **合并表格并标准化**

将训练集与所有其他非测试集的表格合并，去掉缺失值，用剩下的数据进行第一步的数据预处理，字符串转化成数字。

* 1. **特征选择**

采用LassoCV模型，以除了‘Y’之外的特征作为X,’Y’作为目标y,选择出coef不是0的特征，作为第一步的粗选特征。

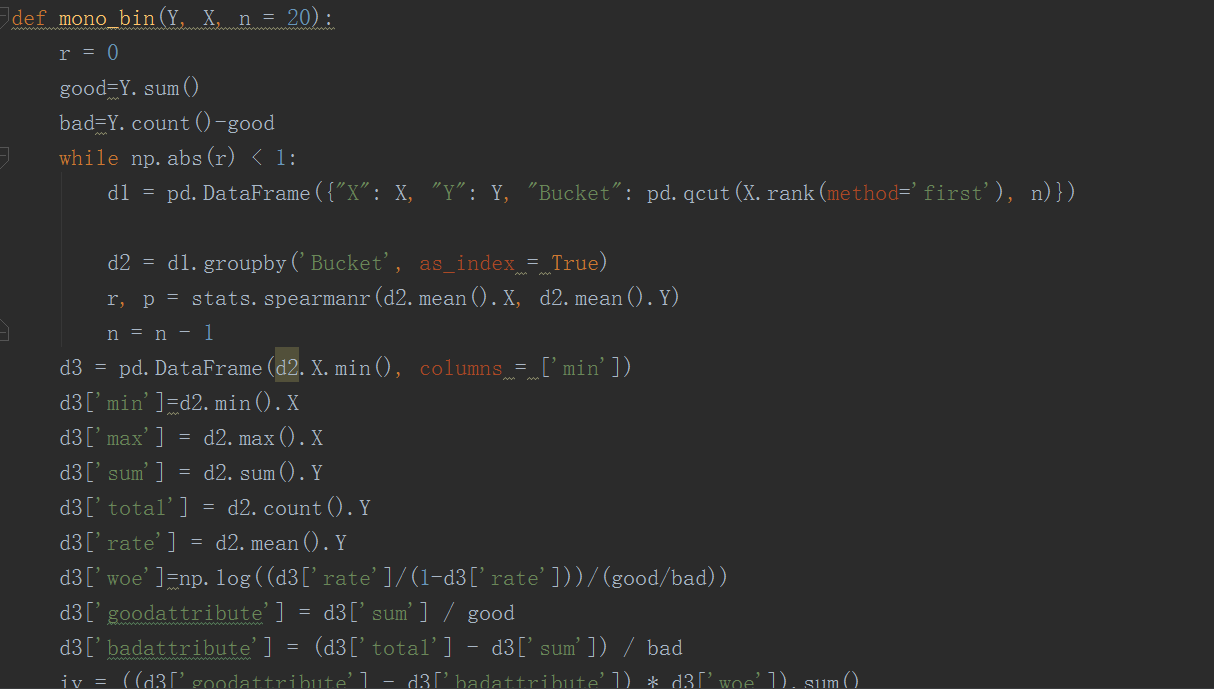
1. **信用评分模型的变量选择方法**

用上一步的得到的粗选特征，我们进行WOE分析方法，即是通过比较指标分箱和对应分箱的违约概率来确定指标是否符合经济意义。

* 1. 分箱处理

变量分箱（binning）是对连续变量离散化（discretization）的一种称呼。信用评分卡开发中一般有常用的等距分段、等深分段、最优分段。其中等距分段（Equval length intervals）是指分段的区间是一致的，比如年龄以十年作为一个分段；等深分段（Equal frequency intervals）是先确定分段数量，然后令每个分段中数据数量大致相等；最优分段（Optimal Binning）又叫监督离散化（supervised discretizaion），使用递归划分（Recursive Partitioning）将连续变量分为分段，背后是一种基于条件推断查找较佳分组的算法。

我们首先选择对连续变量进行最优分段，在连续变量的分布不满足最优分段的要求时，再考虑对连续变量进行等距分段。最优分箱的代码如下:



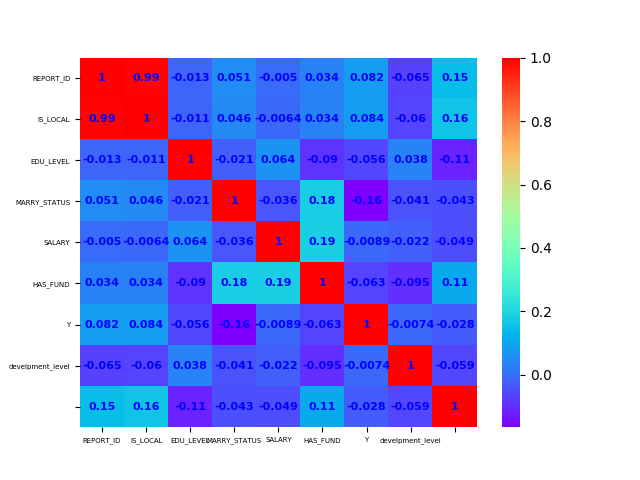
* 1. WOE

WoE分析， 是对指标分箱、计算各个档位的WoE值并观察WoE值随指标变化的趋势。其中WoE的数学定义是:woe=ln(goodattribute/badattribute)在进行分析时，我们需要对各指标从小到大排列，并计算出相应分档的WoE值。其中正向指标越大，WoE值越小；反向指标越大，WoE值越大。正向指标的WoE值负斜率越大，反响指标的正斜率越大，则说明指标区分能力好。WoE值趋近于直线，则意味指标判断能力较弱。若正向指标和WoE正相关趋势、反向指标同WoE出现负相关趋势，则说明此指标不符合经济意义，则应当予以去除。

woe函数实现在上一节的mono\_bin()函数里面已经包含，这里不再重复

* 1. 相关性分析和IV筛选

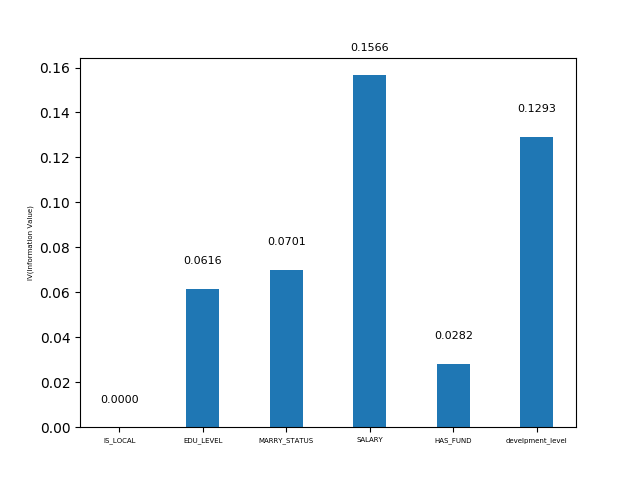
接下来，我们会用经过清洗后的数据看一下变量间的相关性。注意，这里的相关性分析只是初步的检查，进一步检查模型的VI（证据权重）作为变量筛选的依据。



变量之间相关性图

接下来，我进一步计算每个变量的Infomation Value（IV）。IV指标是一般用来确定自变量的预测能力。 其公式为：  
IV=sum((goodattribute-badattribute)\*ln(goodattribute/badattribute))

如下图：



1. **WOE转换**

证据权重（Weight of Evidence,WOE）转换可以将sklearn中的各种回归模型转变为标准评分卡格式。引入WOE转换的目的并不是为了提高模型质量，只是一些变量不应该被纳入模型，这或者是因为它们不能增加模型值，或者是因为与其模型相关系数有关的误差较大，其实建立标准信用评分卡也可以不采用WOE转换。这种情况下，sklearn中的各种回归模型需要处理更大数量的自变量。尽管这样会增加建模程序的复杂性，但最终得到的评分卡都是一样的。  
在建立模型之前，我们需要将筛选后的变量转换为WoE值，便于信用评分。

1. **划分train集**

由于官方发布的test文件没有Y值，预测结果无法评价，于是采用train\_test\_split方法随机切分第四步形成的WoE数据，将1/4的数据作为测试集，3/4的数据作为训练集。

代码如下：

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, random\_state=1)

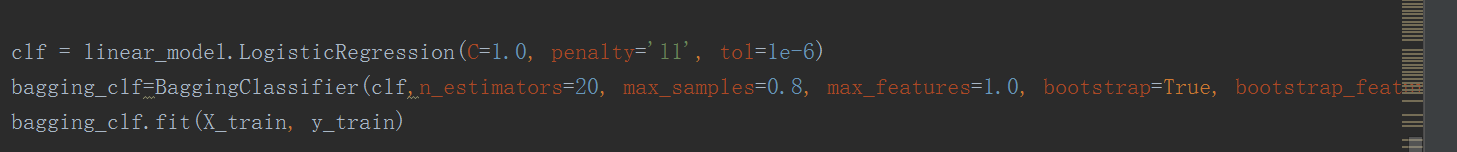
1. **建立模型**

**采用四种模型，分别是LogisticRegression模型，DecisionTreeClassifier模型，RidgeClassifier模型，SGDClassifier模型。并且分别用上一步得出的训练集代入模型中，训练，得出模型，保存。**

**（百度，写具体一点）**

1. **模型融合**

用bagging方法。Bagging是并行式集成学习代表方法。基于“自助采样法”（bootstrap sampling）。自助采样法机制：给定包含m个样本的数据集，我们先随机取出一个样本放入采样集中，再把该样本放回初始数据集，使得下一次采样时该样本还会被采到。这样，经过m次样本采集，我们得到包含m个样本的采样集。采样集中，有的样本出现过很多次，有的没有出现过。Bagging机制：我们采样出T个含m个样本的采样集。然后基于每个采样集训练出一个学习器，再将学习器进行结合。对分类任务使用投票法，对回归任务采用平均值法。  
代码如下：



1. **交叉验证**

**用第6步划分出的模型，预测第五步得出的测试集，得出准确率如下：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **模型** | **逻辑回归** | **决策树** | **Ridge** | **SGD** |
| **准确率** | **93.80%** | **96．87%** | **93.80%** | **93.80%** |

**得出决策树模型的准确率最高，即是在此四类中的最佳模型**

1. **模型预测**

**test集合同上书步骤一样，预处理，缺失值处理，WoE转化后将数据代最佳模型中预测出Y值。**