基于机器学习的贷款违约预测

Datawhale September

StarLEE

1. 背景

个人资产业务，即个人信贷业务。主要指运用从负债业务筹集的资金，将资金的使用权在一定期限内有偿让渡给个人，并在贷款到期时收回资金本息以取得收益的业务。个人资产业务作为商业银行主要的收益来源，对商业银行的经营成果起着重要的作用。互联网技术和移动通信技术的不断创新以及与金融行业的快速融合，使得互联网金融中的重要发展模式——网络借贷得到了飞速的发展。网络借贷是指互联网公司利用互联网技术和移动通信技术搭建一个线上平台，为资金提供者和资金需求者提供直接的资金融通服务。这种基于网络平台的借贷方式相较于银行借贷，不仅手续简便、效率高，且允许信用贷款，为自有资本不足、缺乏担保的小额借款人提供了新的出路。它的出现满足了人们多元化的投融资需求，提高了社会闲散资金的利用率。

在以往，商业银行对贷款用户进行信用风险评估时，往往依靠风控人员依赖 5C 分类法来主观判断，从其个人品格、信用额度、偿付能力、市场经济状况这 5 个因素对贷款用户进行判断和权衡，以此作为是否贷款给该用户的参考，决定是否发放贷款，这种依赖主观判断的方法显然效率低下，而且评估的时候十分依赖风控人员的主观判断能力，从公司内部管控的角度来看，甚至存在风控人员内部作弊的可能性，不能适应市场经济的快速发展，满足贷款用户的需求，也不能满足网贷平台风险管理的需求。面对数以万计甚至是数以十万计的申请借款的用户时，网贷平台则需要采用各种机器学习的方法来减少监控与检测过程中的人工参与部分，利用自动化的方法提高放款审核的准确率和效率。

目前，网贷行业运用大数据技术进行风险控制管理已经取得了一定的成效，比较成熟的产品有 Zest Finance 公司所开发的基于数据挖掘和机器学习理论的分析模型，以及美国使用最广泛的个人信用评分系统——FICO 信用评分，都是美国借贷行业贷款决策的重要参考标准。而在国内，2014 年 10 月，宜人贷将采用了大数据技术的“极速模式”添加进“宜人贷借款”APP 中，积木盒子根据建立的“读秒”标准判断用户的信用等级，拍拍贷于 2015 年推出的魔镜风控系统被认为是行业内首个基于大数据的风控模型，首个能准确预测借款标的风险概率的风控系统，爱钱进在其两周年发布会上，推出了基于机器学习、深度学习等技术的全新风控体系——“云图动态风控系统”。由此可见，机器学习的蓬勃兴起使网络借贷平台利用多维大数据构建智能风控模型，更加准确的评估个人信用状况，有效地降低违约风险。

2. 数据理解

2.1 变量信息

训练集数据包含了 800000 个观测值以及 47 个变量（包括 15 列匿名变量），变量字段表如下：

表 2.1 变量字段表

Field	Description	Field	Description
id	为贷款清单分配的唯一信用证标识	dti	债务收入比
loanAmnt	贷款金额	delinquency_2years	借款人过去 2 年信用档案中逾期 30 天以上的违约事件数
term	贷款期限（year）	ficoRangeLow	借款人在贷款发放时的 fico 所属的下限范围
interestRate	贷款利率	ficoRangeHigh	借款人在贷款发放时的 fico 所属的上限范围
installment	分期付款金额	openAcc	借款人信用档案中未结信用额度的数量
grade	贷款等级	pubRec	贬损公共记录的数量
subgrade	贷款等级之子级	pubRecBankruptcies	公开记录清除的数量
employmentTitle	就业职称	revolBal	信贷周转余额合计
employmentLength	就业年限（年）	revolUtil	循环额度利用率，或借款人使用的相对于所有可用循环信贷的信贷金额
homeownership	借款人在登记时提供的房屋所有权状况	totalAcc	借款人信用档案中当前的信用额度总数
annuallncome	年收入	initialListStatus	贷款的初始列表状态
verificationStatus	验证状态	applicationType	表明贷款是个人申请还是与两个共同借款人的联合申请
issueDate	贷款发放月份	earliesCreditLine	借款人最早报告的信用额度开立的月份
purpose	借款人在贷款申请时的贷款用途类别	title	借款人提供的贷款名称
postCode	借款人在贷款申请中提供的邮政编码的前 3 位数字	policyCode	公开可用的策略_代码=1 新产品不公开可用的策略_代码=2
regionCode	地区编码	n 系列匿名特征	匿名特征 n0-n14，为一些贷款人行为计数特征的处理

2.2 评价指标

2.2.1 混淆矩阵 Confuse Matrix

- True Positive：实例为正，预测为正；

- False Negative：实例为正，预测为负；
- False Positive：实例为负，预测为正；
- True Negative：实例为负，预测为负。

2.2.2 准确率 Accuracy

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Accuracy 不适用于样本分布不均衡的情况（正样本过多或负样本过多）。

2.2.3 精确率 Precision

$$Precision = \frac{TP}{TP + FP}$$

Precision 为所有预测为正样本中真实为正的占比。

2.2.4 召回率 Recall

$$Recall = \frac{TP}{TP + FP}$$

Recall 为所有正样本中被预测为正的占比。

2.2.5 F1 Score

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

Precision 和 recall 是相互影响的，精确率提高则召回率下降。F1 Score 作为精确率和召回率的调和平均数，很好地综合了两者的信息。

2.2.6 P-R 曲线（Precision-Recall Curve）

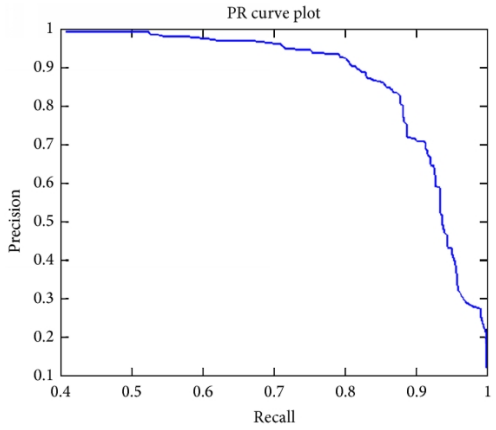


图 2.1 P-R 曲线示意图

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

Precision 和 recall 是相互影响的，精确率提高则召回率下降。F1 Score 作为精确率和召回率的

调和平均数，很好地综合了两者的信息。图 2.1 展示了 P-R 曲线样例，可见 Precision 和 Recall 存在反相关关系。

2.2.7 ROC（Receiver Operating Characteristic）

ROC（receiver operating characteristic）曲线说明了二元分类器系统的鉴别阈值变化时的诊断能力，它是根据一系列不同的二分类方式（分界值或决定阈），以真阳性率（TPR，正样本中预测为正占比）为 y 轴，假阳性率（FPR，负样本中预测为正占比）为 x 轴绘制而成，样例如图 2.2 所示。

$$TPR = 2 \cdot \frac{TP}{TP + FN}$$

$$FPR = 2 \cdot \frac{FP}{FP + TN}$$

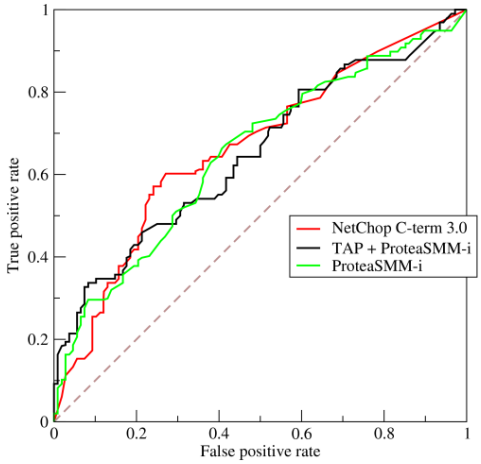


图 2.2 ROC 示意图

2.2.8 AUC（Area Under Curve）

AUC（Area Under Curve）被定义为 ROC 曲线与坐标轴围成的面积。使用 AUC 值作为模型的评价标准是因为很多时候 ROC 曲线并不能清晰地说明哪个分类器效果更好，而作为一个数值，对应 AUC 更大的分类器效果更好。AUC 就是衡量学习器优劣的一种性能指标。

在金融风控领域中，K-S（Kolmogorov-Smirnov）曲线与 ROC 曲线类似但稍有区别：K-S 曲线将 TPR 和 FPR 作为 y 轴，以选定的阈值作为 x 轴，公式为：$KS = \max(TPR - FPR)$；KS 值小于 0.2 认为模型没有区分能力；KS 值处于[0.2, 0.3]认为模型有一定区分能力；KS 值处于[0.3, 0.5]认为模型有较强区分能力；KS 值大于 0.75 认为模型异常。

2.3 数据读取及评价指标实现

CSV 文件数据的读取可由 Pandas 轻松实现，而评价指标都可以在 sklearn 中找到已有的 api。数据读取结果如图 2.3 所示，调用 api 实现评价指标样例结果如图 2.4~2.6 所示：

```
import pandas as pd
```

```
train = pd.read_csv('../train.csv')
testA = pd.read_csv('../testA.csv')
print('Train data shape:',train.shape)
print('TestA data shape:',testA.shape)
```

```
Train data shape: (800000, 47)
TestA data shape: (200000, 48)
```

```
train.head()
```

	id	loanAmnt	term	interestRate	installment	grade	subGrade	employmentTitle	employmentLength	homeOwnership	...	n5	n6	n7	n8	n9	n1
0	0	35000.0	5	19.52	917.97	E	E2	320.0	2 years	2	...	9.0	8.0	4.0	12.0	2.0	7.
1	1	18000.0	5	18.49	461.90	D	D2	219843.0	5 years	0	...	NaN	NaN	NaN	NaN	NaN	13.
2	2	12000.0	5	16.99	298.17	D	D3	31698.0	8 years	0	...	0.0	21.0	4.0	5.0	3.0	11.
3	3	11000.0	3	7.26	340.96	A	A4	46854.0	10+ years	1	...	16.0	4.0	7.0	21.0	6.0	9.
4	4	3000.0	3	12.99	101.07	C	C2	54.0	NaN	1	...	4.0	9.0	10.0	15.0	7.0	12.

5 rows × 47 columns

图 2.3 训练集及测试集数据读取

```
## 混淆矩阵
import numpy as np
from sklearn.metrics import confusion_matrix
y_pred = [0, 1, 0, 1, 0, 1, 1, 0]
y_true = [0, 1, 1, 0, 1, 0, 1, 0]
print('混淆矩阵:\n',confusion_matrix(y_true, y_pred))
```

```
混淆矩阵:
 [[2 2]
 [2 2]]
```

```
## accuracy
from sklearn.metrics import accuracy_score
y_pred = [0, 1, 0, 1, 0, 1, 1, 0]
y_true = [0, 1, 1, 0, 1, 0, 1, 0]
print('ACC:',accuracy_score(y_true, y_pred))
```

```
ACC: 0.5
```

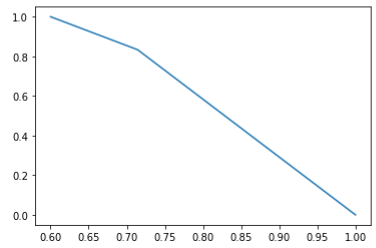
```
## Precision,Recall,F1-score
from sklearn import metrics
y_pred = [0, 1, 0, 1, 0, 1, 1, 0]
y_true = [0, 1, 1, 0, 1, 0, 1, 0]
print('Precision',metrics.precision_score(y_true, y_pred))
print('Recall',metrics.recall_score(y_true, y_pred))
print('F1-score:',metrics.f1_score(y_true, y_pred))
```

```
Precision 0.5
Recall 0.5
F1-score: 0.5
```

图 2.4 混淆矩阵，准确率，精确率，召回率及 F1-score 调用样例

```
## P-R曲线
import matplotlib.pyplot as plt
from sklearn.metrics import precision_recall_curve
y_pred = [0, 1, 1, 0, 1, 1, 0, 1, 1, 1]
y_true = [0, 1, 1, 0, 1, 0, 1, 1, 0, 1]
precision, recall, thresholds = precision_recall_curve(y_true, y_pred)
plt.plot(precision, recall)
```

```
[<matplotlib.lines.Line2D at 0x2c6d2598040>]
```




```
## ROC曲线
from sklearn.metrics import roc_curve
y_pred = [0, 1, 1, 0, 1, 1, 0, 1, 1, 1]
y_true = [0, 1, 1, 0, 1, 0, 1, 1, 0, 1]
FPR, TPR, thresholds=roc_curve(y_true, y_pred)
plt.title('ROC')
plt.plot(FPR, TPR,'b')
plt.plot([0,1],[0,1],'r—')
plt.ylabel('TPR')
plt.xlabel('FPR')
```

```
Text(0.5, 0, 'FPR')
```

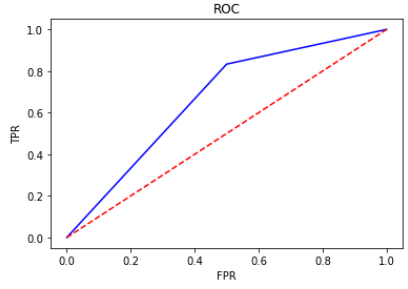


图 2.5 P-R，ROC 曲线调用样例

```python
## AUC
import numpy as np
from sklearn.metrics import roc_auc_score
y_true = np.array([0, 0, 1, 1])
y_scores = np.array([0.1, 0.4, 0.35, 0.8])
print('AUC socre:',roc_auc_score(y_true, y_scores))
```

```
AUC socre: 0.75
```

```python
## KS值 在实际操作时往往使用ROC曲线配合求出KS值
from sklearn.metrics import roc_curve
y_pred = [0, 1, 1, 0, 1, 1, 0, 1, 1, 1]
y_true = [0, 1, 1, 0, 1, 0, 1, 1, 1, 1]
FPR,TPR,thresholds=roc_curve(y_true, y_pred)
KS=abs(FPR-TPR).max()
print('KS值：',KS)
```

```
KS值： 0.5238095238095237
```

图 2.6 AUC，KS 值调用样例