基于 XGBoost 的信用评分模型(下)

3.2 XGBoost 调参

至此，我们把 XGBoost 的理论介绍完了，现在将使用该算法来训练本文的数据，在训练过程中需要对参数进行调整，以得到最优的参数。一般来说，我们调节的主要参数如表 1 所示：

表 1 XGBoost 调节的主要参数表

主要参数	说明	通常取值范围
learning_rate	学习率	[0,1]，默认为 0.3
n_estimators	模型中所建立的树的数量	[0,1000]，默认为 100
max_depth	树的最大深度，越大越容易过拟合	[3,15]，默认为 6
min_child_weigh t	划分到某个叶子节点的所有样本权重之和的最小值	[0,10]，默认为 1
gamma	划分叶子节点所需要达到的最小损失减少值	[0,10]，默认为 0
subsample	训练时使用的子样本占全部训练集的比例	[0.5,1]，默认为 1
colsample_bytre e	构建树时使用的特征占总特征的比例	[0.5,1]，默认为 1
reg_alpha	L1 正则化权重，越大越可以防止过拟合	[0,10]，默认为 0
reg_lambda	L2 正则化权重，越大越可以防止过拟合	[0,10]，默认为 1

本文使用 xgboost 包来训练模型，并结合 sklearn 包的网格搜索函数来寻找最优参数。调参时，先给每个参数一个固定的取值，再给出需要调节的参数的取值范围同时保持其他参数值不变，由此计算该参数的不同取值下模型的 AUC 评分，最大的 AUC 分数对应的参数值即为该参数的的最优取值，将此时的最优参数值替代模型中该参数的给定值再按此方法进行下一个参数的调节。按上述方法得出的最优参数如表 2 所示：

表 2 XGBoost 调参结果表

所调节的参数	最优参数值	参数得分
learning_rate	0.07	0.906
n_estimators	180	0.907
max_depth	11	0.911
min_child_weight	1	0.911
gamma	0	0.911
subsample	0.9	0.913
colsample_bytree	0.7	0.913
reg_alpha	0.01	0.912
reg_lambda	3	0.912

我们从上述调参顺序的模型得分可以看出，从最开始的 learning_rate 到 colsample_bytree，模型的得分一直在上升，这说明随着最优参数的不断调节模型模型越来越好。到 reg_alpha 和 eg_lambda 这里，模型得分有些下降，这是因为我们调节了 L1 正则化和 L2 正则化的最优参数来防止模型过拟合，所以得分有些下降。我们将各个参数的最优值输入到模型中去，以此得到最优模型，接下来用包含最优参数的 XGBoost 模型对预测集数据进行预测和精度评价。

3.3 模型精度评价

我们使用剩下的 20%数据作为验证集来验证模型的精度。不过这里一定要注意的是，对于银行而言，若将一个没有风险的客户判断为有风险的，银行只是损失了一个客户；但若将一个有风险的客户判断为没有风险的，这将面临贷款难以收回的后果。后者的危害远大于前者，所以在评价模型好坏时不能使用常用的错误率，这里我们推荐使 AUC 值和代价敏感错误率作为评价指标。

AUC 值实际就是 ROC 曲线的面积，它是一种独立于类别分布的评价指标，比较适用于评价类别不平衡学习问题，关于 AUC 的介绍已经很多了，这里不再详述。代价敏感错误率 E（以二分类为例）的定义如下：

真实类别	预测类别	
	第 0 类	第 1 类
第 0 类	0	$\cos t_{01}$
第 1 类	$\cos t_{10}$	0

$$E=\frac{1}{n}\left(\sum_{x_i\in D^+}\mathrm{I}(f(x_i)\neq y_i)\times\cos t_{01}+\sum_{x_i\in D^-}\mathrm{I}(f(x_i)\neq y_i)\times\cos t_{10}\right)$$

其中，n 为样本总数，D^+ 为属于第 0 类的样本数，D^- 为属于第 1 类的样本数，$\mathrm{I}(f(x_i)\neq y_i)$ 为 $f(x_i)\neq y_i$ 的样本数。我们这里将第 0 类表示为信用好的类别，第 1 类表示为信用差的类别，则将第 1 类的用户误判为第 0 类的后果比将第 0 类的误判为第 1 类的大得多，即 $\mathrm{cost}_{10}>\mathrm{cost}_{01}$。为简化模型，我们认为 cost_{10} 是 cost_{01} 的 k 倍，令 $\mathrm{cost}_{01}=1$，则 $\mathrm{cost}_{10}=k$。我们我们可以观察 k 的值变化导致 E 的值变化幅度大小，因为将第 0 类误判为第 1 类的错误只是 $\times 1$，而将第 1 类误判为第 0 类的错误 $\times k$，若 k 变化导致 E 变化较小，这说明将第 0 类误判为第 1 类的情况较多，而第 1 类误判为第 0 类的情况较少。反之亦然。

银行将有风险的客户判断为没有风险所带来的后果与银行将没有风险的客户判断为有风险所带来的后果的倍数关系（即 k 的取值）目前还没有统一的标准，我们只能取多个值来进行比较，这里取 k=3, 6, 10，然后计算 E 值变化与 k 值变化比值的平均值。预测结果的混淆矩阵如图 1 所示，其中，第 0 类表示信用好，第 1 类表示信用差。

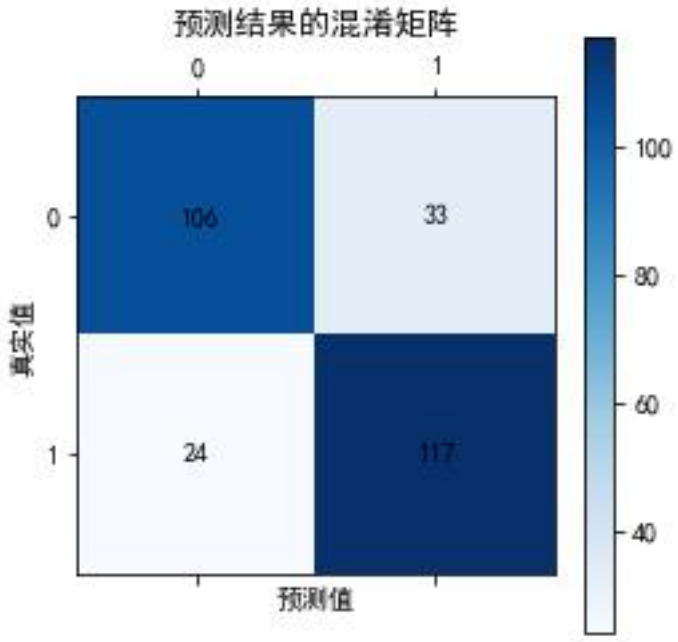


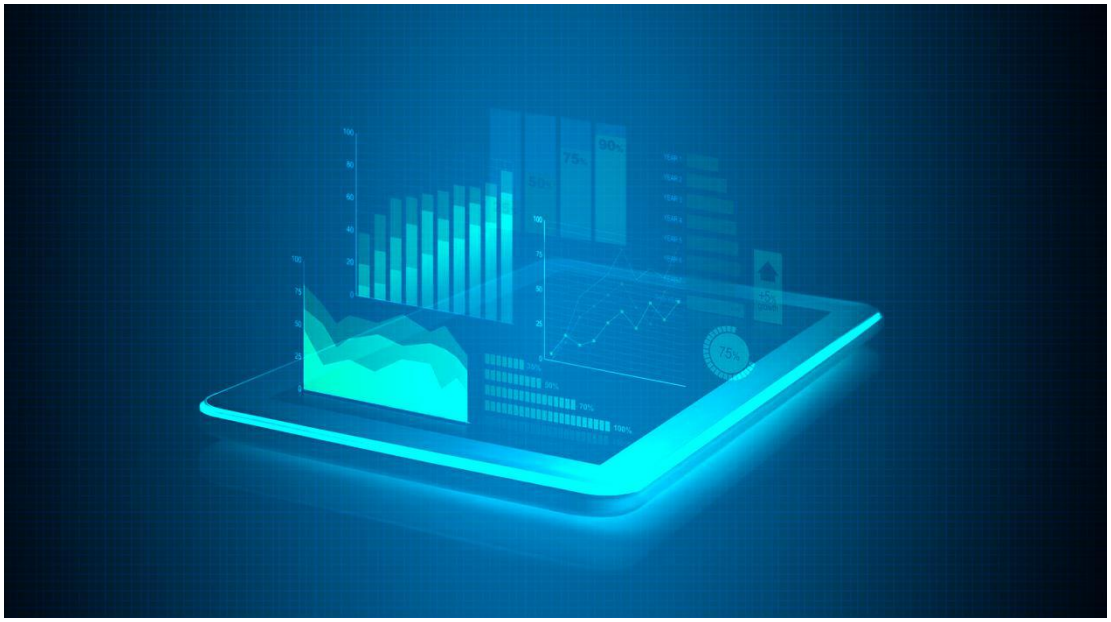
图 1 预测结果的混淆矩阵图

预测结果的 AUC 值和代价敏感错误率 E 的平均变化幅度如表 3 所示：

表 3 预测结果精度评价表

AUC	代价敏感错误率平均变化幅度
0.901	0.086

由此我们看到，XGBoost 模型的 AUC=0.901，这说明模型精度很好。随着 k 的变化，代价敏感错误率 E=0.086 变化幅度很小，这说明 XGBoost 的误判主要表现在将信用好的用户判断为信用差的，而将信用差的用户误判为信用差好的情况较少；并且从混淆矩阵也可以看出，模型将信用好的用户判断为信用差的误判人数比将信用差的误判为信用好的人数多了 9 个，这都说明模型在误判时更倾向于将信用好的误判为信用差的。综合来看，XGBoost 模型不论是从精度还是误判来看表现都很好。



4. 用户信用评分

由于通常情况下我们一般都是根据模型输出的用户类别的预测值来直接判断用户所属的类别，例如用户属于信用好或信用差。但模型在做判断时往往是根据用户所属类别的概率来判断的，例如用户属于信用好的类别概率为 0.55，属于信用不好的类别概率为 0.45，这时模型给出的结果是该用户信用好，但实际这样的判断在银行这种需要非常强的风险管控的的行业是有点武断的。所以在金融行业中我们不仅可以使用模型给出的用户的类别的信息，还可以将模型中用户所属类别的概率利用起来，将这一概率与银行的信用评分卡相结合，不仅判断用户的信用类别，还可以建立用户信用评分体系。

在用户信用评分体系中，设违约的概率为 p，则正常的概率为 1-p，违约与正常的概率比 $odds=\frac{p}{1-p}$，从而定义评分卡的刻度 $score$=A-Blog($odds$)

为求解 A 和 B，我们先做两个假设：（1）在某个特定的比率 $odds=\theta_0$ 时对应的分值为 P_0；（2）制定比率翻倍时分数的变化值为 ΔP。从而我们可以得到 $score$ 方程中的两个点，(θ_0,P_0) 和 $(2\theta_0,P_0+\Delta P)$，将这两个点带入 $score$ 方程中，有：

$$P_0=A-B\log(\theta_0)$$
$$P_0+\Delta P=A-B\log(2\theta_0)$$

从而，我们可以解出 $B=\frac{\Delta P}{\log(2)}, A=P_0+B\log(\theta_0)$。所以，只要确定了 θ_0,P_0 和 ΔP，$score$ 的值也就确定了。

这里，我们把由 XGBoost 模型计算的结果和评分卡模型结合来评价用户的信用分值。模型中往往认为用户属于信用好的类别概率>0.5 就判定其信用好。但用于银行业风险管控要求很高，我们这里把标准定得高一点：用户属于信用好的类别概率到达 70%以上（设θ_0=0.7）才认为基本达到信用好的要求，此时的分值为及格分 60（设P_0=60），当违约比上升一倍时，分值下降 5 分（设ΔP=5）。将θ_0=0.7、P_0=60 和ΔP=5 带入 *score*=A-Blog(*odds*)来计算信用分数。计算出的信用分值概况如表 4 所示，信用分值分布表 5 所示，其中风险客户比率值指 60 分以下的用户占比。

表 4 信用分值概况表

信用分值最小值	信用分值最大值	信用分值平均值	信风险客户比率
14	94	54.82	60%

表 5 信用分值表

分值区间	用户数量
[0, 20)	6
[20, 30)	28
[30, 40)	42
[40, 50)	35
[50, 60)	57
[60, 70)	43
[70, 80)	31
[80, 100)	38

参考资料

[1]Chawla N V , Bowyer K W , Hall L O , et al. SMOTE: Synthetic Minority Over-sampling Technique[J]. Journal of Artificial Intelligence Research, 2011, 16(1):321-357.

[2]Chen T , Guestrin C . XGBoost: A Scalable Tree Boosting System[C]// Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2016.

[3]周志华.机器学习[M]北京：清华大学出版社，2016

[4]（美）Mamdouh Refaat 著.信用风险评分卡研究[M].王奇松等译.北京：社会科学文献出版社 2013.