# app.py

import streamlit as st

import pandas as pd

import xgboost as xgb

import shap

import matplotlib.pyplot as plt

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

# 2. 指定自变量和因变量

data = pd.read\_excel(

r"F:\数据\（1）机器学习预测模型数据\A012-CHARLS 2011-2020年已清洗最全"

r"\charls2011~2020清洗好+原版数据\整理完的-charls数据\Lasso回归\data\_selected.xlsx"

)

selected\_features = [

'age', 'gender', 'familysize', 'exercise', 'totmet',

'srh', 'diabe', 'cancre', 'hearte', 'satlife',

'iadl', 'pain'

]

X = data[selected\_features]

y = data['hibpe']

# 3. 划分训练集和测试集

X\_train, X\_test, y\_train, y\_test = train\_test\_split(

X, y, test\_size=0.3, random\_state=123

)

# 4. 创建并训练 XGBoost 分类模型

model = xgb.XGBClassifier(

learning\_rate=0.1, max\_depth=3, n\_estimators=100, subsample=0.8

)

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

@st.cache\_resource

def load\_model\_and\_explainer():

# 这是函数体，缩进了 4 个空格

explainer = shap.TreeExplainer(model)

return model, explainer # ← 也是同样的 4 个空格

# 函数定义完毕，回到顶格

model, explainer = load\_model\_and\_explainer()

# —— 2. Streamlit 页面布局 ——

st.title("健康风险预测 (CHARLS) + SHAP 可解释性")

st.sidebar.header("输入患者特征")

age = st.sidebar.number\_input("Age", min\_value=18, max\_value=120, value=60)

gender = st.sidebar.selectbox("Gender (0=F, 1=M)", [0,1])

familysize = st.sidebar.number\_input("Family size", min\_value=1, max\_value=10, value=3)

exercise = st.sidebar.selectbox("Exercise (0=No, 1=Yes)", [0,1])

totmet = st.sidebar.number\_input("Total metabolism", min\_value=0.0, step=0.1, value=5.0)

srh = st.sidebar.number\_input("Self-rated health (1–5)",min\_value=1, max\_value=5, value=3)

diabe = st.sidebar.selectbox("Diabetes (0=No, 1=Yes)", [0,1])

cancre = st.sidebar.selectbox("Cancer (0=No, 1=Yes)", [0,1])

hearte = st.sidebar.selectbox("Heart disease (0=No, 1=Yes)", [0,1])

satlife = st.sidebar.number\_input("Life satisfaction (1–5)",min\_value=1, max\_value=5, value=3)

iadl = st.sidebar.number\_input("IADL score", min\_value=0.0, max\_value=10.0, step=0.1, value=2.0)

pain = st.sidebar.selectbox("Pain (0=No, 1=Yes)", [0,1])

if st.sidebar.button("Predict"):

# —— 3. 构造输入、预测 ——

X\_new = pd.DataFrame([{

"age": age, "gender": gender, "familysize": familysize,

"exercise": exercise, "totmet": totmet, "srh": srh,

"diabe": diabe, "cancre": cancre, "hearte": hearte,

"satlife": satlife, "iadl": iadl, "pain": pain

}])

proba = model.predict\_proba(X\_new)[0]

pred = model.predict(X\_new)[0]

st.subheader("预测结果")

st.write(f"- \*\*Predicted Class:\*\* {pred}")

st.write(f"- \*\*Prediction Probabilities:\*\* `{proba}`")

# —— 4. SHAP 力图（静态 matplotlib 版） ——

shap\_values = explainer.shap\_values(X\_new)

fig, ax = plt.subplots(figsize=(8, 1.5))

shap.force\_plot(

explainer.expected\_value,

shap\_values[0],

X\_new.iloc[0],

matplotlib=True,

show=False

)

plt.title("SHAP Force Plot for This Sample")

plt.tight\_layout()

st.pyplot(fig)