**人工智能训练师（三级）操作技能考核**

**试题单**

准考证号：

试题代码：

试题名称：智能信用评分Logistic回归模型开发与测试

考核时间：20min

**1.**场地设备要求

（1）人工智能训练师主机 1 台；

（2）Python 编译环境；

（3）Finance数据集。

**2.**工作任务

互联网金融飞速发展，使得个人金融理财变得越来越容易。而其中信用评分技术是一种对贷款申请人（信用卡申请人）做风险评估分值的统计模型，可以根据客户提供的资料、客户的历史数据、第三方平台数据（芝麻分、京东、微信等），对客户的信用进行评估。现要求根据提供的finance数据集，补全2.2.1.ipynb代码。选择合适的特征，开发一个申请的评分模型，利用测试工具对模型进行测试，并对测试结果进行分析，完成测试报告，并运用工具对错误原因进行纠正。

（1）正确加载数据集，显示前五行的数据。

（2）使用Logistic模型进行模型训练，要求设定自变量和因变量，并根据自变量特征进行模型训练，最终将训练好的模型以文件名2.2.1\_model.pkl保存到考生文件夹，结果文件以2.2.1\_results.txt保存到考生文件夹。

（3）使用测试工具对模型进行测试，并记录测试结果，命名2.2.1\_report.txt，保存到考生文件夹

（4）对测试结果进行详细分析，并编写测试报告，包括模型性能评估、错误分析及改进建议，将答案写到答题卷文件中，答题卷文件命名为“2.2.1.docx”，保存到考生文件夹。

（5）运用工具分析算法中错误案例产生的原因并进行纠正，重新得到模型训练结果，以文件名2.2.1\_results\_xg.txt保存到考生文件夹。

（6）将以上代码以及运行结果，以html格式保存并命名为2.2.1.html，保存到考生文件夹，考生文件夹命名为“准考证号+身份证后6位”。

数据集说明：

Unnamed: 0 - 索引号。

SeriousDlqin2yrs - 个人在过去两年内是否出现过严重的拖欠（1 表示有严重拖欠，0 表示没有）。

RevolvingUtilizationOfUnsecuredLines - 这是指个人未偿还的信用额度与总信用额度的比例。

age - 客户的年龄。

NumberOfTime30-59DaysPastDueNotWorse - 在过去一段时间内，贷款逾期30至59天的次数。

DebtRatio - 债务比率。

MonthlyIncome - 客户的月收入。

NumberOfOpenCreditLinesAndLoans - 正在使用的信贷账户或贷款的数量。

NumberOfTimes90DaysLate - 贷款逾期超过90天的次数。

NumberRealEstateLoansOrLines - 持有的房地产相关贷款或信贷的数量。

NumberOfTime60-89DaysPastDueNotWorse - 贷款逾期60至89天的次数。

NumberOfDependents - 家庭中依赖该个人的人数。

**3.**技能要求

(1) 能维护日常训练集与测试集。

(2) 能使用工具对算法进行训练。

(3) 能使用测试工具对人工智能产品的使用进行测试。

(4) 能对测试结果进行分析，编写测试报告。

(5) 能运用工具，分析算法中错误案例产生的原因并进行纠正。

**4.**质量指标

(1) 深入理解业务，训练符合业务需求的模型。

(2) 数据预处理步骤完整，方法选择合理。

(3) 代码实现正确，结果符合预期。

(4) 测试结果分析全面，报告详细。

**人工智能训练师（三级）操作技能考核**

**试题评分表**

准考证号：

试题代码：

试题名称：智能信用评分Logistic回归模型开发与测试

考核时间：20min

测量分评分表

| 细则编号 | 配分 | 评分细则描述 | 规定或  标称值 | 结果或  实际值 | 得分 |
| --- | --- | --- | --- | --- | --- |
| M1 | 2 | 正确加载数据集，得1分，显示前五行的数据，得1分，总计得2分； | 根据数据 |  |  |
| M2 | 1 | 训练集与测试集的准确划分得1分； | 根据数据 |  |  |
| M3 | 2 | 初始化 Logistic 回归模型得2分； | 根据数据 |  |  |
| M4 | 2 | 训练 Logistic 回归模型得2分； | 根据数据 |  |  |
| M5 | 1 | 正确保存Logistic 回归模型文件得1分； | 根据数据 |  |  |
| M6 | 1 | 使用 Logistic 回归模型在测试集上进行预测得1分； | 根据数据 |  |  |
| M7 | 2 | 正确计算Logistic 回归模型的准确率得2分； | 根据数据 |  |  |
| M8 | 2 | 正确处理数据不平衡问题得2分； | 根据数据 |  |  |
| M9 | 2 | 模型重新训练得2分 | 根据数据 |  |  |
| M10 | 1 | 正确生成新预测结果得1分； | 根据数据 |  |  |
| M11 | 1 | 正确分析新的测试结果得1分； | 根据数据 |  |  |
| M12 | 3 | 正确填写2.2.1.docx测试报告：其中模型性能评估得1分、错误分析得1分，改进建议得1分，总计得3分； | 根据数据 |  |  |
| 合计配分 | 20 | 合计得分 | |  |  |

**参考答案**：

（1）2.2.1.ipynb代码

import pandas as pd

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LogisticRegression

import pickle

from sklearn.metrics import classification\_report

from imblearn.over\_sampling import SMOTE

# 加载数据

data = pd.read\_csv('finance数据集.csv')

# 显示前五行的数据

print(data.head())

# 选择自变量和因变量

X = data.drop(['SeriousDlqin2yrs', 'Unnamed: 0'], axis=1)

y = data['SeriousDlqin2yrs']

# 分割训练集和测试集（测试集20%）

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# 训练Logistic回归模型（最大迭代次数为1000次）

model = LogisticRegression(max\_iter=1000)

#训练 Logistic 回归模型

model.fit(X\_train, y\_train)

# 保存模型

with open('2.2.1\_model.pkl', 'wb') as file:

pickle.dump(model, file)

# 预测并保存结果

y\_pred = model.predict(X\_test)

pd.DataFrame(y\_pred, columns=['预测结果']).to\_csv('2.2.1\_results.txt', index=False)

# 生成测试报告

report = classification\_report(y\_test, y\_pred, zero\_division=1)

with open('2.2.1\_report.txt', 'w') as file:

file.write(report)

# 分析测试结果

accuracy = (y\_test == y\_pred).mean()

print(f"模型准确率: {accuracy:.2f}")

# 处理数据不平衡

smote = SMOTE(random\_state=42)

X\_resampled, y\_resampled = smote.fit\_resample(X\_train, y\_train)

# 重新训练模型

model.fit(X\_resampled, y\_resampled)

# 重新预测

y\_pred\_resampled = model.predict(X\_test)

# 保存新结果

pd.DataFrame(y\_pred\_resampled, columns=['预测结果']).to\_csv('2.2.1\_results\_xg.txt', index=False)

# 生成新的测试报告

report\_resampled = classification\_report(y\_test, y\_pred\_resampled, zero\_division=1)

with open('2.2.1\_report\_xg.txt', 'w') as file:

file.write(report\_resampled)

# 分析新的测试结果

accuracy\_resampled = (y\_test == y\_pred\_resampled).mean()

print(f"重新采样后的模型准确率: {accuracy\_resampled:.2f}")

（2）

Logistic模型（说明：只要小数点后第一位正确即可得满分）

模型性能

precision recall f1-score support

0（没有严重逾期） 0.95 0.99 0.97 26779

1（有严重逾期） 0.57 0.14 0.22 1737

错误分析

0（没有严重逾期）：

准确率很高，召回率也很高，表明模型在这一类别上的性能非常好。

可能的错误主要来自于少数漏报情况，即极少数实际没有严重逾期的样本被错误预测为有严重逾期。

1（有严重逾期）：

准确率较低，召回率也很低，F1-Score仅为0.22，表明模型在这一类别上的性能较差。

主要问题在于大量的漏报（真正有严重逾期的样本被预测为没有）和一定的误报（将没有严重逾期的样本预测为有）。

改进建议

1. 数据处理策略调整

重采样技术：由于数据集存在明显的不平衡，可以考虑使用过采样（如SMOTE）或欠采样技术来平衡两个类别的数量。

2. 特征工程优化

特征选择：仔细审查现有特征，去除冗余或不相关的特征，可能有助于提升模型性能。

特征构造：尝试创建新的、更具区分力的特征，如基于现有特征的交互项或衍生指标。