**使用机器学习制作借贷申请评分卡**

1. **项目背景与意义**
   1. **数据来源**

**二．分析思路**

**二．1 具体思路以及思维导图**

**三．数据分析**

**三．1 数据分析流程**

**三．2 重复值处理**

**三．3 异常值处理**

**四．特征工程**

**四．1样本平衡化处理**

**四．2 特征选择**

**四．2.1 卡方分箱**

**四．3 单变量分析**

**四．4 多变量分析**

**五．模型建立**

**五．1分类模型选择**

**五．1.2 KNN分类**

**五．1.3 SVM分类**

**五．1.4 逻辑回归分类**

**五．1.5 使用交叉验证选择模型**

**五．2逻辑斯蒂回归建模**

**五．2.1逻辑回归导论**

**五．2.2 逻辑回归的优点**

**五．2.3 逻辑回归的缺点**

**五．2.4 具体步骤**

**五．2.5 参数调整**

**五．2.6 性能评估**

**六．分析结果**

**七．总结。 七。1 优点与创新点**

**七．1.1使用随机森林填补缺失值**

**七．1.2基于交叉验证的模型选择**

**七．2 不足与改进**

**七．2 分类模型的不足与改进**

1. **问题背景**

在银行与P2P借贷场景中，评分卡是一种以分数形式来衡量一个客户的信用风险大小的手段，它衡量受信人或需要融资的公司不能如期履行合同中的还本付息责任致使授信人或银行等金融机构承担经济损失的可能性的一种方式。

为什么要开发申请评分卡？？（画成图像）

1风险控制：借贷生命周期的第一个关口

2营销：优质客户识别

3资本管理：可作为PD模型的一个因子

评分卡的特性

1稳定性

当总体逾期/违约概率不变时，分数的分布也应不变

2区分性

违约人群与正常人群的分数应当有显著差异

3预测能力

低分人群的违约率更高

4和逾期概率等价

评分可以精准地反映违约/逾期概率，反之亦然

在以往的传统金融学分析中，主要采用专家赋值的办法产生供业务人员使用的评分卡，不过在当代，金融机构在风险管理的每个环节都尽可能地引入计量分析方法，依托大数据进行后台的分析回顾，不断的优化调整，使得金融机构在风险与收益的博弈过程中更快达到平衡，实现局部甚至更多空间的利润最大化。

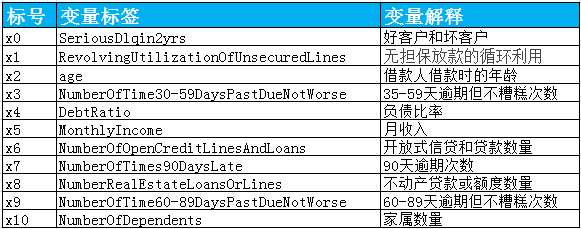
【这里插入一些相关图片】

对于个人来说，我们有”四张卡“来评判个人的信用程度：A卡，B卡，C卡和F卡。 而众人常说的“评分卡”其实是指A卡，又称为申请者评级模型，是一个贷前评分体系，主要应用于相关融资类业务中新用户的主体评级，即判断金融机构是否应该借钱给一个新用户，反之如果，如果这个人的风险太高，我们可以拒绝贷款。

**数据集展示**

我们使用的数据集来自于竞赛网站Kaggle，名字叫做GivemeSomeCredit,源自14年一家德国银行的真实数据。

总共有10个特征值1个标签值:



这些特征经过分类，可以归为以下几个属性：

– 基本属性：包括了借款人当时的年龄。

– 偿债能力：包括了借款人的月收入、负债比率。

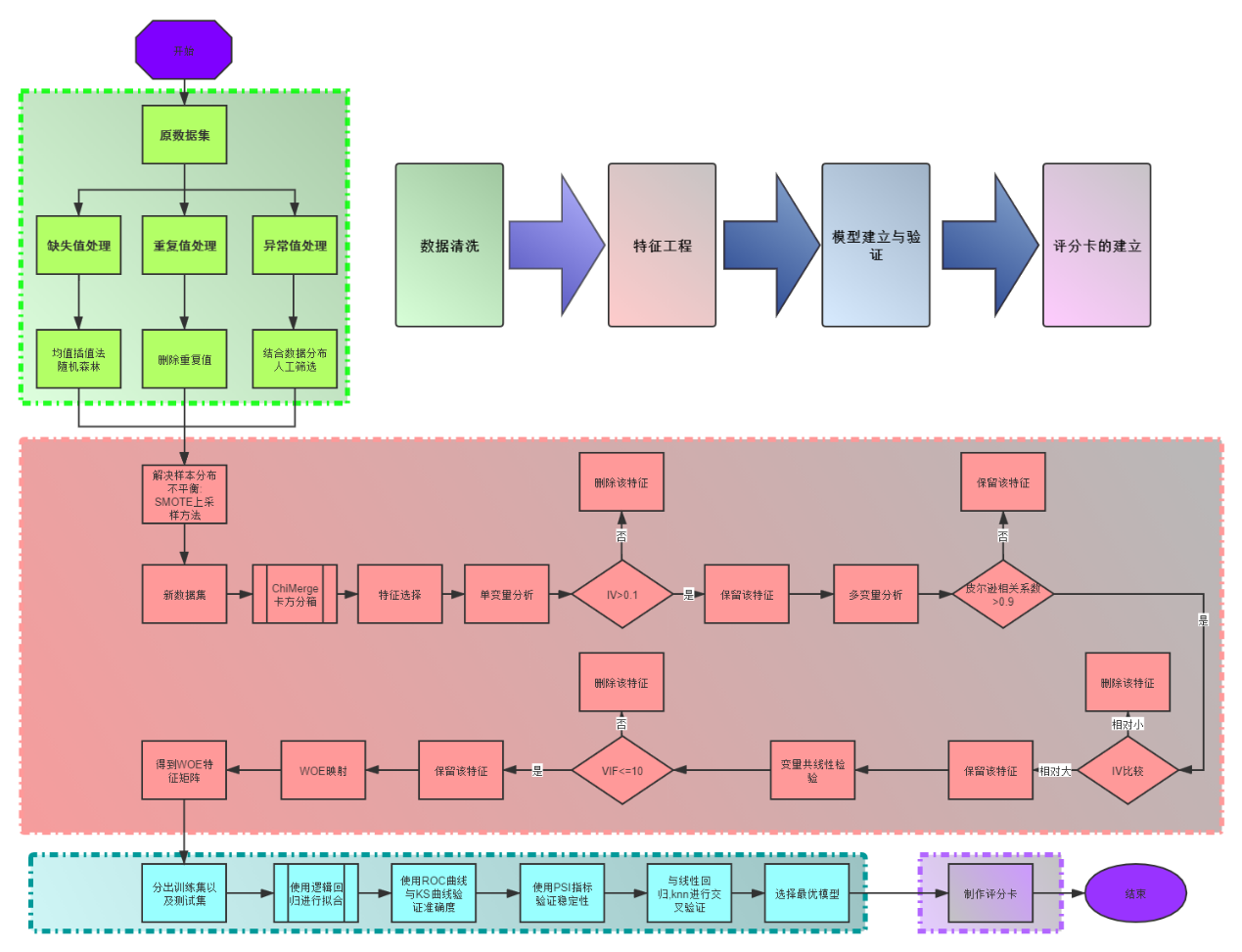
– 信用往来：两年内35-59天逾期次数、两年内60-89天逾期次数、两年内90天或高于90天逾期的次数。

– 财产状况：包括了开放式信贷和贷款数量、不动产贷款或额度数量。

– 其他因素：包括了借款人的家属数量（不包括本人在内）。

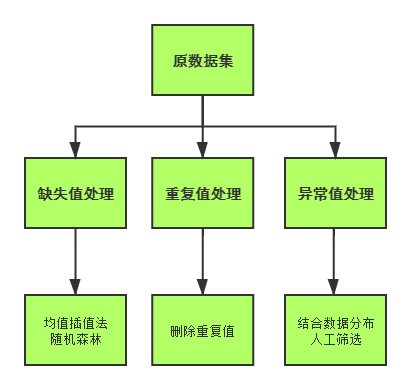
**分析思路**

**项目总流程图**

****

**数据清洗**

**数据清理流程图**



**重复值处理**

对于银行数据，有一个普遍的问题就是样本重复，有时候是认为输入重复，也可能是系统录入重复，总之我们需要对数据进行重复处理，当然，也有可能两个样本的特征本来就是一模一样的比如一某一样名字，年龄，学历，工资，性别……当特征很少的时候，是有可能发生这种情况的，但是所有的特征都一样的可能性是微乎其微的，即使真的出现了，我们也可以当作少量的形式损失来对待，将这条数据当做重复值去除。我们使用python的pandas模块来进行处理。

**处理前的数据**

【处理成表格】

特征 样本量 是否为空值 数据类型

SeriousDlqin2yrs 149391 non-null int64

RevolvingUtilizationOfUnsecuredLines 149391 non-null float64

age 149391 non-null int64

NumberOfTime30-59DaysPastDueNotWorse 149391 non-null int64

DebtRatio 149391 non-null float64

MonthlyIncome 120170 non-null float64

NumberOfOpenCreditLinesAndLoans 149391 non-null int64

NumberOfTimes90DaysLate 149391 non-null int64

NumberRealEstateLoansOrLines 149391 non-null int64

NumberOfTime60-89DaysPastDueNotWorse 149391 non-null int64

NumberOfDependents 145563 non-null float64

**处理后的数据**

【处理成表格】

特征 样本量 是否为空值 数据类型

SeriousDlqin2yrs 149391 non-null int64

RevolvingUtilizationOfUnsecuredLines 149391 non-null float64

age 149391 non-null int64

NumberOfTime30-59DaysPastDueNotWorse 149391 non-null int64

DebtRatio 149391 non-null float64

MonthlyIncome 120170 non-null float64

NumberOfOpenCreditLinesAndLoans 149391 non-null int64

NumberOfTimes90DaysLate 149391 non-null int64

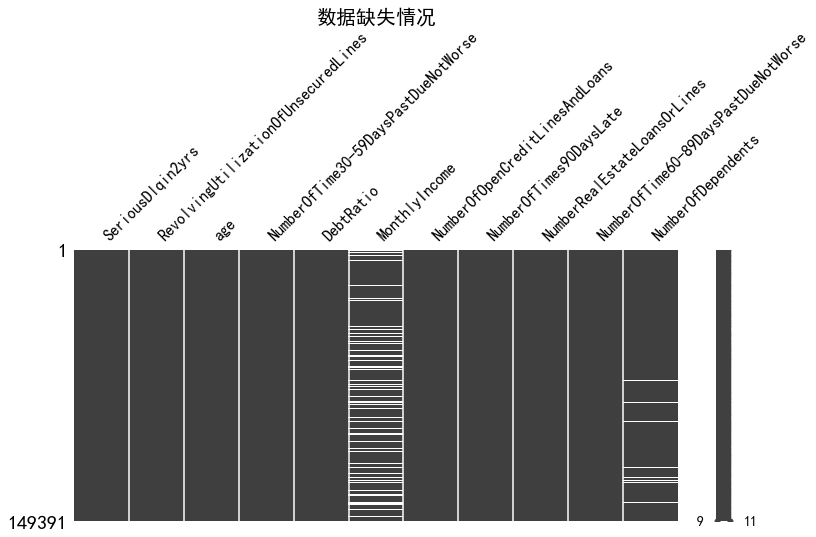
NumberRealEstateLoansOrLines 149391 non-null int64

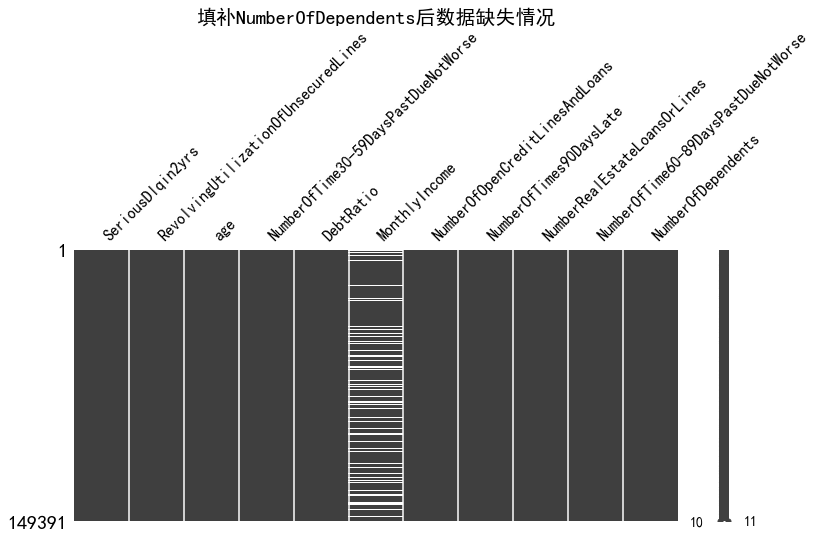
NumberOfTime60-89DaysPastDueNotWorse 149391 non-null int64

NumberOfDependents 145563 non-null float64

**处理缺失值**

第二个问题，就是处理缺失值，对于缺失值，我们一般有两种处理方法，一种是直接删除整行的数据，一种是进行填补，从下图我们可以看出数据的缺失情况：



白色代表缺失，在这里我们需要填补的特征是“月收入”和“家属人数”,家庭人数的缺失不严重，只占到该特征的2%左右，对于“家庭人数”**，**我们可以直接删去，也可以填充，这里使用均值法进行填充，即取缺失数据周围数据的均值进行填充：

“月收入”的缺失比较严重，占到了20%左右，同时，从业务的角度考虑，收入应该是对信用评分来说一个很重要的因素，因此这个特征必须要进行填补，但是均值填补法，对于缺失比例较大的“月收入”，显然是不合适的，我们可以这样去考虑这个问题，一个人来借钱，他应该知道，高收入和稳定收入对于他而言应该是一个有益的证据，因此，如果收入较高或者稳定的客户，他会更加倾向于将自己的收入填上，那么收入栏缺失的客户，更有可能是收入不稳定或者收入比较低的，根据这种判断，我们可以用四分位数来填补缺失值，将收入栏空的客户全部当成低收入人群，当然，这种方法并不严谨，也有可能这种缺失确实是数据收集过程中造成的失误，最好的方法是去和业务人员沟通，观察缺失值是如何产生的。在这里，我们采用随机森林法来进行填充

随机森林法

【原理部分】

【使用随机森林】

**五．模型建立**

**五．1模型选择**

**对于分类问题，在机器学习领域中已经有许多成熟的算法可以用来解决。但是对于一个未知的样本集，是没有办法提前确定一套最好的方案的，目前采用最广泛的方案是使用交叉验证集法来进行模型的选择，在这里，我们横向比较了目前比较流行的三种分类模型的分类能力，这三个模型分别为Knn（K-NearestNeighbor ，最近邻算法）、SVM（Support Vector Machine，支持向量机），LR（logstic Regression，逻辑斯蒂回归），受限于机器的性能，我们只选取了2%的样本（即2000条数据）进行检测。**

**五．1.1KNN模型**

**在此数据集中，我们检测了三个KNN模型的准确度，从中挑选了准确度最高的一种，加入到交叉验证模型组中，这三组分别为：**

**~普通KNN**

**近邻样本数：10**

**准确率：0.7049999999999998**

**~带权重的KNN**

**近邻样本数：10,权重模式：“distance”**

**准确率：0.721**

**~指定半径的KNN**

**近邻样本数：10,半径为500**

**准确率：0.48999999999999994**

**五．1.2 SVM模型**

在这里，我们使用了高斯核函数，并且使用GridSearch参数最优化确定参数gamma:

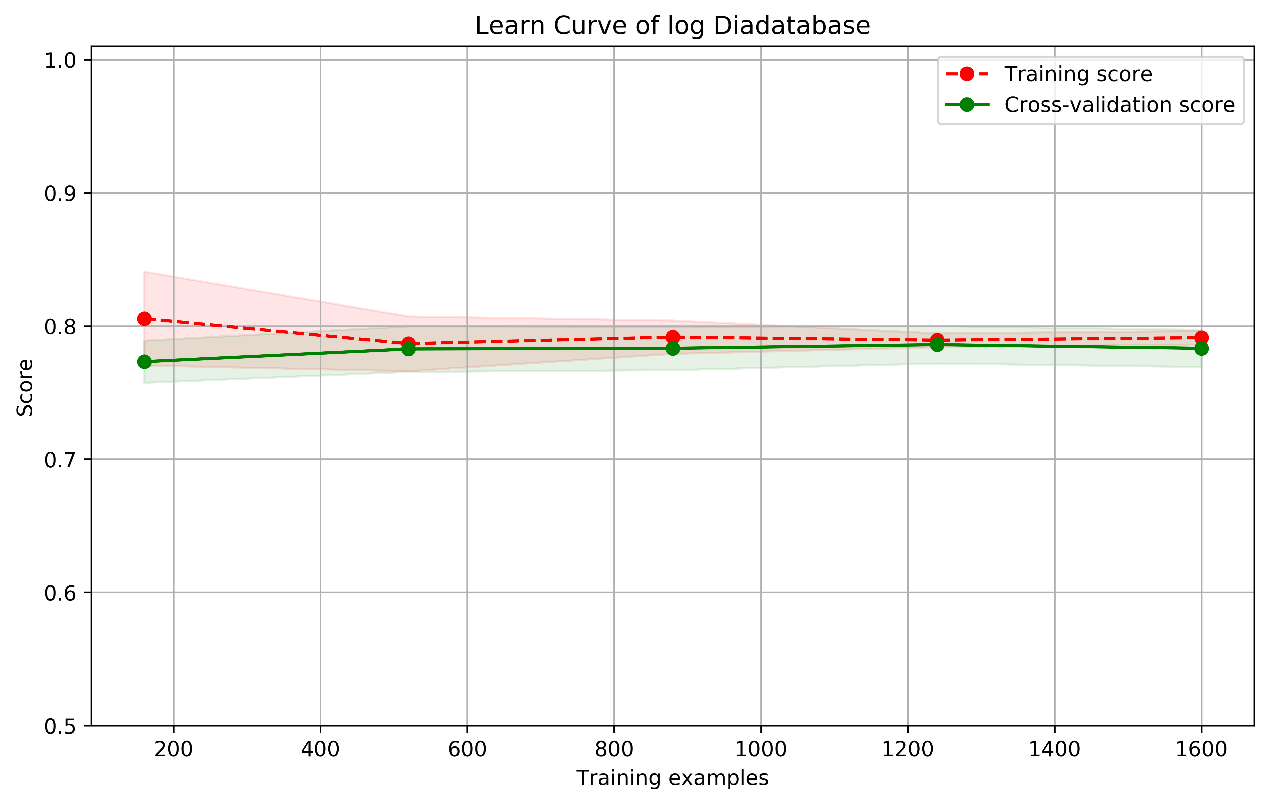
准确度：0.7785

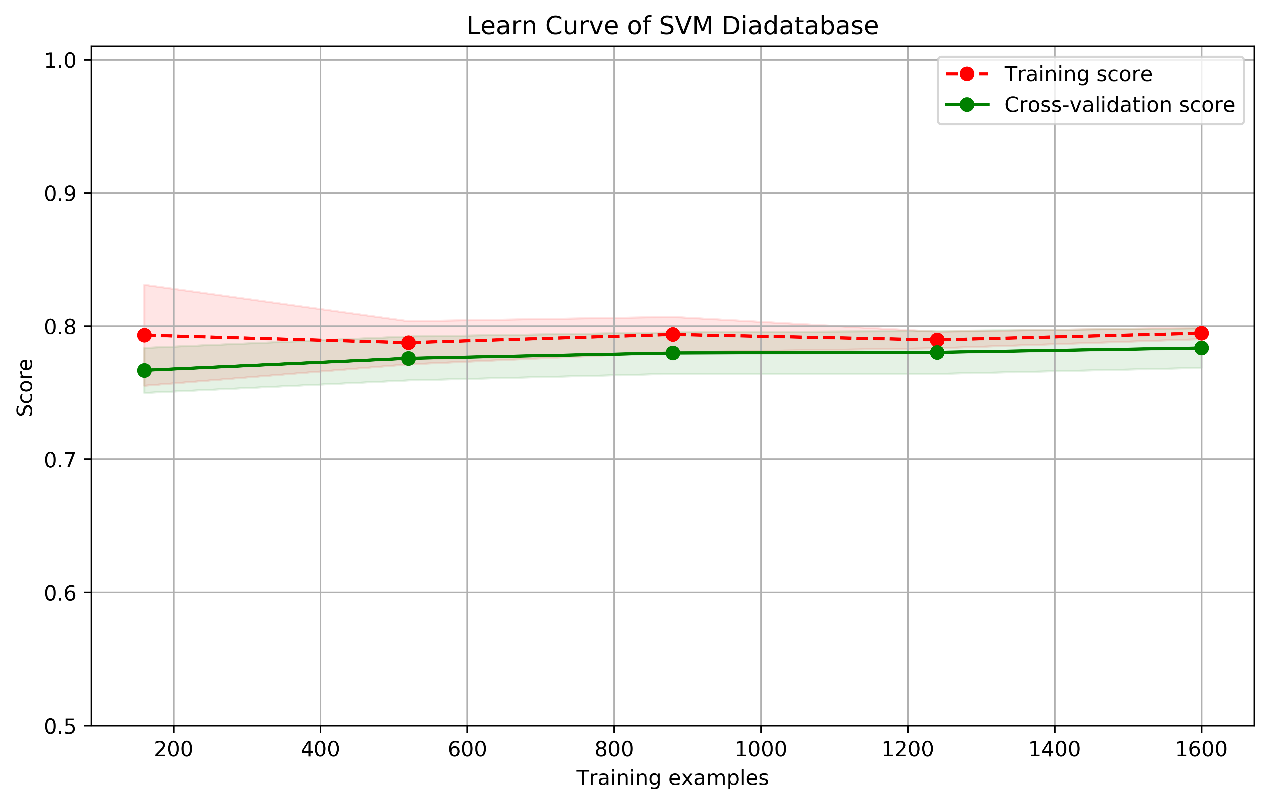
**五．1.3 Logsitic模型**

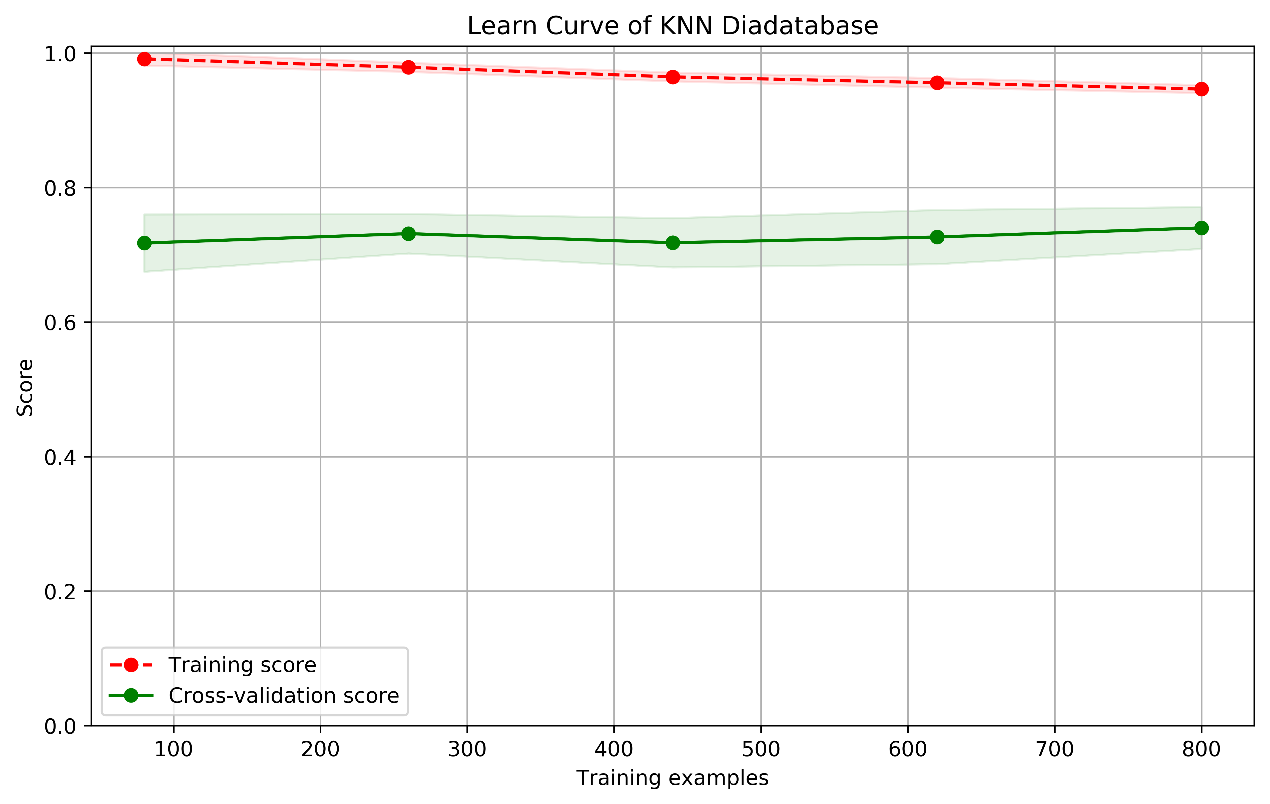
**在这里我们使用L2正则化的逻辑斯蒂模型**

**准确率：0.78**

**三种模型的学习曲线比较：**

**逻辑斯蒂回归：**

**SVM：带权重KNN：**

****

可以看出，其中SVM与逻辑斯蒂表现良好，而KNN则出现了过拟合的现象，其次考虑到逻辑斯蒂模型在平均准确率与运行效率上表现均高于SVM，我们最后决定采用逻辑斯蒂回归模型进行建模。

五．2 逻辑斯蒂模型

五．2.1 逻辑斯蒂回归导论

【逻辑斯蒂回归基础理论部分】

【正则化部分】

五．2.1 逻辑斯蒂回归的优点

1.模型简单，不宜过拟合

2.模型的可解释性好

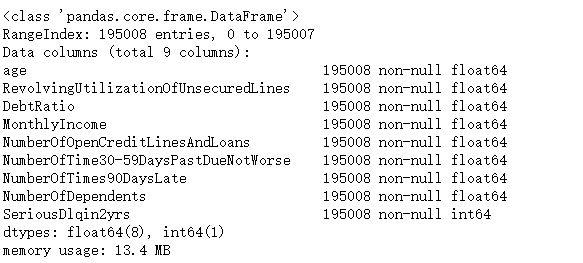
五．2.2逻辑斯蒂回归的缺点

1.对于非线性的数据拟合能力较差

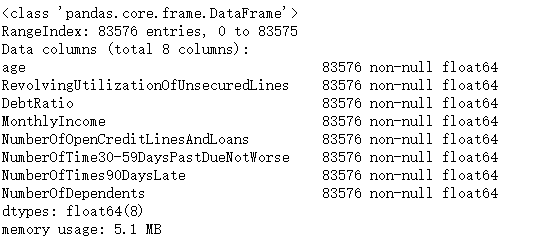
五．2.3具体步骤：

首先，使用已经处理好的建模数据：

训练集：



测试集：



我们使用L2正则化训练逻辑回归模型后，得到的准确率为：0.7736072556714846

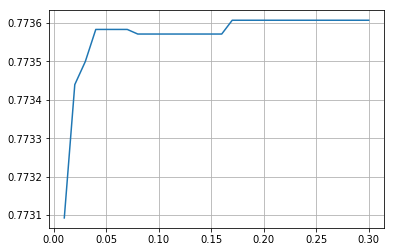
五．2.4 参数调整

对于逻辑斯蒂回归模型，主要有两个参数可以优化，一个是正则化参数C，另一个是迭代轮数，前者属于超参数调整，因为不同的正则化参数，对于损失函数的惩罚力度是不同的，而由于在这里的逻辑斯蒂回归使用的是梯度下降法寻找损失函数最小值的策略，所以容易陷入局部最优，所以，多次迭代选择不同的初始点可以一定程度改善这个问题

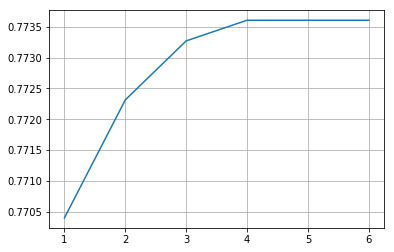
对于正则化参数C，使用线性搜索法，从0.0到0.3，步长为0.003，逐一搜

索：

【图像一】



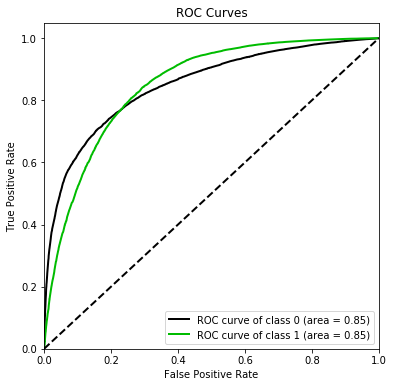
对于迭代轮数：我们一共选择了6轮来观察最优的那个轮数



从上面两张图片中以看出，模型的性能到达瓶颈，无论是调整正则化参数还是增加轮数，模型的准确度均会收敛于0.7736左右，不过这为我们下一步的优化指明了方向（L2正则化（lasso）倾向于删掉变量，而随着C的增大，模型的准确度上升说明，更少的特征系数被压缩到0，或者说越少的特征被删除，可能会导致模型的准确率上升，所以需要增加特征来提高准确度。）

五．2.5 性能评估

对于一个分类器，光用准确率进行评判其实是不准确的，比如说，有99个人不违约，但是有一个人违约，这时分类器判断100个人全部不违约，那么这个模型的准确率就是99%.但是这样的分类器是没有意义的，这个时候我们需要使用ROC曲线来评判模型



简而言之，ROC曲线下面积即AUC越大，证明模型的泛化能力越好，越有用，在这里，判断为“不违约”的AUC与“违约”的AUC是相等的为0.85，说明模型的分类能力较强，当然，一般地我们希望分类器得到的分类结果是完全正确的，也就是正例样本全部都能够被检测出来，也就是全部都是真正例，或者真反例，这个时候TPR=1且FPR=0，反应在图像上好的分类器的折线应该更加接近左上角。从我们的图像上可以看出，当把把阈值设为0.5时，可以看出模型更擅长分类正例（即“违约”）。