

第 7 部分

操作技能考核模拟试卷

注 意 事 项

1. 考生根据操作技能考核通知单中所列的试题做好考核准备；
2. 请考生仔细阅读试题单中具体考核内容和要求，并按要求完成操作或进行笔答或口答，若有笔答请考生在答题卷上完成。
3. 操作技能考核时要遵守考场纪律，服从考场管理人员指挥，以保证考核安全顺利进行。

注：操作技能考核试题评分表及答案是考评员对考生考核过程及考核结果的评分记录表，也是评分依据。

人工智能训练师（三级）操作技能考核通知单

姓名：

准考证号：

考核日期：

试题 1

试题代码：1.1.1

试题名称：智能医疗系统中的业务数据处理流程设计

考核时间：30min

配分：25 分

试题 2

试题代码：2.1.1

试题名称：智慧交通中燃油效率模型的数据清洗和标注流程设计

考核时间：20min

配分：15 分

试题 3

试题代码：2.2.1

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

配分：20 分

试题 4

试题代码：3.1.1

试题名称：智能音箱产品的数据分析与优化

考核时间：20min

配分：15 分

试题 5

试题代码：3.2.1

试题名称：图像识别评估系统交互流程设计

考核时间：20min

配分：20 分

试题 6

试题代码：4.2.1

试题名称：智能零售分析系统数据采集和处理指导

考核时间：10min

配分：5 分

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：1.1.1

试题名称：智能医疗系统中的业务数据处理流程设计

考核时间：30min

1. 场地设备要求

人工智能训练师主机：CPU (intel i5 及以上)、内存 (不少于 16GB)、操作系统 (windows10)、支持深度学习训练；

2. 工作任务

某医疗机构计划引入智能医疗系统，以提升诊断效率和准确性。通过分析患者的历史数据，使用机器学习算法预测患者的健康风险，从而辅助医生进行诊断和治疗。为此，该机构需要设计一套全面的业务数据处理流程，确保数据处理的高效性和准确性，为人工智能模型提供可靠的输入数据。

我们提供一个患者数据集 (patient_data.csv)，包含以下字段：

- PatientID: 患者 ID
- Age: 年龄
- BMI: 体重指数
- BloodPressure: 血压
- Cholesterol: 胆固醇水平
- DaysInHospital: 住院天数

你作为人工智能训练师，根据提供的数据集和 Python 代码框架 (1.1.1.ipynb)，完成以下数据的统计和分析，为智能医疗系统提供可靠的数据支持。

(1) 通过补全并运行 Python 代码 (1.1.1.ipynb) 分别统计住院天数超过 7 天的患者数量以及其占比。这类患者被定义为高风险患者，反之为低风险患者。将上述统计结果截图以 JPG 的格式保存，命名为“1.1.1-1”。

(2) 通过补全并运行 Python 代码 (1.1.1.ipynb) 统计不同 BMI 区间中高风险患者的

比例和统计不同 BMI 区间中的患者数。BMI 区间分类设置为：偏瘦（低于 18.5），正常（18.5～23.9），超重（24.0～27.9），肥胖（28.0 及以上），将上述统计结果截图以 JPG 的格式保存，命名为“1.1.1-2”。

（3）通过补全并运行 Python 代码（1.1.1. ipynb）统计不同年龄区间中高风险患者的比例和统计不同年龄区间中的患者数。年龄区间分类设置为：年龄区间分类设置为：≤25 岁，26-35 岁，36-45 岁，46-55 岁，56-65 岁，>65 岁，将上述统计结果截图以 JPG 的格式保存，命名为“1.1.1-3”。

所有结果文件储存在桌面新建的考生文件夹中，文件夹命名为“准考证号+身份证号后六位”。

3. 技能要求

（1）能结合人工智能技术要求和业务特征，设计整套业务数据处理流程；

4. 质量指标

（1）设计出的业务数据底层逻辑清晰，有效合理。

人工智能训练师（三级）操作技能考核

试题评分表及答案

准考证号：

试题代码：1.1.1

试题名称：智能医疗系统中的业务数据处理流程设计

考核时间：30min

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 | 1 | 1.1.1-1. JPG 中数据正确得 1 分； | 根据数据 | | |
| M2 | 1 | 1.1.1-2. JPG 中数据正确得 1 分； | 根据数据 | | |
| M3 | 1 | 1.1.1-3. JPG 中数据正确得 1 分； | 根据数据 | | |
| M4 | 1 | 读取数据集代码正确得 1 分； | 根据数据 | | |
| M5 | 3 | 创建新列'RiskLevel'，根据住院天数判断风险等级代码正确得 3 分； | 根据数据 | | |
| M6 | 2 | 统计不同风险等级的患者数量代码正确得 2 分； | 根据数据 | | |
| M7 | 1 | 计算高风险患者占比代码正确得 1 分； | 根据数据 | | |
| M8 | 1 | 计算低风险患者占比代码正确得 1 分； | 根据数据 | | |
| M9 | 4 | 根据年龄值划分指定区间代码正确得 4 分； | 根据数据 | | |
| M10 | 2 | 计算每个 BMI 区间中高风险患者的比例代码正确得 2 分； | 根据数据 | | |
| M11 | 1 | 统计每个 BMI 区间的患者数量代码正确得 1 分； | 根据数据 | | |
| M12 | 4 | 根据年龄值分配年龄区间标签代码正确得 4 分； | 根据数据 | | |
| M13 | 2 | 算每个年龄区间中高风险患者的比例代码正确得 2 分； | 根据数据 | | |
| M14 | 1 | 统计每个年龄区间的患者数量代码正确得 1 分； | 根据数据 | | |
| 合计配分 | 25 | 合计得分 | | | |

参考答案

Python 参考代码 (1.1.1. ipynb):

```
import pandas as pd
import numpy as np
# 读取数据集 1 分
data = pd.read_csv('patient_data.csv')
# 1. 统计住院天数超过 7 天的患者数量及其占比
# 创建新列'RiskLevel', 根据住院天数判断风险等级 3 分
data['RiskLevel'] = np.where(data['DaysInHospital'] > 7, '高风险患者', '低风险患者')
# 统计不同风险等级的患者数量 2 分
risk_counts = data['RiskLevel'].value_counts()
# 计算高风险患者占比 1 分
high_risk_ratio = risk_counts['高风险患者'] / len(data)
# 计算低风险患者占比 1 分
low_risk_ratio = risk_counts['低风险患者'] / len(data)
# 输出结果
print("高风险患者数量:", risk_counts['高风险患者'])
print("低风险患者数量:", risk_counts['低风险患者'])
print("高风险患者占比:", high_risk_ratio)
print("低风险患者占比:", low_risk_ratio)
# 2. 统计不同 BMI 区间中高风险患者的比例和统计不同 BMI 区间中的患者数
# 定义 BMI 区间和标签
bmi_bins = [0, 18.5, 24, 28, np.inf]
bmi_labels = ['偏瘦', '正常', '超重', '肥胖']
# 根据 BMI 值划分指定区间 4 分
data['BMIRange'] = pd.cut(data['BMI'], bins=bmi_bins, labels=bmi_labels, right=False) # 使用
左闭右开区间
# 计算每个 BMI 区间中高风险患者的比例 2 分
bmi_risk_rate = data.groupby('BMIRange')['RiskLevel'].apply(lambda x: (x == '高风险患者')
).mean())
# 统计每个 BMI 区间的患者数量 1 分
bmi_patient_count = data['BMIRange'].value_counts()
# 输出结果
print("BMI 区间中高风险患者的比例和患者数:")
print(bmi_risk_rate)
```

```
print(bmi_patient_count)
# 3. 统计不同年龄区间中高风险患者的比例和统计不同年龄区间中的患者数
# 定义年龄区间和标签
age_bins = [0, 26, 36, 46, 56, 66, np.inf]
age_labels = ['≤25 岁', '26-35 岁', '36-45 岁', '46-55 岁', '56-65 岁', '>65 岁']
# 根据年龄值划分指定区间 4 分
data['AgeRange'] = pd.cut(data['Age'], bins=age_bins, labels=age_labels, right=False) # 使用左
闭右开区间
# 计算每个年龄区间中高风险患者的比例 2 分
age_risk_rate = data.groupby('AgeRange')['RiskLevel'].apply(lambda x: (x == '高风险患者
').mean())
# 统计每个年龄区间的患者数量 1 分
age_patient_count = data['AgeRange'].value_counts()
# 输出结果
print("年龄区间中高风险患者的比例和患者数:")
print(age_risk_rate)
print(age_patient_count)
```

1.1.1-1.jpg

```
高风险患者数量: 413
低风险患者数量: 587
高风险患者占比: 0.413
低风险患者占比: 0.587
```

1.1.1-2.jpg

BMI区间中高风险患者的比例和患者数:

BMIRange

偏瘦 0.444444

正常 0.406699

超重 0.388235

肥胖 0.415094

Name: RiskLevel, dtype: float64

肥胖 477

正常 209

超重 170

偏瘦 144

Name: BMIRange, dtype: int64

1.1.1-3.jpg

年龄区间中高风险患者的比例和患者数:

AgeRange

≤25岁 0.456693

26-35岁 0.398496

36-45岁 0.386364

46-55岁 0.444444

56-65岁 0.401575

>65岁 0.401254

Name: RiskLevel, dtype: float64

>65岁 319

46-55岁 162

26-35岁 133

36-45岁 132

≤25岁 127

56-65岁 127

Name: AgeRange, dtype: int64

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：2.1.1

试题名称：智慧交通中燃油效率模型的数据清洗和标注流程设计

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)汽车燃油效率数据集（auto-mpg.csv）；

2. 工作任务

在现代交通中，燃油效率（MPG）是衡量汽车性能和交通系统优化的重要指标之一。高效的燃油利用不仅能够降低车辆运营成本，还能减少碳排放，促进环保。开发一个用于预测汽车燃油效率的模型可以帮助智慧交通系统优化路线规划和车辆调度，从而提升整体交通效率和减少能源消耗。此外，这样的模型还可以帮助消费者做出更明智的购车决策，并帮助厂商优化汽车设计。

现要求根据提供的汽车燃油效率数据集，补全 2.1.1. ipynb 代码。选择合适的特征，开发一个燃油效率预测模型。在开发预测模型之前，首先要对数据进行数据清洗和标注，请完成下面的数据预处理任务，并设计一套标注流程规范：

- (1)正确加载数据集，并显示前五行的数据及数据类型。
- (2)检查数据集中的缺失值并删除缺失值所在的行。
- (3)将“horsepower”列转换为数值类型，并处理转换中的异常值。
- (4)对数值型数据进行标准化处理，确保数据在同一量纲下进行分析。
- (5)根据业务需求和数据特性，选择对燃油效率预测最有用的特征：选择以下特征：
'cylinders'、'displacement'、'horsepower'、'weight'、'acceleration'、'model year'、
'origin'
- (6)将“mpg”设为目标变量并标注；

-
- (7) 对数据进行标注和划分;
 - (8) 保存处理后的数据, 并命名为: 2.1.1_cleaned_data.csv, 保存到考生文件夹;
 - (9) 制定数据清洗和标注规范, 将答案写到答题卷文件中, 答题卷文件命名为“2.1.1.docx”, 保存到考生文件夹;
 - (10) 将以上代码以及运行结果, 以 html 格式保存并命名为 2.1.1.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”。

3. 技能要求

- (1) 能结合人工智能技术要求和业务特征, 设计数据清洗和标注流程;
- (2) 能结合人工智能技术要求和业务特征, 制定数据清洗和标注规范。

4. 质量指标

- (1) 深入理解业务, 训练符合业务需求的模型

人工智能训练师（三级）操作技能考核

试题评分表及答案

准考证号：

试题代码：2.1.1

试题名称：智慧交通中燃油效率模型的数据清洗和标注流程设计

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 | 1 | 数据集正确加载得 1 分； | 根据数据 | | |
| M2 | 2 | 检查缺失值得 1 分，删除缺失值得 1 分，总计得 2 分； | 根据数据 | | |
| M3 | 1 | 处理转换中的异常值得 1 分； | 根据数据 | | |
| M4 | 1 | 对数据进行标准化处理得 1 分； | 根据数据 | | |
| M5 | 1 | 选择特征得 1 分； | 根据数据 | | |
| M6 | 2 | 创建自变量得 1 分，目标变量得 1 分，总计得 2 分； | 根据数据 | | |
| M7 | 1 | 数据集划分得 1 分； | 根据数据 | | |
| M8 | 1 | 保存处理后的数据得 1 分； | 根据数据 | | |
| M9 | 2 | 回答数据清洗规范：每回答正确 1 个规范点，得 1 分，最高得 2 分 | 根据数据 | | |
| M10 | 3 | 回答数据标注规范：每回答正确 1 个规范点，得 1 分，最高得 3 分 | 根据数据 | | |
| 合计配分 | 15 | 合计得分 | | | |

参考答案

(1) 2.1.1. ipynb:

```
import pandas as pd
```

```
# 加载数据集并显示数据集的前五行 1 分
```

```
data = pd.read_csv('auto-mpg.csv')
```

```
print("数据集的前五行:")
```

```
print(data.head())

# 显示每一列的数据类型
print(data.dtypes)

# 检查缺失值并删除缺失值所在的行 2 分
print("\n 检查缺失值:")
print(data.isnull().sum()) #或者 print(data.isna().sum())
data = data.dropna()

# 将 'horsepower' 列转换为数值类型, 并处理转换中的异常值 1 分
data['horsepower'] = pd.to_numeric(data['horsepower'], errors='coerce')
data = data.dropna(subset=['horsepower'])

# 显示每一列的数据类型
print(data.horsepower.dtypes)

# 检查清洗后的缺失值
print("\n 检查清洗后的缺失值:")
print(data.isnull().sum())

from sklearn.preprocessing import StandardScaler
# 对数值型数据进行标准化处理 1 分
numerical_features = ['displacement', 'horsepower', 'weight', 'acceleration']
scaler = StandardScaler()
data[numerical_features] = scaler.fit_transform(data[numerical_features])

from sklearn.model_selection import train_test_split
# 选择特征和目标变量 2 分
selected_features = ['cylinders', 'displacement', 'horsepower', 'weight',
'acceleration', 'model year', 'origin']
X = data[selected_features]
y = data['mpg']
```

划分数据集为训练集和测试集 1 分

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,  
random_state=42)
```

将特征和目标变量合并到一个数据框中

```
cleaned_data = X.copy()  
cleaned_data['mpg'] = y
```

保存清洗和处理后的数据

```
cleaned_data.to_csv('2.1.1_cleaned_data.csv', index=False)
```

打印消息指示文件已保存

```
print("\n 清洗后的数据已保存到 2.1.1_cleaned_data.csv")
```

(2) 数据清洗和标注规范

数据清洗规范（答对 2 点即可）

1. 数据加载：使用 pandas 库加载数据集，检查数据的基本结构和类型。
2. 检查缺失值：统计每列的缺失值数量，并删除包含缺失值的行以确保数据完整性。
3. 转换与处理异常值：将数值列（如“horsepower”）转换为数值类型，并处理无法转换的值（例如，将其变为缺失值）。
4. 数据标准化：对数值型数据进行标准化，以消除量纲影响，使用标准化方法。
5. 保存清洗后的数据：将经过清洗和处理后的数据保存为新的 CSV 文件，以便后续使用。

数据标注规范（答对 3 点即可）

1. 数据来源：标注数据的来源，包括数据集的名称、获取日期和数据提供者。
2. 数据描述：提供详细的数据描述，包括每列数据的含义、单位和可能的取值范围。
3. 特征选择：确定对目标变量预测最有用的特征。
4. 目标变量设定：将数据集中用于预测的目标变量定义为“mpg”（燃油效率）。
5. 数据划分：将数据分为训练集和测试集，通常采用 80/20 的比例，以便于模型的训练和评估。
6. 保存处理后的数据：保存处理后的数据，并记录保存文件的路径和文件名。
7. 数据清洗和标注规范文档

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：2.2.1

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 - 一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

试题评分表及答案

准考证号：

试题代码：2.2.1

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
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
model = LogisticRegression(max_iter=1000)
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。

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：3.1.1

试题名称：智能音箱产品的数据分析与优化

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）智能音箱数据集。

2. 工作任务

智能音箱作为智能家居生态的重要组成部分，近年来经历了爆炸式的增长。随着人工智能技术的成熟，尤其是自然语言处理（NLP）和语音识别技术的进步，智能音箱已经成为许多家庭中不可或缺的智能设备。它们不仅能够播放音乐、提供天气预报和新闻更新，还能控制家中的智能设备，设定提醒，甚至帮助用户购物。然而，随着市场竞争的加剧，智能音箱制造商面临着如何持续优化产品性能，提升用户体验的挑战。数据分析在智能音箱的持续优化中扮演着核心角色。通过对用户行为的深入挖掘，企业可以了解用户偏好、使用模式和潜在的痛点。例如，分析用户何时最常使用智能音箱、他们最喜欢的功能是什么、以及哪些功能的响应时间过长等问题，可以帮助企业做出有针对性的改进决策。

（1）你作为人工智能训练师，根据给定的数据集（智能音箱数据集.xlsx），从以下三方面：
用户使用习惯：分析哪些功能最常被使用；

功能使用频率：识别最受欢迎的功能和较少使用的功能；

响应时间：考察不同功能的平均响应时间，找出可能的瓶颈。

给出一份在用户使用习惯、功能使用频率和响应时间方面的分析报告，将其保存为 docx 文件，命名为 3.1.1-1.docx。

（2）为了进一步提升用户体验，给出智能音箱产品的 3 个优化方向 and 对应解决方案，将其保存为 docx 文件，命名为 3.1.1-2.docx。

所有结果文件储存在桌面新建的考生文件夹中，文件夹命名为“准考证号+身份证号后六位”。

3. 技能要求

- (1) 能对单一智能产品使用的数据进行全面分析，输出分析报告；
- (2) 能对单一智能产品提出优化需求；
- (3) 能为单一智能产品的应用设计智能解决方案。

4. 质量指标

- (1) 分析报告全面可靠；
- (2) 优化方向合理，具有良好应用价值；
- (3) 解决方案切实可行。

人工智能训练师（三级）操作技能考核

试题评分表及答案

准考证号：

试题代码：3.1.1

试题名称：智能音箱产品的数据分析与优化

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 | 3 | 回答用户使用习惯中最常被使用的三个功能：每个功能得1分，本项最高得3分； | 根据数据 | | |
| M2 | 3 | 回答最受欢迎的功能和较少使用的功能：最受欢迎的功能得1分，较少使用的功能每个得1分，本项最高得3分； | 根据数据 | | |
| M3 | 3 | 回答不同功能的平均响应时间：响应时间较长的功能得1分；响应时间适中的功能得1分；响应时间较短的功能的1分；本项最高得3分； | 根据数据 | | |
| M4 | 6 | 回答优化方向和该方向对应解决方案：每1个正确的优化方向得1分，对应解决方案得1分，本项最多得6分； | 根据数据 | | |
| 合计配分 | 15 | 合计得分 | | | |

参考答案

（1）分析报告

一、用户使用习惯

最常被使用的功能：调整音量、查询新闻、查天气

二、功能使用频率

最受欢迎的功能：调整音量

较少使用的功能：播放音乐、控制家居

三、响应时间分析

响应时间较长的功能：控制家居

响应时间适中的功能：提醒事项/调整音量/查询知识/设置闹钟/播放音乐/查天气

响应时间较短的功能：查询新闻

（2）优化方向及解决方案（仅需回答 3 个优化方向 and 对应解决方案）

1. 优化方向：网络连接与设备交互优化

解决方案：强化网络连接稳定性，采用更高效的通信协议，减少控制家居功能的响应时间

2. 优化方向：提升信息检索效率

解决方案：优化查询知识的检索算法，加快数据处理速度，提升用户体验

3. 优化方向：增强本地处理能力

解决方案：增加本地缓存和预加载机制，尤其在播放音乐和查天气上，以减少对外部资源的依赖

4. 优化方向：改善闹钟与提醒功能

解决方案：设计更直观的闹钟和提醒事项设置流程，增加个性化选项，鼓励用户更频繁地使用这些功能

5. 优化方向：用户界面与语音识别优化

解决方案：提升语音识别准确率，简化用户界面设计，确保用户能够快速、准确地调用所需功能

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：3.2.1

试题名称：图像识别评估系统交互流程设计

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) Pytorch 框架。

2. 工作任务

图像识别评估系统是在深度学习技术日益成熟的背景下发展起来的，旨在解决传统图像识别方法在面对复杂场景和大规模数据集时的局限性。随着互联网和物联网技术的飞速发展，图像数据量呈指数级增长，对图像内容的自动理解和智能分析提出了更高的要求。ResNet 作为一种深度卷积神经网络架构，凭借其深度残差连接机制，能够有效缓解梯度消失问题，实现更深层次的网络结构，从而捕获更加丰富和抽象的图像特征，极大地提高了图像识别的准确性和效率，推动了人工智能技术在现实世界中的广泛应用和商业化进程。

AI 模型说明：“resnet.onnx”模型是使用 Pytorch 框架和基于深度卷积神经网络网络训练得到的，专门用于进行图像识别。对应的标签文件为“labels.txt”。该模型的使用交互流程为：

- 1) 加载“resnet.onnx”模型和“labels.txt”类别标签；
- 2) 加载本地测试图片“img_test.jpg”，并进行预处理图像以符合模型输入要求；
- 3) 使用“resnet.onnx”模型对加载的图片进行识别；
- 4) 输出加载图片的识别结果（输出概率值最大的 5 组类别和对应概率值）

你作为一名人工智能训练师，请完成以下工作任务：

- (1) 补全该模型的使用交互流程对应的 Python 代码（3.2.1.ipynb），实现本地测试图片“img_test.jpg”的识别，将其识别结果截图保存为 jpg 格式文件，命名为 3.2.1-1.jpg。

(2) 在上面的使用交互流程基础上, 给出在图像识别评估系统中使用 “resnet.onnx” 模型的一种人机交互的最优方式, 将其保存为 docx 文件, 命名为 3.2.1.docx。

所有结果文件储存在桌面新建的考生文件夹中, 文件夹命名为 “准考证号+身份证号后六位”。

3. 技能要求

- (1) 能确保模型在单一场景下稳定运行;
- (2) 能通过分析, 找到单一场景下人工和智能交互的最优方式。

4. 质量指标

- (1) 模型运行稳定, 使用正常;
- (2) 单一场景下人工和智能交互的最优方式切实可行。

人工智能训练师（三级）操作技能考核

试题评分表及答案

准考证号：

试题代码：3.2.1

试题名称：图像识别评估系统交互流程设计

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 | 模型加载代码正确得 2 分； | 根据数据 | | |
| M2 | 2 | 从指定路径加载一张图片代码正确得 2 分； | 根据数据 | | |
| M3 | 2 | 预处理图片代码正确得 2 分； | 根据数据 | | |
| M4 | 2 | 使用模型对图片进行识别代码得 2 分； | 根据数据 | | |
| M5 | 2 | 获取识别分类后的概率代码正确得 2 分； | 根据数据 | | |
| M6 | 3 | 获取最高的 5 个概率和对应的类别索引代码正确得 3 分； | 根据数据 | | |
| M7 | 1 | 本地保存识别后的截图 3.2.1-1.jpg：得 1 分； | 根据数据 | | |
| M8 | 5 | 截图 3.2.1-1.jpg 中识别的前五个类别和概率正确：每个类别和概率正确得 1 分，总计得 5 分； | 根据数据 | | |
| M9 | 1 | 人机交互最优流程（不少于 3 条）正确得 1 分； | 根据数据 | | |
| 合计配分 | 20 | 合计得分 | | | |

参考答案

源代码（3.2.1.ipynb）：

```
import onnxruntime as ort
import numpy as np
import scipy.special
from PIL import Image
```

预处理图像

```
def preprocess_image(image, resize_size=256, crop_size=224, mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225]):  
    image = image.resize((resize_size, resize_size), Image.BILINEAR)  
    w, h = image.size  
    left = (w - crop_size) / 2  
    top = (h - crop_size) / 2  
    image = image.crop((left, top, left + crop_size, top + crop_size))  
    image = np.array(image).astype(np.float32)  
    image = image / 255.0  
    image = (image - mean) / std  
    image = np.transpose(image, (2, 0, 1))  
    image = image.reshape((1,) + image.shape)  
    return image
```

模型加载 2 分

```
session = ort.InferenceSession('resnet.onnx')
```

加载类别标签

```
labels_path = 'labels.txt'  
with open(labels_path) as f:  
    labels = [line.strip() for line in f.readlines()]
```

获取模型输入和输出的名称

```
input_name = session.get_inputs()[0].name  
output_name = session.get_outputs()[0].name
```

加载图片 2 分

```
image = Image.open('img_test.jpg').convert('RGB')
```

```
# 预处理图片 2 分
processed_image = preprocess_image(image)

# 确保输入数据是 float32 类型
processed_image = processed_image.astype(np.float32)

# 进行图片识别 2 分
output = session.run([output_name], {input_name: processed_image})[0]

# 应用 softmax 函数获取概率 2 分
probabilities = scipy.special.softmax(output, axis=-1)

# 获取最高的 5 个概率和对应的类别索引 2 分
top5_idx = np.argsort(probabilities[0])[-5:][::-1]
top5_prob = probabilities[0][top5_idx]

# 打印结果
print("Top 5 predicted classes:")
for i in range(5):
    print(f"{i+1}: {labels[top5_idx[i]]} - Probability: {top5_prob[i]}")
```

3.2.1-1.jpg

```
Top 5 predicted classes:
1: tusker - Probability: 0.931233286857605
2: Indian elephant - Probability: 0.06785939633846283
3: African elephant - Probability: 0.0009034110116772354
4: gorilla - Probability: 4.78091294553451e-07
5: water buffalo - Probability: 3.9817274455344887e-07
```

3.2.1. docx

人机交互最优方式为：

1. 用户界面设计：提供一个简洁直观的用户界面，方便用户上传图片和查看识别结果。
2. 模型加载：系统启动时预加载模型和标签，减少用户等待时间。
3. 图像上传与预处理：允许用户上传图片，并自动进行必要的预处理。
4. 图像识别：后台使用“resnet.onnx”模型处理预处理后的图片。
5. 结果展示：清晰展示识别结果，包括最可能的类别和概率。
6. 交互反馈：提供用户反馈选项，以改进模型性能。
7. 帮助与支持：内置帮助文档和客服支持，提升用户体验。
8. 性能优化：优化系统响应速度和模型处理能力。
9. 安全性考虑：确保用户数据安全和隐私保护。
10. 多语言支持：界面支持多语言，满足全球用户需求。
11. 可访问性：确保 UI 无障碍，方便所有用户操作。

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：4.2.1

试题名称：智能零售分析系统数据采集和处理指导

考核时间：10min

1. 场地设备要求

人工智能训练师主机：CPU (intel i5 及以上)、内存 (不少于 16GB)、操作系统 (windows10)、办公软件；

2. 工作任务

在快速发展的零售行业中，商家面临着巨大的竞争压力，需要精准地了解消费者行为、优化库存管理、预测销售趋势以及提升顾客体验。为了应对这些挑战，某大型连锁超市决定部署一套智能零售分析系统，该系统将利用人工智能和大数据分析来提升其业务效率和客户满意度。

智能零售分析系统的目标包括：

- 提高销售转化率和客户忠诚度。
- 减少库存成本，避免滞销和断货。
- 降低运营成本，提高供应链效率。
- 加强市场竞争力，快速响应市场变化。

智能零售分析系统的业务需求包括：

- 顾客行为分析：收集顾客购物习惯、偏好和购买频率，以便于定制个性化营销策略。
- 库存优化：实时监测货架上的商品存量，预测补货需求，减少过度库存和缺货情况。
- 销售预测：基于历史销售数据和市场趋势，预测未来销售，辅助决策。
- 动态定价：根据供需关系、竞争对手价格变动等因素，动态调整商品价格。

-
- **客户满意度提升：**通过分析顾客反馈和评论，识别服务短板，改进服务质量。

你作为一名人工智能训练师，根据上述的智能零售分析系统的系统目标和业务需求，补全智能零售分析系统的数据采集和处理指导方案（见素材文件夹中的 4.2.1.docx）。

所有结果文件储存在桌面新建的考生文件夹中，文件夹命名为“准考证号+身份证号后六位”。

3. 技能要求

- （1）能指导五级/初级工、四级/中级工解决数据采集、处理问题
- （2）能指导五级/初级工、四级/中级工优化数据采集、处理问题

4. 质量指标

- （1）数据采集和处理的指导方案内容合理、可行；
- （2）数据采集全面；
- （3）数据处理准确与完整。

人工智能训练师（三级）操作技能考核

试题评分表及答案

准考证号：

试题代码：4.2.1

试题名称：智能零售分析系统数据采集和处理指导

考核时间：10min

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 | 5 | 正确补充方案内容的每1个内容点得1分，总计得5分； | 根据数据 | | |
| | | 第1个内容点正确，得1分 | | | |
| | | 第2个内容点正确，得1分 | | | |
| | | 第3个内容点正确，得1分 | | | |
| | | 第4个内容点正确，得1分 | | | |
| | | 第5个内容点正确，得1分 | | | |
| 合计配分 | 5 | 合计得分 | | | |

参考答案

智能零售分析系统数据采集和处理指导方案

1. 数据源确定

- 销售点（POS）数据：从收银系统获取交易记录，包括商品种类、数量、价格和购买时间。
- 顾客信息：会员卡使用数据，包括会员基本信息、消费记录和积分兑换。
- 库存管理系统：实时库存量、入库和出库记录。
- 顾客反馈：在线评价、投诉和建议。
- 外部数据：天气预报、节假日信息、竞争对手价格数据。

2. 数据采集方法

- API 接口：与内部系统（如 POS、CRM）和外部数据提供商建立 API 连接，自动化数据抓取。

-
- 传感器和物联网设备：在货架上安装 RFID 标签和重量传感器，监测商品存量。
 - 社交媒体监听：通过社交媒体 API 监听品牌相关的公众讨论和评价。
 - 顾客调查：定期发送电子问卷，收集顾客反馈。

3. 数据预处理

- 清洗：去除重复、错误或无关的数据项。
- 标准化：统一数据格式，例如日期和时间的表示方式。
- 整合：将来自不同来源的数据合并到单一数据库中，创建关联字段。

4. 数据安全性与合规

- 加密传输：确保数据在传输过程中的安全。
- 访问控制：限制对敏感数据的访问权限，只允许授权人员查看。
- 匿名化处理：对个人信息进行去标识化，遵守 GDPR 等数据保护法规。

5. 数据存储与管理

- 云存储：选择可靠的云服务商，如 AWS 或 Azure，存储海量数据。
- 备份与恢复：定期备份数据，并测试恢复流程，以防数据丢失。

6. 数据分析与应用

- 建模：使用机器学习算法预测销售趋势、顾客偏好和库存需求。
- 可视化：开发仪表盘展示关键指标，帮助管理层做出决策。
- 报告：定期生成销售、库存和顾客满意度报告，提供业务洞察。
- 通过上述方案，智能零售分析系统能够有效地采集、处理和分析大量数据，为企业提供决策支持，最终达到提升业务效率和顾客满意度的目标。