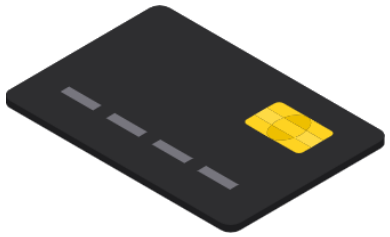
西安交通大學
XI'AN JIAOTONG UNIVERSITY

深度Boosting决策树算法在欺诈检测中的应用

答辩人：周梦豪　　指导老师：王尧

研究问题

欺诈检测会出现在很多管理学场景中，在不同场景中均会出现欺诈行为



a. 信用卡欺诈

b. 理赔欺诈

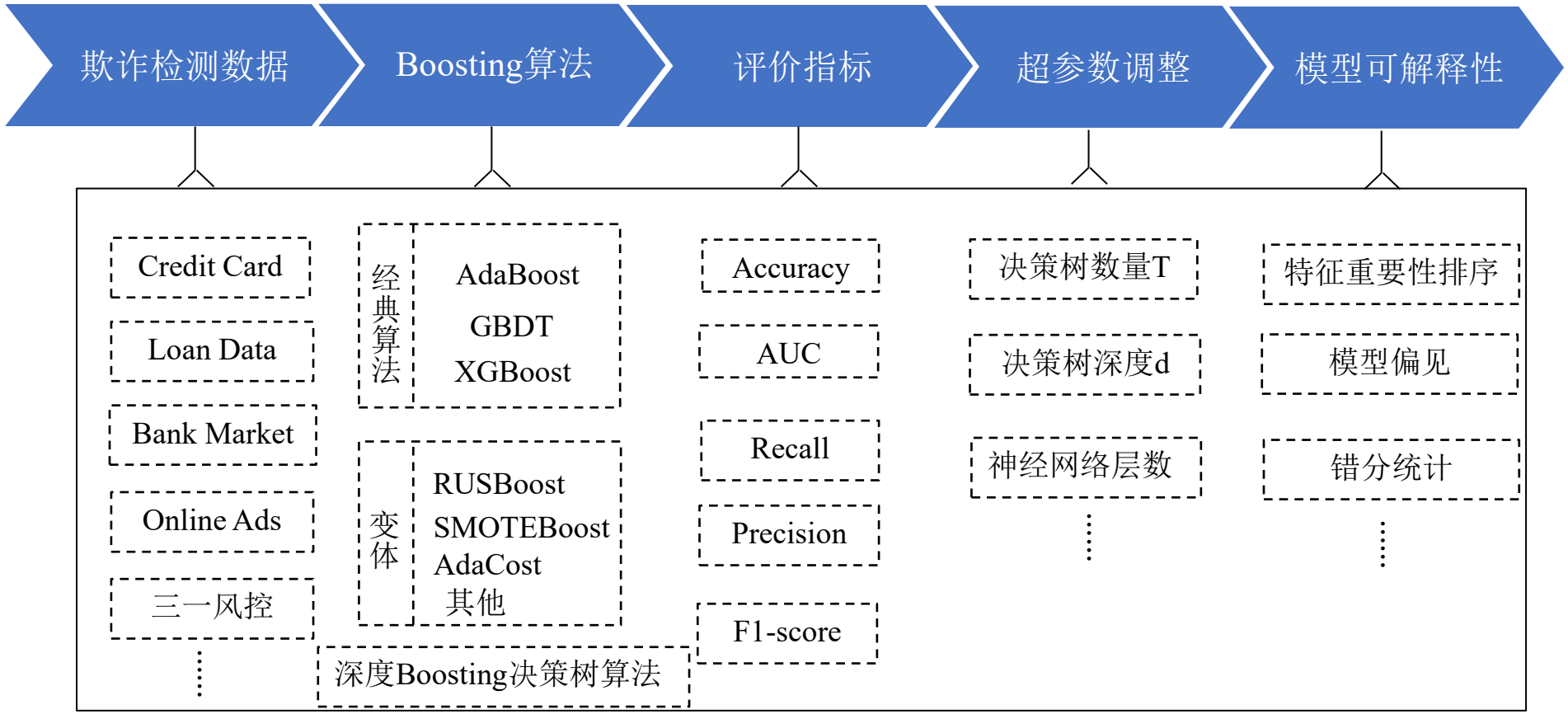
c. 在线广告点击流量欺诈

d. 网贷欺诈

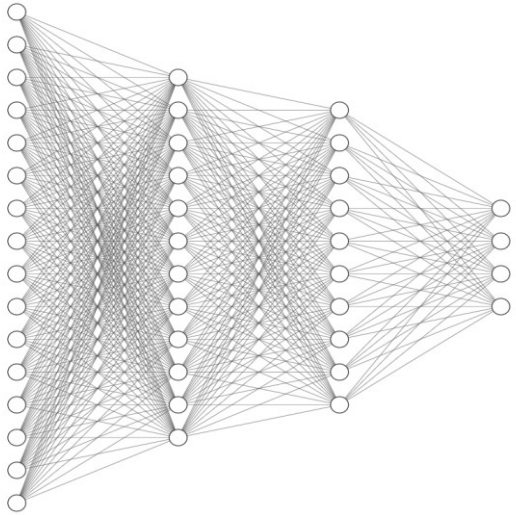
- 欺诈呈现出新的特点：专业化、产业化、隐蔽化、场景化
- 传统方法面临挑战：维度单一、时效性差、范围受限
- 机器学习方法显示出巨大应用潜力，成为新的研究思路

研究目的与思路

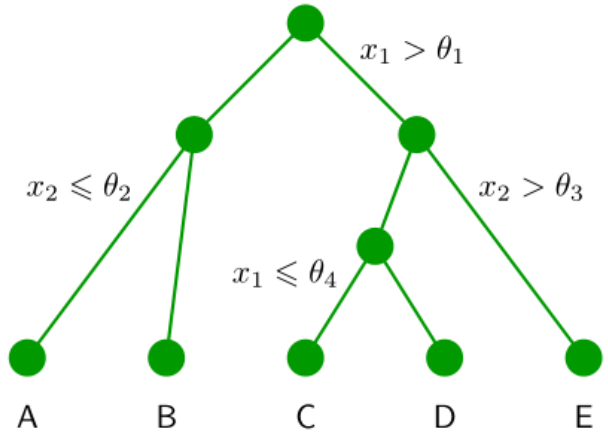
从**非均衡分类**角度探究**深度Boosting决策树**算法在**欺诈检测**中的应用



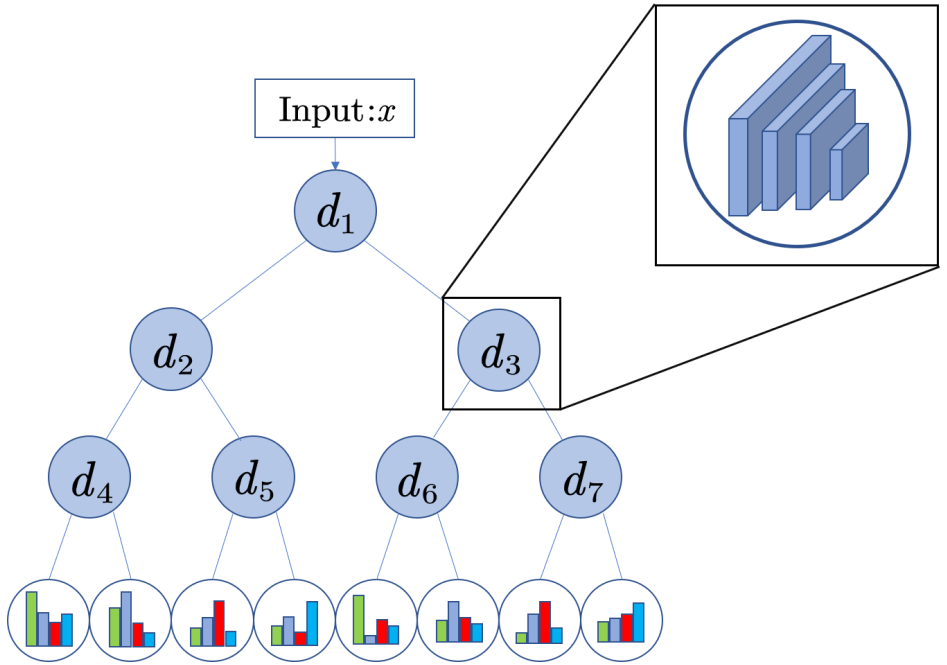
Soft Decision Tree



a. Deep Neural Network（泛化性能好，可解释性差）

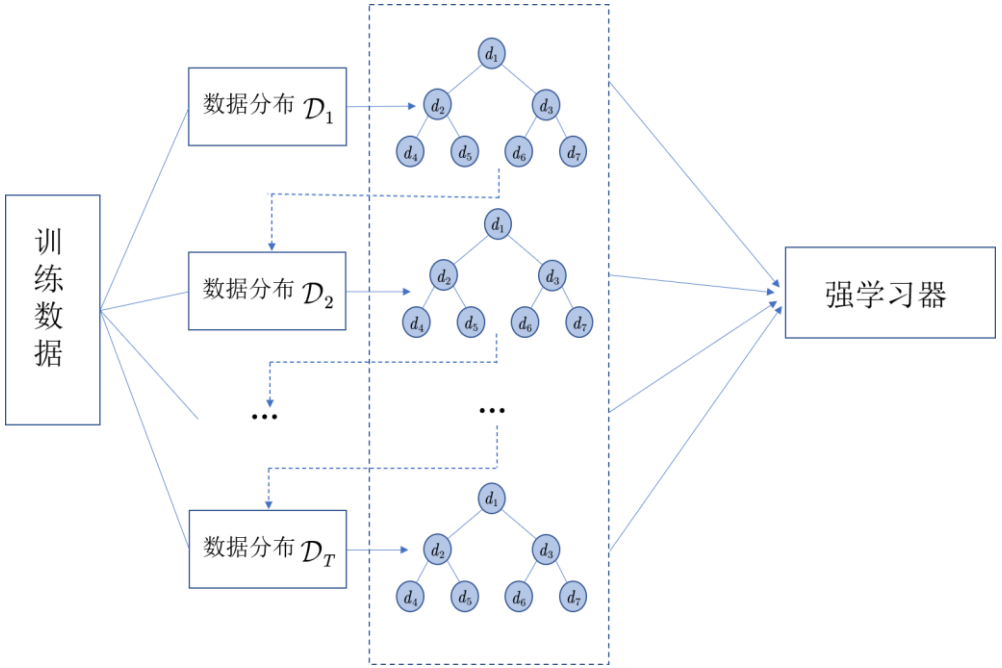


b. Decision Tree（可解释性好，泛化性能差）



c. Soft Decision Tree（兼具二者优点）

Boosting框架



a. Boosting框架

输入： 训练集$\mathcal{T}=\{(\boldsymbol{x}_1,y_1),\cdots,(\boldsymbol{x}_N,y_N)\},\boldsymbol{x}_i\in\mathcal{X}=\mathbb{R}^p,y_i\in\mathcal{Y}=\{-1,+1\}$;模型超参数：软决策树数量$T$、树的深度$d$、树节点内网络层数$c$、正则项系数$\lambda_1,\lambda_2$和迭代次数nEpochs

初始化数据权重 $D_i(\boldsymbol{x})\leftarrow\frac{1}{N}$ for $i=1,\cdots,N$
for $t=1\rightarrow T$ **do**
  随机初始化$^{\circledR}\Theta$
  for $i=1\rightarrow$ nEpochs **do**
    将$\mathcal{T}$拆分成小批量
    for all minibatch from $\mathcal{T}$ **do**
      通过SGD更新Θ
    end for
  end for
  输出软决策树h_t
  $\epsilon_t\leftarrow\mathbb{E}_{x\sim\mathcal{D}_t}[h_t(x)\neq y]$
  $\alpha_t\leftarrow\frac{1}{2}\log\frac{1-\epsilon_t}{\epsilon_t}$
  $\mathcal{D}_{t+1}(x)\leftarrow\mathcal{D}_t(x)\exp(-\alpha_t h_t(x)y)$
  $\mathcal{D}_{t+1}(\boldsymbol{x})\leftarrow\frac{\mathcal{D}_{t+1}(\boldsymbol{x})}{\sum_{\boldsymbol{x}}\mathcal{D}_{t+1}(\boldsymbol{x})}$
end for

输出： 预测模型$H(x)=\arg\max_{y\in\{-1,+1\}}P(f(x)=y|x)$

b. 深度Boosting决策树算法学习过程伪代码

- 数据重构：加大错分样本权重，减小分对样本权重
- 结合策略：准确的分类器权重高，不准确的分类器权重低

算法数学描述

Inner Nodes

$$d_i(\boldsymbol{x};\Theta)=\sigma(\boldsymbol{w}_i^T\boldsymbol{x}+b_i)$$

Leaf Nodes

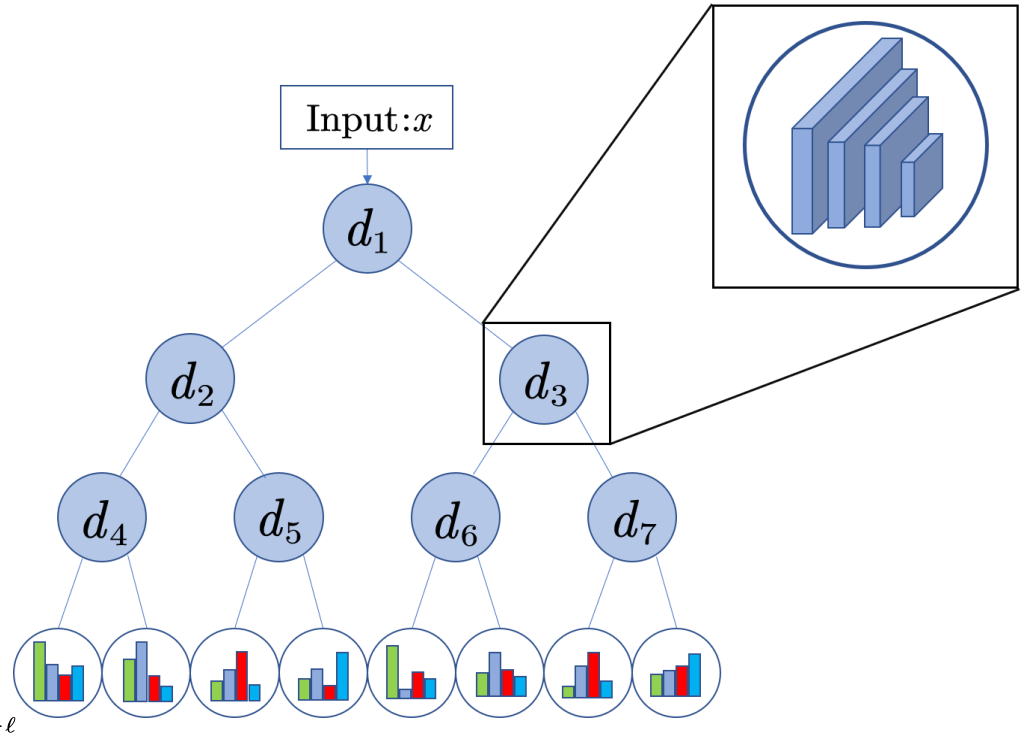
$$Q_k^\ell=\frac{\exp(\phi_k^\ell)}{\sum_{k'}\exp(\phi_{k'}^\ell)}$$

路径概率

$$\pi_i(\boldsymbol{x}|\Theta)=\sum_{1\leq j<i}d_j(\boldsymbol{x};\Theta)^{\mathbb{I}_j^r}(1-d_j(\boldsymbol{x};\Theta))^{\mathbb{I}_j^\ell}$$

预测结果

$$\mathbb{P}[y=k|\boldsymbol{x},\Theta]=\sum_\ell Q_k^\ell\pi_\ell(\boldsymbol{x}|\Theta)$$



c. Soft Decision Tree

算法数学描述

训练数据：$\mathcal{D}=\{(\boldsymbol{x}_i,y_i)\}\,(|\mathcal{D}|=N,\boldsymbol{x}_i\in\mathbb{R}^p,y_i\in\{-1,+1\})$

目标函数：$(\phi_t,h_t)=\arg\min_{\phi,h}\sum_i^N L\left(y_i,f_{t-1}(x_i)+\phi_t h_t(x_i)\right)+C_t+\Omega_t(w)$

正则化项1：$C_t=-\lambda_1\times 2^{-d}\sum_{i\in Inner\ Nodes}0.5\log(\alpha_i)+0.5\log(1-\alpha_i)$

正则化项2：$\Omega_t=\lambda_2\sum_{i\in Inner\ Nodes}\|w\|_2$

损失函数：$L(f(x_i),y_i)=y_i\exp(-y_i f(x_i))$

优化算法：Adam

超参数：λ_1、λ_2、软决策树数量T、树的深度d、神经网络层数c

数据集介绍

a. 数据集基本统计情况

数据集	样本数	特征数	类别数	少数类占比(%)
Credit Card	26107	17	2	5.3
Loan Data	95791	13	2	1.6
Bank Market	41189	17	2	11.2

- 数据集1：Credit Card Fraud

  包含由欧洲持卡人于2013年9月使用信用卡在两天内发生的交易

- 数据集2：Loan Data

  一家美国互联网金融公司Lending Club在2007年到2015年的一些业务数据

- 数据集3：Bank Marketing

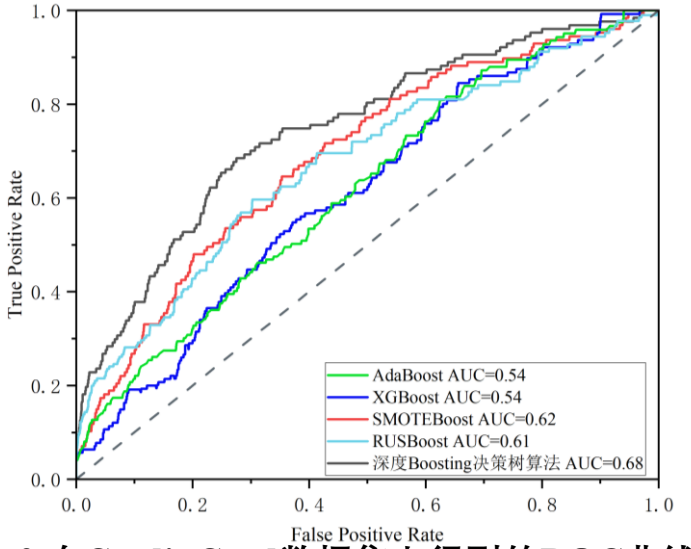
  一家银行从2008年5月到2010年10月的交易数据

数据来源：
①https://mlg.ulb.ac.be/wordpress/projects/
②https://www.kaggle.com/swetashetye/lending-club-loan-data-imbalance-dataset
③https://archive.ics.uci.edu/ml/datasets/Bank+Marketing

在数据集上结果

a1. 在Credit Card数据集上结果比较

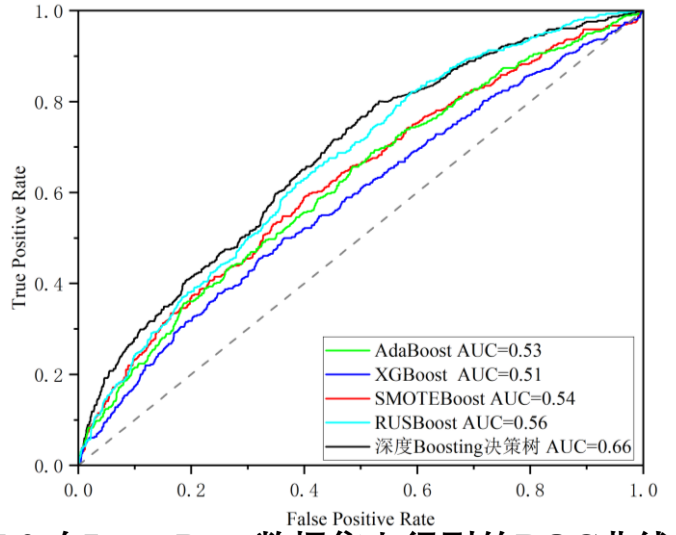
模型	指标 1	指标 2	指标 3		
	Accuracy	AUC	*Precision*	*Recall*	*F1*
1) AdaBoost	98.1%	0.54	100%	5%	0.09
2) XGBoost	98.4%	0.54	100%	5%	0.09
3) SMOTEBoost	93.5%	0.62	4%	13%	0.06
4) RUSBoost	89.1%	0.61	3%	54%	0.05
5) 深度 Boosting 决策树	98.3%	***0.68***	67%	5%	0.09



a2.在Credit Card数据集上得到的ROC曲线

b1. 在Loan Data数据集上结果比较

模型	指标 1	指标 2	指标 3		
	Accuracy	AUC	*Precision*	*Recall*	*F1*
1) AdaBoost	83.5%	0.51	43%	8%	0.14
2) XGBoost	83.8%	0.53	58%	2%	0.03
3) SMOTEBoost	81.0%	0.54	30%	14%	0.19
4) RUSBoost	79.0%	0.56	21%	46%	0.29
5)深度 Boosting 决策树	83.9%	***0.66***	69%	4%	0.08

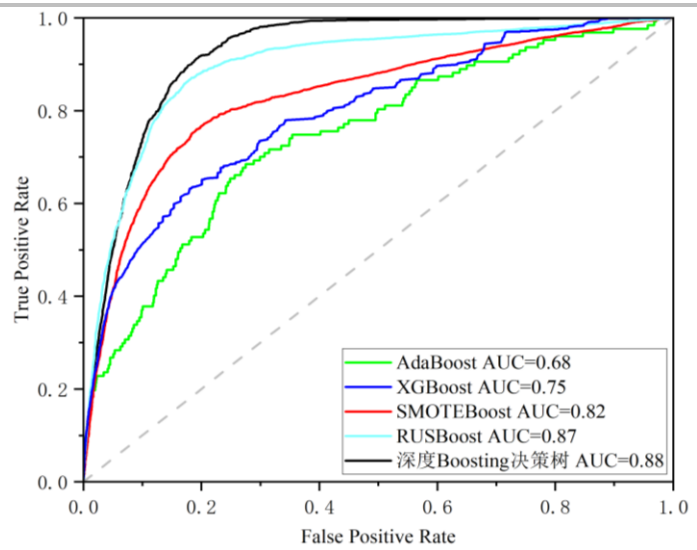


b2.在Loan Data数据集上得到的ROC曲线

在数据集上结果

c1. 在Bank Marketing数据集上结果比较

模型	指标 1	指标 2	指标 3		
	Accuracy	AUC	*Precision*	*Recall*	*F1*
1) AdaBoost	90.8%	0.68	66%	38%	0.49
2) XGBoost	91.9%	0.75	68%	53%	0.60
3) SMOTEBoost	89.7%	0.82	53%	73%	0.53
4) RUSBoost	86.1%	0.87	44%	90%	0.59
5)深度 Boosting 决策树	90.9%	***0.88***	88%	77%	0.82



c2.在Bank Marketing数据集上得到的ROC曲线

1) Friedman检验结果

原假设H_0：所有算法性能相同

检验统计量$\tau_F = \dfrac{(N-1)\tau_{\chi^2}}{N(k-1)-\tau_{\chi^2}} \sim \chi^2(k-1)$

在显著性水平α分别为0.05和0.1条件下拒绝原假设H_0

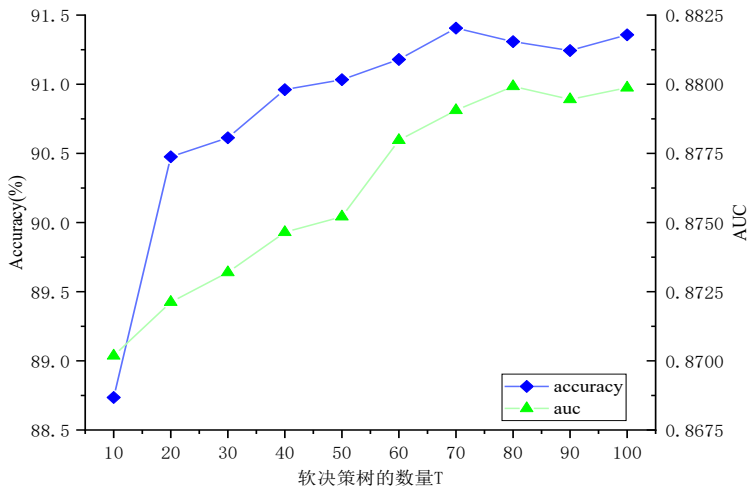
2) Nemenyi检验结果

平均序值差别的临界值域$CD = q_\alpha \sqrt{\dfrac{k(k+1)}{6N}}$

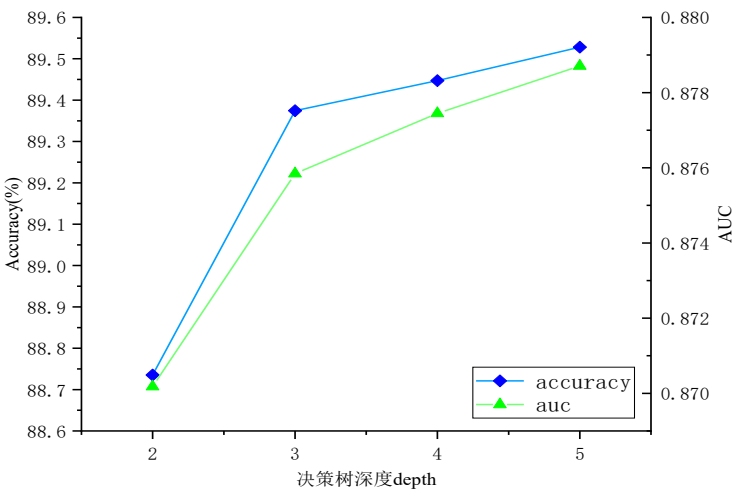
在显著性水平α分别为0.05和0.1条件下算法分两类：

① 深度Boosting决策树、RUSBoost和SMOTEBoost
② AdaBoost、XGBoost

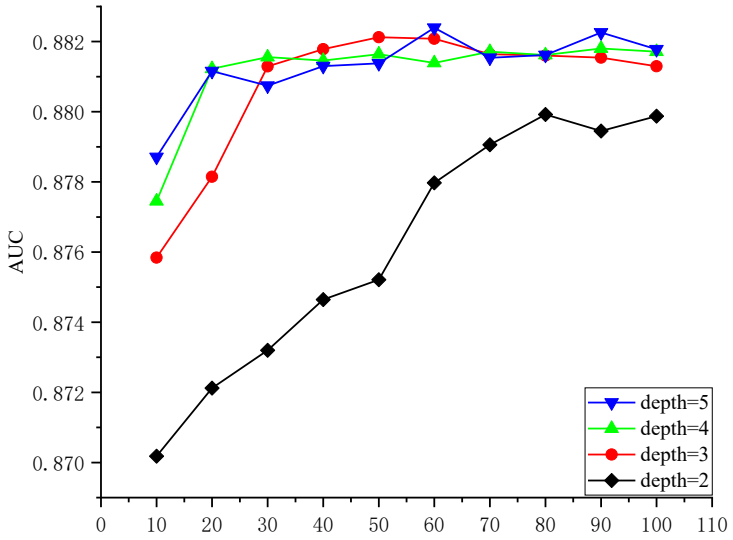
超参数调整-树的深度depth与树的数量T



a. 树的数量对模型性能的影响



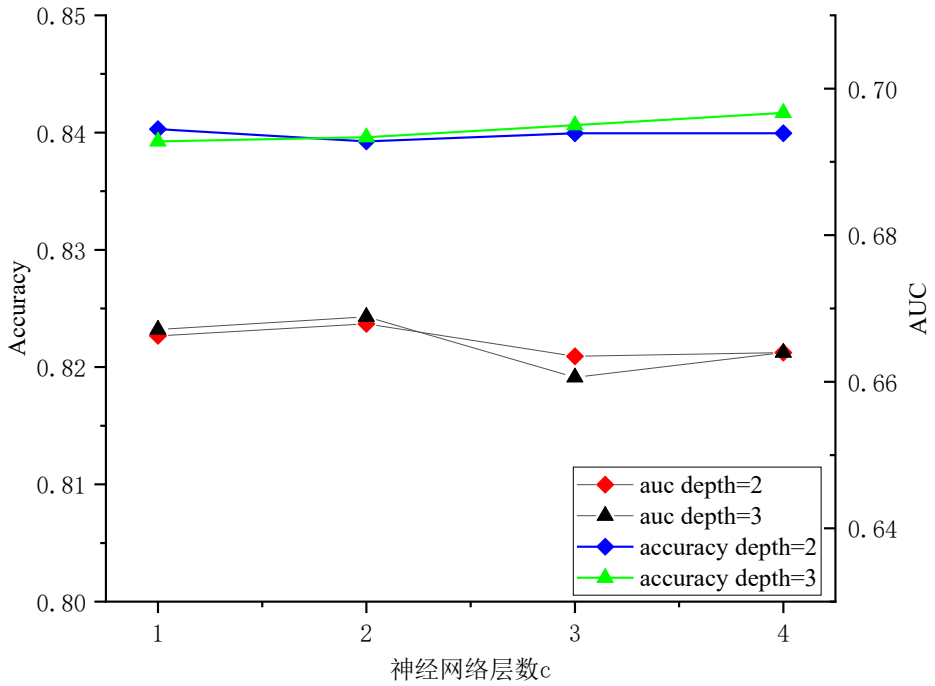
b. 树的深度对模型性能的影响



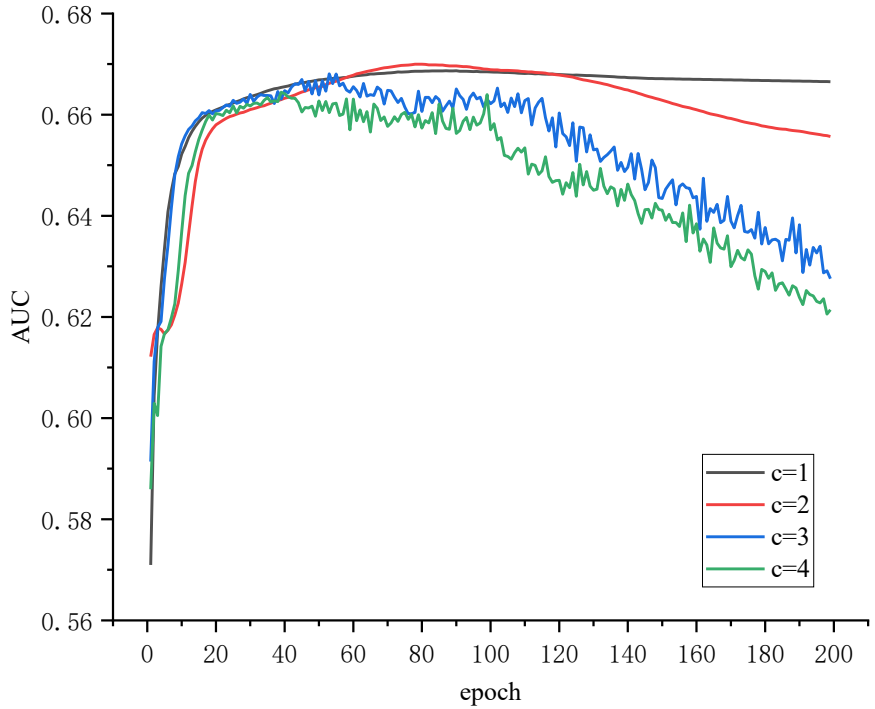
c. 树的数量与深度对模型性能的影响

- 增加树的数量与深度均能提升模型性能
- 深度越大，达到模型上限所需要的软决策树数量越少

超参数调整-神经网络的层数c



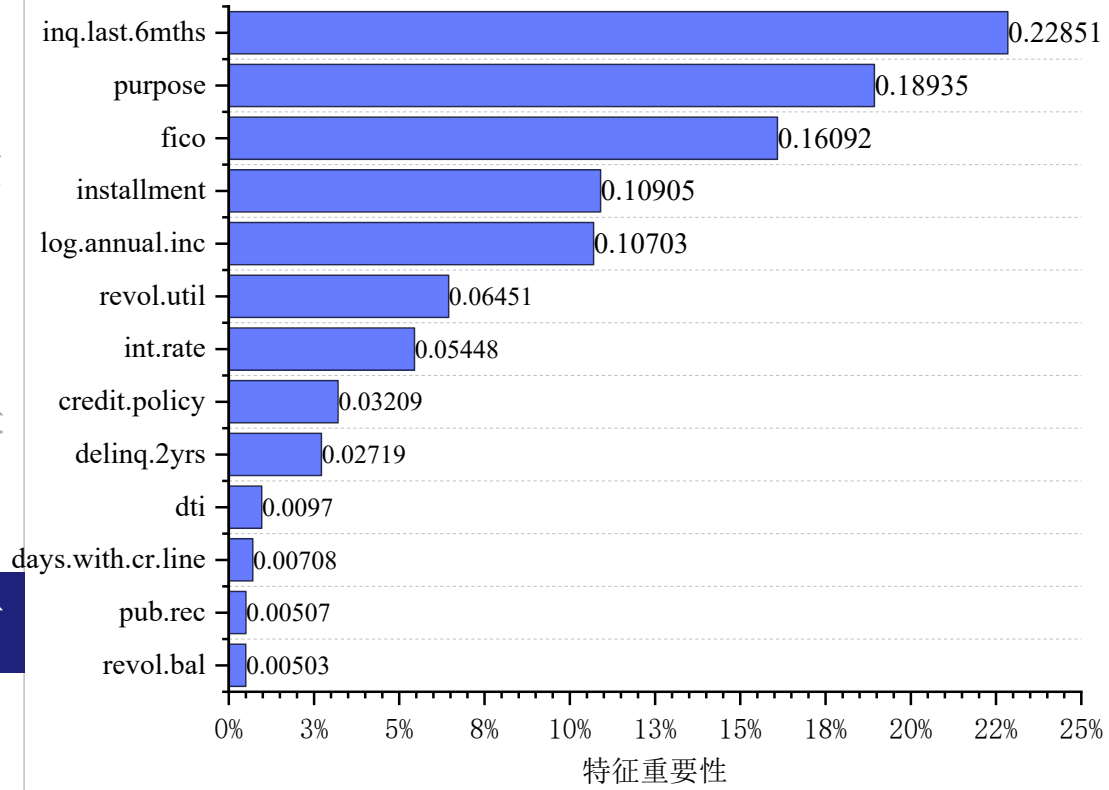
a. 增加网络层数对性能提升的影响



b. 增加网络层数造成的问题

- 针对该任务，增加层数无法显著提升模型性能(非表格数据：图像？文本？音频？视频？)
- 增加层数容易导致模型过拟合

特征重要性排序



a. 在Loan Data数据集上获得的特征重要性排序结果

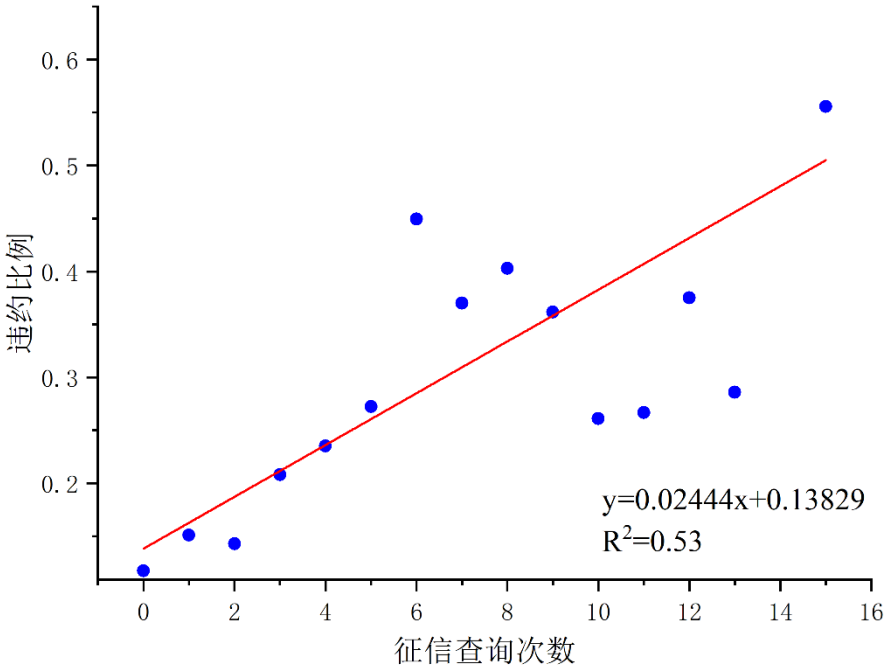
特征名称	特征含义
inq.last.6mths	借款人在过去 6 个月征信被查询的次数
purpose	贷款目的：信用卡、债务合并、教育、家庭装修、小生意、其他
fico	借款人的 FICO 信用评分
installment	借款人分期付款金额
log.annual.inc	借款人年收入的自然对数
revol.util	借款人的循环余额利用率(使用的信贷额度相对于可用信贷总额)
int.rate	贷款利率
credit.policy	1-满足公司担保标准；0-否
delinq.2yrs	借款人在过去 2 年内逾期还款 30 天以上的次数
dti	借款人的债务-收入比率
days.with.cr.line	借款人获得信用额度的天数
pub.rec	借款人的负面公共记录数量(破产申请、税收留置权或判决)
revol.bal	借款人的循环余额(信用卡账单周期结束时未付的金额)

b. 特征含义

特征重要性排序前5名分别为

inq.last.6mths(22.8%)>purpose(18.9%)>fico(16.1%)>installment(10.9%)>log.annual.inc(10.7%)

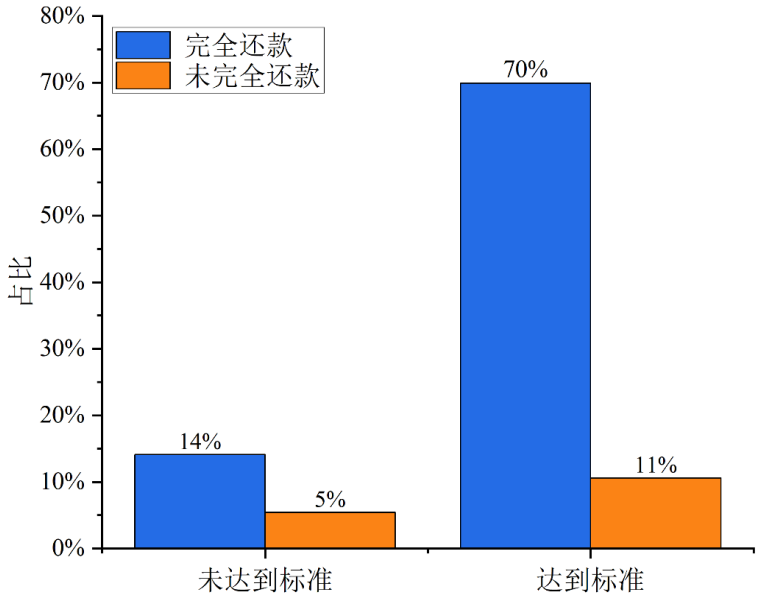
特征排序结果分析



a. 征信查询次数与违约比例线性拟合结果

inq.last.6mths(22.8%)

- 借款人一段时期以内借款活动越频繁借款人违约风险越大



b. 是否达到公司借款标准中的真实还款情况

credit.policy(3.2%)

- 未达到标准(19%)，14%的借款人会完全还款
- 该群体被忽略，现有标准无法有效分类

结论与展望

- **结论**

1. 深度Boosting决策树算法在欺诈检测中效果优于现有的Boosting算法
2. 深度Boosting决策树算法兼具泛化性能好和可解释性好的优点，二者不矛盾
3. 构建具有可解释性的机器学习模型的一条可行路径：复杂模型与简单模型融合

- **未来研究方向**

1. 探究不同优化算法对模型性能的影响
2. 结合非均衡问题处理技术，对现有算法进行改进
3. 从模型偏见、错分统计等角度对模型可解释性进行探索

西安交通大学
XI'AN JIAOTONG UNIVERSITY

请各位老师批评指正

答辩人：周梦豪　　指导老师：王尧