福瑞泽项目xgboost实验文档

**一、综述**

本文档为福瑞泽项目xgboost实验相关思路和代码的描述文档。项目要解决的问题是，现有一个包含了大量用户金融信息以及是否发生违约标记的数据集，要求把数据集划分为train，val与test集，并根据train训练出一个模型，能够在val和test上取得较高的AUC与KS值。该问题是一个二分类问题，且标记类别极度不平衡。

本实验使用的主要方法为xgboost算法。实验的主要过程包括，使用xgboost按照特征重要性筛选特征——按照数据缺失情况增补特征及填充缺失值——分割数据集，使用xgboost作为分类器调参并获得最好效果——把概率分布改变为近似正态分布并画出混淆矩阵。

实验分为两组，第一组按照时间划分train，val与test，主要代码现为“4.13代码整合\_time.py”。第二组采用十折交叉验证，完全随机划分train，val与test，主要代码现为“4.13代码整合\_notime.py”。

**二、环境配置**

实验使用的操作系统为Ubuntu，语言为python-3.5.1，主要工具包为numpy-1.31.1，pandas-0.20.3，xgboost-0.7.post4，matplotlib-2.0.2，scikit-learn-0.18.2，scipy-0.19.1。使用Ctrl+Alt+T打开终端，安装python-3.5.1的命令为：

>> sudo apt-get install python3.5.1

安装工具包的命令为：

>> pip3 install numpy, scipy, pandas, matplotlib, scikit-learn

上述包安装时可以指定版本，例如 >> pip3 install numpy==1.31.1，版本不同可能会有细微区别。如果使用集体服务器务必加--user，例如 >> pip3 install numpy --user。

Xgboost包的安装需要先从GitHub下载源文件然后编译安装：

>> git clone --recursive https://github.com/dmlc/xgboost

>> cd xgboost; make -j4

**三、特征筛选**

公司给的原始数据维度为(79054, 286)，即样本数为79054，每个样本含285个特征和1个label标签，存在缺失值。label包括0和1，分别表示未违约与发生违约。其中0的数量为78391, 1的数量为663，极度不平衡。

筛选特征的代码为FeatureExact.py中的feature\_select()函数。该函数首先读入原始数据集，剔出label列，按照0.25的test比例用train\_test\_split()进行划分，再用MinMaxScaler()进行数据归一化。使用xgboost在划分出的dtrain上做训练得到xg\_model，然后xg\_model的get\_fscore接口可以返回每维特征的重要性importance，再对importance由高到低排序，选取前113列与label列合并并返回，作为筛选后的特征。本过程还把每维特征的名称和重要性存储为feat\_importance.csv，方便后续使用。

特征筛选的目的主要是降低数据冗余度，提高后续训练的效率。

**四、根据缺失情况增补特征及填充缺失值**

由于数据存在缺失值-1和-2，实验根据缺失情况增补了一系列的特征，作为后续训练的依据，并进行了缺失值填充。这部分代码在utils.py中的data\_init()函数中。“4.13代码整合\_time.py”与“4.13代码整合\_notime.py”在使用时直接调用此函数。

**1、根据缺失情况增补特征**

选择特征重要性值level = 6为分界线，特征重要性大于6的定义为重要特征，小于6的定义为非重要特征。首先读入之前存储的feat\_importance.csv文件，根据每个特征的重要性，把重要特征和非重要特征的名称分别存入imp和unimp两个list中，然后把数据集按照特征重要与否分为imp\_data与unimp\_data。再在每个样本中分别统计重要特征与非重要特征的-1与-2各自占比以及总占比，分别记做missValue\_1\_imp，missValue\_2\_imp，missValue\_all\_imp，missValue\_1\_unimp，missValue\_2\_unimp，missValue\_all\_unimp，6个list。再统计-1与-2在每个样本所有特征数量中的占比及总占比，分别记做missValue\_1\_all，missValue\_2\_all，missValue\_all\_all，3个list。这样总共得到了9个list，即9列新特征。把这些新特征并入数据集中。

**2、缺失值填充**

首先使用了replace函数把-1和-2全部替换为np.nan，即为空值。然后计算数据集每列的中值，新建一个从列序号映射到该列中值的字典miss\_value\_map，使用fillna()方法和该字典来进行填充。即，把每列的缺失值替换为该列的中值。

**3、level值与填充方法选取依据**

主要依据是后续拆分数据集、训练时，AUC与KS表现最好。然而由于xgboost调参过程太复杂，耗时太多，在此实验之前做了一个由7个lr基分类器+bagging/stacking+GridSearchCV自动调参的实验，分别验证了level取值从1到10，以及缺失值填充方式为中值、众数、均值等不同情况下的效果，结果发现level = 6以及中值填充时，不论是否按照何种方式划分数据集，均取得最好效果。因此在此基础上决定了处理缺失值的方法，用于后续xgboost调参。

**五、分割数据集并用xgboost调参**

如前所述，分割数据集方法包括时间与非时间两种。第一种方法按照时间，1-10月份的数据作为train，11月份数据作为val，12月份数据作为test。第二种方法完全随机，并且采用十折交叉验证，即把数据集随机分为10份，每次取8份作为train，1份作为val，1份作为test，进行使用，共进行10次，保证每份都做一次val和test。

两种方法在划分完数据集后，均使用xgboost作为分类器。按时间划分时，先归一化再分类；不按时间划分时，十折的每次训练前都要先进行归一化。由于xgboost参数太多，不能遍历所有情况，xgboost的sklearn接口效果并不好，也没有使用GridSearchCV，因此我们完全进行了手动调参。基本方法是，首先使用一组比较平均的参数，然后依次调整每个参数，观察AUC与KS在val与test集上的表现，得到每个参数的最优值。这样能得到一个全局较优解，可能不是最优解，但还是可以接受的。

最终结果为，按照时间划分时，现能达到的最佳效果如下：

|  |  |  |  |
| --- | --- | --- | --- |
|  | Train | Val | Test |
| AUC | 0.8855 | 0.8797 | 0.7831 |
| KS | 0.6242 | 0.6047 | 0.4818 |

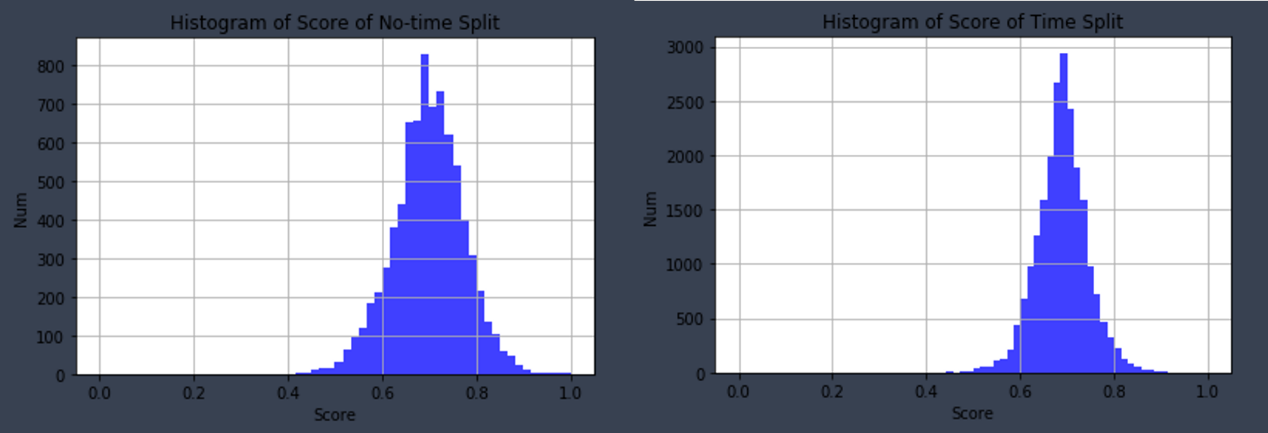
非时间划分时，能达到的最佳效果是：

|  |  |  |  |
| --- | --- | --- | --- |
|  | Train | Val | Test |
| AUC | 0.8513 | 0.8092 | 0.8104 |
| KS | 0.5523 | 0.5081 | 0.5205 |

**六、改变概率分布与画混淆矩阵**

**1、把概率分布转化为近似正态分布**

Xgboost模型输出的结果是划分为每类的概率，我们看重的每个样本预测为1的概率，即判断为坏人的概率，这个概率多集中在0-0.05之间。应公司要求，为了方便以后做区分，我们把这个概率转化为近似正态分布。首先采用如下公式转化为近似正态分布的score：

公式中P为原概率，设定参数factor为ln(60)，offset为600。这组参数是经过一系列验证，认为效果较好的一组参数。然后再根据score得分的最高值max与最低值min，用所有预测的概率值减去min再加0.01，然后除以（max-min+0.01），这样能把score转化为0-1分布的概率，且避免了出现0和1，可以作为最终结果。画出的两幅概率分布图如下：

**2、画出混淆矩阵**

以二分类问题为例，模型输出总共可能会出现四种情况，即为混淆矩阵：

|  |  |  |  |
| --- | --- | --- | --- |
|  | 预测1 | 预测0 | 合计 |
| 真实1 | True Positive (TP) | False Negative (FN) | Actual Positive |
| 真实0 | False Positive (FP) | True Negative (TN) | Actual Negative |
| 合计 | Predicted Positive | Predicted Negative | 1 |

对于一个二分类模型，输出的最初结果是连续的概率值；假设已经确定一个阀值，那么最初结果大于阀值时，则输出最终结果为1，小于阀值则输出为0。我们以不同阈值下的FPR为横坐标，TPR为纵坐标，把不同的点连成曲线，就得到了ROC曲线。ROC曲线下方的面积为Area Under the ROC Curve，简称为AUC。这是评价模型的一个方法，AUC值越大，说明模型的分辨效果越好。

常用的模型评价还有K-S曲线。如果以阈值为横坐标，FPR和TPR为纵坐标，即可做出FPR曲线与TPR曲线。常用的KS值为max(TPR-FPR)，即是两条曲线之间的最大间隔距离。KS值越大，说明模型的分辨效果越好。

计算混淆矩阵使用confusion\_matrix()函数即可。先确定一个阈值，然后划分得到预测的label。综合各种因素，实验最终选取了召回率为50%对应的阈值。时间划分数据集时，混淆矩阵为[19155, 2811; 74, 75]，非时间划分时，混淆矩阵为[6838, 1001; 33, 33]。顺序为[TN, FP; FN, TP]

**七、总结**

本文档较为详细地描述了福瑞泽项目xgboost实验中的多数步骤与细节，希望能对读者了解该实验有所帮助。