# -\*- coding: utf-8 -\*-

import streamlit as st

import pandas as pd

import joblib

import os

# Set Streamlit page config

st.set\_page\_config(page\_title="CKD Prediction")

# Title

st.title("Chronic Kidney Disease Prediction App")

st.markdown("Enter the patient's clinical data below to predict the presence of Chronic Kidney Disease (CKD).")

# Check if model file exists

model\_path = "tuned\_balanced\_rf\_ckd.pkl"

if not os.path.exists(model\_path):

st.error(f"Model file '{model\_path}' not found.")

st.stop()

# Load the trained model

model = joblib.load(model\_path)

# Input fields

age = st.number\_input("Age", min\_value=0, max\_value=120)

bmi = st.number\_input("BMI")

serumcreatinine = st.number\_input("Serum Creatinine")

gfr = st.number\_input("GFR")

systolicbp = st.number\_input("Systolic Blood Pressure")

diastolicbp = st.number\_input("Diastolic Blood Pressure")

hba1c = st.number\_input("HbA1c")

bunlevels = st.number\_input("BUN Levels")

cholesterolldl = st.number\_input("LDL Cholesterol")

acr = st.number\_input("Albumin-to-Creatinine Ratio (ACR)")

# Predict button

if st.button("Predict"):

input\_df = pd.DataFrame([[age, bmi, serumcreatinine, gfr, systolicbp,

diastolicbp, hba1c, bunlevels, cholesterolldl, acr]],

columns=['age', 'bmi', 'serumcreatinine', 'gfr', 'systolicbp',

'diastolicbp', 'hba1c', 'bunlevels', 'cholesterolldl', 'acr'])

prediction = model.predict(input\_df)[0]

if prediction == 1:

st.error("⚠️ Chronic Kidney Disease Detected")

else:

st.success("✅ No CKD Detected")