**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：日常运动量随机森林预测模型开发与测试

考核时间：20min

**1.**场地设备要求

（1）人工智能训练师主机 1 台；

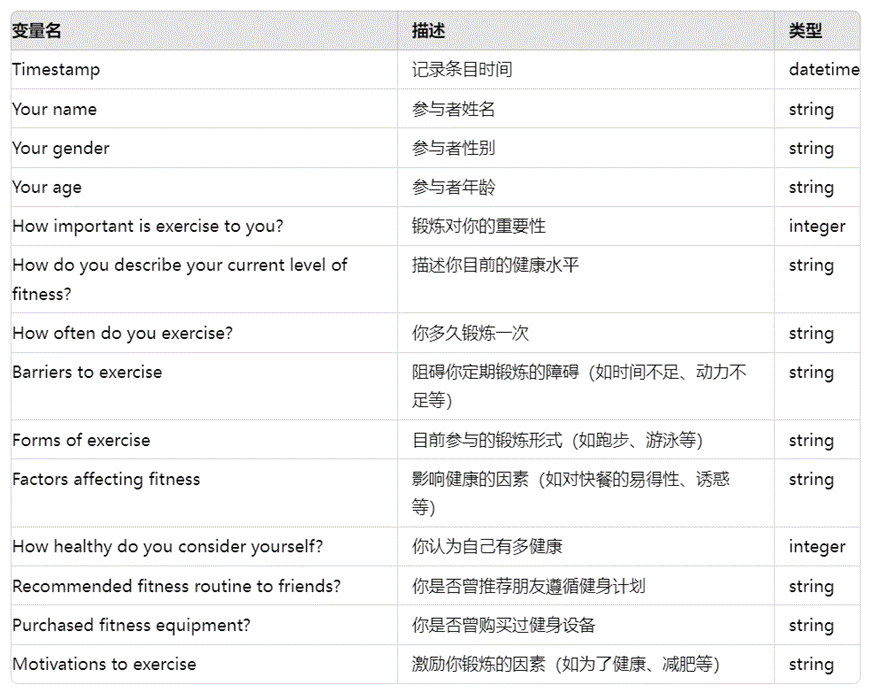
（2）Python 编译环境；

（3）fitness analysis数据集。

**2.**工作任务

随着人们健康意识的增强，越来越多的人开始关注日常运动和健康管理。使用提供的训练数据，补全2.2.3.ipynb代码。选择合适的特征，开发一个预测模型，基于个体性别，个体对运动的看法和个人健康评价来预测个体年龄。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。

详细说明如下：



（1）正确加载数据集，并显示前五行的数据

（2）使用随机森林模型进行模型训练，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2.2.3\_model.pkl保存到考生文件夹，结果文件以2.2.3\_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2.2.3\_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“2.2.3.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2.2.3\_results\_xgb.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2.2.3.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：日常运动量随机森林预测模型开发与测试

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 | 2 | 将分类变量转为数值变量正确得1分，将年龄段转为数值变量正确得1分，总计得2分； | 根据数据 |  |  |
| M3 | 1 | 训练集与测试集的准确划分得1分； | 根据数据 |  |  |
| M4 | 1 | 初始化随机森林模型得1分； | 根据数据 |  |  |
| M5 | 1 | 训练随机森林模型得1分； | 根据数据 |  |  |
| M6 | 1 | 保存随机森林模型文件得1分； | 根据数据 |  |  |
| M7 | 1 | 使用随机森林模型在测试集上进行结果预测得1分； | 根据数据 |  |  |
| M8 | 1 | 计算随机森林模型训练集以及测试集分数得1分； | 根据数据 |  |  |
| M9 | 1 | 计算随机森林模型均方误差得1分； | 根据数据 |  |  |
| M10 | 1 | 计算随机森林模型决定系数得1分； | 根据数据 |  |  |
| M11 | 1 | 初始化XGBoost回归模型得1分； | 根据数据 |  |  |
| M12 | 1 | 训练XGBoost回归模型得1分； | 根据数据 |  |  |
| M13 | 1 | 使用XGBoost回归模型在测试集上进行结果预测得1分； | 根据数据 |  |  |
| M14 | 1 | 计算XGBoost回归模型训练集以及测试集分数得1分； | 根据数据 |  |  |
| M15 | 1 | 计算XGBoost回归模型均方误差和决定系数得1分； | 根据数据 |  |  |
| M16 | 3 | 正确填写2.2.3.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）2.2.3.ipynb代码

import pandas as pd

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

from sklearn.ensemble import RandomForestRegressor

import pickle

from sklearn.metrics import mean\_squared\_error, r2\_score

from xgboost import XGBRegressor # 修正导入方式

# 加载数据集

df = pd.read\_csv('fitness analysis.csv')

# 显示前五行数据

print(df.head())

# 去除所有字符串字段的前后空格

df = df.applymap(lambda x: x.strip() if isinstance(x, str) else x)

# 检查和清理列名

df.columns = df.columns.str.strip()

# 选择相关特征进行建模

X = df[['Your gender', 'How important is exercise to you ?', 'How healthy do you consider yourself?']]

X = pd.get\_dummies(X) # 将分类变量转为数值变量

# 将年龄段转为数值变量

y = df['Your age'].apply(lambda x: int(x.split(' ')[0])) # 假设年龄段为整数

# 将数据集划分为训练集和测试集

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

# 创建并训练随机森林回归模型

rf\_model = RandomForestRegressor(n\_estimators=100, random\_state=42)

rf\_model.fit(X\_train, y\_train)

# 保存训练好的模型

with open('2.2.3\_model.pkl', 'wb') as model\_file:

pickle.dump(rf\_model, model\_file)

# 进行结果预测

y\_pred = rf\_model.predict(X\_test)

results\_df = pd.DataFrame(y\_pred, columns=['预测结果'])

results\_df.to\_csv('2.2.3\_results.txt', index=False)

# 使用测试工具对模型进行测试，并记录测试结果

train\_score = rf\_model.score(X\_train, y\_train) #训练集分数

test\_score = rf\_model.score(X\_test, y\_test) #测试集分数

mse = mean\_squared\_error(y\_test, y\_pred) #均方误差

r2 = r2\_score(y\_test, y\_pred) #决定系数

with open('2.2.3\_report.txt', 'w') as report\_file:

report\_file.write(f'训练集得分: {train\_score:.4f}\n')

report\_file.write(f'测试集得分: {test\_score:.4f}\n')

report\_file.write(f'均方误差(MSE): {mse:.4f}\n')

report\_file.write(f'决定系数(R^2): {r2:.4f}\n')

# 运用工具分析算法中错误案例产生的原因并进行纠正

# 初始化XGBoost回归模型（构建100棵树）

xgb\_model = xgb.XGBRegressor(n\_estimators=100, random\_state=42)

# 训练XGBoost回归模型

xgb\_model.fit(X\_train, y\_train)

# 使用XGBoost回归模型在测试集上进行结果预测

y\_pred\_xgb = xgb\_model.predict(X\_test)

results\_df\_xgb = pd.DataFrame(y\_pred\_xgb, columns=['预测结果'])

results\_df\_xgb.to\_csv('2.2.3\_results\_xgb.txt', index=False)

with open('2.2.3\_report\_xgb.txt', 'w') as xgb\_report\_file:

xgb\_report\_file.write(f'XGBoost训练集得分: {xgb\_model.score(X\_train, y\_train)}\n')

xgb\_report\_file.write(f'XGBoost测试集得分: {xgb\_model.score(X\_test, y\_test)}\n')

xgb\_report\_file.write(f'XGBoost均方误差(MSE): {mean\_squared\_error(y\_test, y\_pred\_xgb)}\n')

xgb\_report\_file.write(f'XGBoost决定系数(R^2): {r2\_score(y\_test, y\_pred\_xgb)}\n')

1. 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

测试报告分析

1. 模型性能评估

随机森林模型：

训练集得分: 0.1239

测试集得分: -0.0922

均方误差(MSE): 109.7669

决定系数(R^2): -0.0922

XGBoost模型：

XGBoost训练集得分: 0.12619702605403083

XGBoost测试集得分: -0.0910554062361888

XGBoost均方误差(MSE): 109.76692738027478

XGBoost决定系数(R^2): -0.0910554062361888

2. 错误分析

特征不足：仅使用性别、运动重要性和健康自评三个特征，未考虑锻炼频率、饮食等因素。

数据噪声：年龄字段通过简单分拆可能导致误差（例如“15 to 18”转为15，但实际平均年龄为16.5）。

过拟合迹象：随机森林训练集得分显著高于测试集，表明模型可能过拟合。

3. 改进建议

增加特征：纳入锻炼频率、饮食平衡等字段。

优化年龄处理：将年龄段转为中值（如“15 to 18”转为16.5）。

调整模型参数：

随机森林：限制树深度（max\_depth=5）。

XGBoost：添加早停法（early\_stopping\_rounds=10）。

交叉验证：使用K折交叉验证提升泛化能力。。