**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：智能交通系统的数据采集、处理和审核流程设计

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

某智能交通系统希望通过车辆的行驶数据，利用人工智能技术进行交通流量预测和拥堵预警。你作为人工智能训练师，需要设计一套全面的业务数据采集、处理和审核流程，确保数据在进入交通流量分析系统之前经过严格的采集、清洗、审核和预处理。这里提供一个车辆行驶数据集（vehicle\_traffic\_data.csv），包含以下字段：

* VehicleID: 车辆ID
* DriverName: 驾驶员姓名
* Age: 年龄
* Gender: 性别（Male/Female）
* Speed: 车速（km/h）
* TravelDistance: 行驶距离（km）
* TravelTime: 行驶时间（min）
* TrafficEvent: 交通事件（Normal, Accident, Traffic Jam, Breakdown）

你作为人工智能训练师，根据提供的vehicle\_traffic\_data.csv数据集和Python代码框架（1.1.5.ipynb），完成以下数据的采集、处理和审核任务，确保数据的准确性和可靠性。请按照以下要求完成任务，确保结果准确并保存相应的截图。

（1）数据采集：

通过运行Python代码（1.1.5.ipynb），从本地文件vehicle\_traffic\_data.csv中读取数据，并将数据加载到DataFrame中。显示前5行数据截图以JPG的格式保存，命名为“1.1.5-1”。

（2）数据清洗与预处理：

通过运行Python代码（1.1.5.ipynb）对数据进行清洗和预处理，具体要求如下：

* 处理缺失值：对缺失值进行删除。
* 数据类型转换：确保每个字段的数据类型正确。
* 处理异常值：删除不合理的年龄、车速、行驶距离和行驶时间。

清洗后的数据保存为新文件cleaned\_vehicle\_traffic\_data.csv。

（3）数据合理性审核： 通过运行Python代码审核以下字段的合理性：

* 年龄：应在18到70岁之间。
* 车速：应在0到200 km/h之间。
* 行驶距离：应在1到1000 km之间。
* 行驶时间：应在1到1440分钟（24小时）之间。

对不合理的数据进行标记，并将审核结果截图以JPG的格式保存，命名为“1.1.5-2”。

（4）数据统计：

通过运行Python代码（1.1.5.ipynb），完成以下数据统计任务：

* 统计每种交通事件的发生次数。
* 统计不同性别的平均车速、行驶距离和行驶时间。
* 统计不同年龄段的驾驶员数（18-25岁、26-35岁、36-45岁、46-55岁、56-65岁、65岁以上）。

将统计结果分别截图以JPG的格式保存，分别命名为“1.1.5-3”、“1.1.5-4”、“1.1.5-5”。

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，设计整套业务数据采集流程；

（2）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；

（3）能结合人工智能技术要求和业务特征，设计整套业务数据审核流程；

**4.**质量指标

（1）数据完整性：数据无缺失，每项记录完整。

（2）数据合理性：所有数值在合理范围内，无异常点。

（3）数据一致性：字段类型正确，数据格式统一。

（4）分析准确性：统计结果反映真实数据分布，无偏差。

**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：智能交通系统的数据采集、处理和审核流程设计

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 | 2 | 从本地文件中读取数据的代码正确得2分 | 根据数据 |  |  |
| M2 | 2 | 打印数据的前5条记录得代码正确：得2分 | 根据数据 |  |  |
| M3 | 2 | 处理缺失值的代码正确得2分； | 根据数据 |  |  |
| M4 | 1 | Age数据类型转换的代码得1分； | 根据数据 |  |  |
| M5 | 1 | Speed数据类型转换的代码正确得1分； | 根据数据 |  |  |
| M6 | 1 | TravelDistance数据类型转换的代码正确得1分； | 根据数据 |  |  |
| M7 | 1 | TravelTime数据类型转换的代码正确得1分； | 根据数据 |  |  |
| M8 | 2 | 处理异常值的代码正确得2分； | 根据数据 |  |  |
| M9 | 1 | 保存清洗后的数据代码正确得1分； | 根据数据 |  |  |
| M10 | 1 | cleaned\_vehicle\_traffic\_data.csv文件正确：得1分； | 根据数据 |  |  |
| M11 | 2 | 审核字段合理性代码正确得2分； | 根据数据 |  |  |
| M12 | 2 | 统计每种交通事件的发生次数代码正确得2分； | 根据数据 |  |  |
| M13 | 2 | 统计不同性别的平均车速、行驶距离和行驶时间代码正确得2分； | 根据数据 |  |  |
| M14 | 5 | 统计不同年龄段的驾驶员数代码正确得5分：每空1分，总计得5分； | 根据数据 |  |  |
| 合计配分 | 25 | 合计得分 | |  |  |

**参考答案**：

Python参考代码（1.1.5.ipynb）：

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

# 1. 数据采集

# 从本地文件中读取数据

data = pd.read\_csv('vehicle\_traffic\_data.csv')

print("数据采集完成，已加载到DataFrame中")

# 打印数据的前5条记录

print(data.head())

# 2. 数据清洗与预处理

# 处理缺失值（删除包含缺失值的行）

data = data.dropna()

# 数据类型转换

data['Age'] = data['Age'].astype(int) # Age转整型

data['Speed'] = data['Speed'].astype(float) # Speed转浮点型

data['TravelDistance'] = data['TravelDistance'].astype(float) # TravelDistance转浮点型

data['TravelTime'] = data['TravelTime'].astype(float) # TravelTime转浮点型

# 处理异常值（筛选合理范围）

data = data[(data['Age'].between(18, 70)) &

(data['Speed'].between(0, 200)) &

(data['TravelDistance'].between(1, 1000)) &

(data['TravelTime'].between(1, 1440))]

# 保存清洗后的数据

data.to\_csv('cleaned\_vehicle\_traffic\_data.csv', index=False)

print("数据清洗完成，已保存为 'cleaned\_vehicle\_traffic\_data.csv'")

# 3. 数据合理性审核

# 审核字段合理性

unreasonable\_data = data[~((data['Age'].between(18, 70)) |

(~data['Speed'].between(0, 200)) |

(~data['TravelDistance'].between(1, 1000)) |

(~data['TravelTime'].between(1, 1440)))]

print("不合理的数据:\n", unreasonable\_data)

# 4. 数据统计

# 统计每种交通事件的发生次数

traffic\_event\_counts = data['TrafficEvent'].value\_counts()

print("每种交通事件的发生次数:\n", traffic\_event\_counts)

# 统计不同性别的平均车速、行驶距离和行驶时间

gender\_stats = data.groupby('Gender').agg({

'Speed': 'mean',

'TravelDistance': 'mean',

'TravelTime': 'mean'

})

print("不同性别的平均车速、行驶距离和行驶时间:\n", gender\_stats)

# 统计不同年龄段的驾驶员数

age\_bins = [18, 26, 36, 46, 56, 66, np.inf]

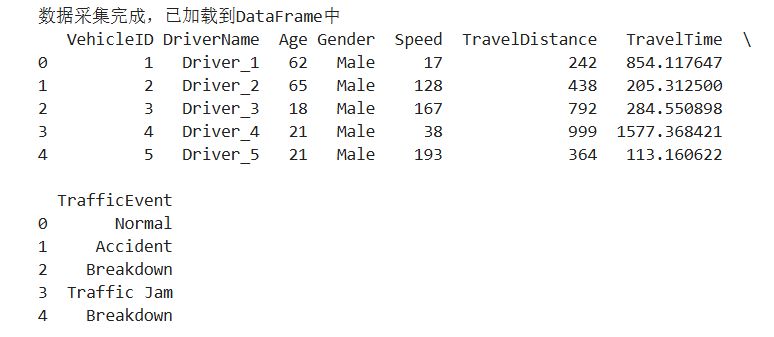
age\_labels = ['18-25', '26-35', '36-45', '46-55', '56-65', '65+']

data['AgeGroup'] = pd.cut(data['Age'], bins=age\_bins, labels=age\_labels, right=False)

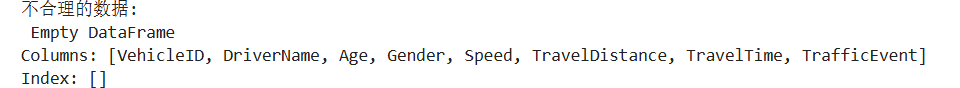
age\_group\_counts = data['AgeGroup'].value\_counts().sort\_index()

print("不同年龄段的驾驶员数:\n", age\_group\_counts)

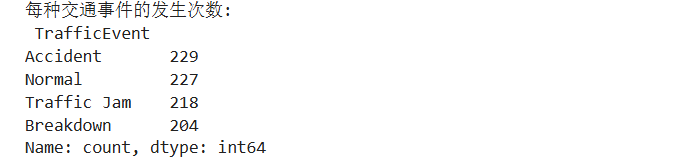
1.1.5-1.jpg



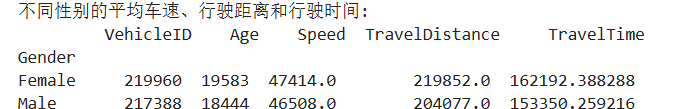
1.1.5-2.jpg



1.1.5-3.jpg



1.1.5-4.jpg



1.1.5-5.jpg

