import streamlit as st

import pickle

# Load the trained model

with open('model.pkl', 'rb') as f:

    model = pickle.load(f)

# Define diagnosis mapping dictionary

diagnoses = {

    0: 'Negative',

    1: 'Hypothyroid',

    2: 'Hyperthyroid'

}

# Predicted diagnosis color

diagnosis\_color = '#F63366'

title\_color = '#F63366'  # Title color

title\_css = f"<h1 style='text-align: center; color: {title\_color};'>Thyroid Classification and Diagnosis</h1>"

# Detect button color

detect\_button\_color = '#F63366'

# Function to preprocess inputs before prediction

def preprocess\_inputs(age, sex, on\_thyroxine, query\_on\_thyroxine, on\_antithyroid\_meds, sick, pregnant,

                      thyroid\_surgery, I131\_treatment, query\_hypothyroid, query\_hyperthyroid, lithium,

                      goitre, tumor, hypopituitary, psych, TSH, T3, TT4, T4U, FTI):

    # Replace 'Yes' with 1 and 'No' with 0

    binary\_map = {'Yes': 1, 'No': 0, '': None}

    on\_thyroxine = binary\_map.get(on\_thyroxine)

    query\_on\_thyroxine = binary\_map.get(query\_on\_thyroxine)

    on\_antithyroid\_meds = binary\_map.get(on\_antithyroid\_meds)

    sick = binary\_map.get(sick)

    pregnant = binary\_map.get(pregnant)

    thyroid\_surgery = binary\_map.get(thyroid\_surgery)

    I131\_treatment = binary\_map.get(I131\_treatment)

    query\_hypothyroid = binary\_map.get(query\_hypothyroid)

    query\_hyperthyroid = binary\_map.get(query\_hyperthyroid)

    lithium = binary\_map.get(lithium)

    goitre = binary\_map.get(goitre)

    tumor = binary\_map.get(tumor)

    hypopituitary = binary\_map.get(hypopituitary)

    psych = binary\_map.get(psych)

    # Replace 'M' and 'F' with binary 0 and 1

    sex = 1 if sex == 'F' else 0 if sex == 'M' else None

    return [age, sex, on\_thyroxine, query\_on\_thyroxine, on\_antithyroid\_meds, sick, pregnant,

            thyroid\_surgery, I131\_treatment, query\_hypothyroid, query\_hyperthyroid, lithium,

            goitre, tumor, hypopituitary, psych, TSH, T3, TT4, T4U, FTI]

# Function to predict the diagnosis based on inputs

def predict\_diagnosis(inputs):

    # Assuming 'model' is a trained machine learning model

    # Replace 'model.predict()' with the actual function to make predictions

    output = model.predict([inputs])[0]

    return output

# Streamlit app

def main():

    # Title

    st.markdown(title\_css, unsafe\_allow\_html=True)

    # Sidebar

    st.sidebar.write("<h1 style='color: #F63366; font-size: 36px;'>THYROID DETECTION</h1>", unsafe\_allow\_html=True)

    st.sidebar.title("About Project :")

    st.sidebar.write("This Streamlit app serves as a Thyroid Diagnosis Predictor. It utilizes machine learning to predict thyroid diagnosis based on various patient attributes such as age, sex, medical history, and laboratory test results. Users can input patient data and receive an immediate diagnosis prediction, helping medical professionals make informed decisions efficiently.")

    st.sidebar.title("Attributes Information :")

    st.sidebar.write("""

        - Age: Age of the patient (int)

        - Sex: Sex patient identifies (str)

        - On Thyroxine: Whether patient is on thyroxine (bool)

        - Query on Thyroxine: Whether patient is on thyroxine (bool)

        - On Antithyroid Meds: Whether patient is on antithyroid meds (bool)

        - Sick: Whether patient is sick (bool)

        - Pregnant: Whether patient is pregnant (bool)

        - Thyroid Surgery: Whether patient has undergone thyroid surgery (bool)

        - I131 Treatment: Whether patient is undergoing I131 treatment (bool)

        - Query Hypothyroid: Whether patient believes they have hypothyroid (bool)

        - Query Hyperthyroid: Whether patient believes they have hyperthyroid (bool)

        - Lithium: Whether patient takes lithium (bool)

        - Goitre: Whether patient has goitre (bool)

        - Tumor: Whether patient has tumor (bool)

        - Hypopituitary: Whether patient has hyperpituitary gland (float)

        - Psych: Whether patient is psych (bool)

        - TSH: TSH level in blood from lab work (float)

        - T3: T3 level in blood from lab work (float)

        - TT4: TT4 level in blood from lab work (float)

        - T4U: T4U level in blood from lab work (float)

        - FTI: FTI level in blood from lab work (float)

    """)

    # Input fields

    col1, col2, col3 = st.columns(3)

    with col1:

        age = st.number\_input('Age', value=None)

        query\_on\_thyroxine = st.selectbox('Query On Thyroxine', options=['', 'No', 'Yes'])

        pregnant = st.selectbox('Pregnant', options=['', 'No', 'Yes'])

        query\_hypothyroid = st.selectbox('Query Hypothyroid', options=['', 'No', 'Yes'])

        goitre = st.selectbox('Goitre', options=['', 'No', 'Yes'])

        psych = st.selectbox('Psych', options=['', 'No', 'Yes'])

        TT4 = st.number\_input('TT4', value=None)

    with col2:

        sex = st.selectbox('Sex', options=['', 'M', 'F'])

        on\_antithyroid\_meds = st.selectbox('On Antithyroid Meds', options=['', 'No', 'Yes'])

        thyroid\_surgery = st.selectbox('Thyroid Surgery', options=['', 'No', 'Yes'])

        query\_hyperthyroid = st.selectbox('Query Hyperthyroid', options=['', 'No', 'Yes'])

        tumor = st.selectbox('Tumor', options=['', 'No', 'Yes'])

        TSH = st.number\_input('TSH', value=None)

        T4U = st.number\_input('T4U', value=None)

    with col3:

        on\_thyroxine = st.selectbox('On Thyroxine', options=['', 'No', 'Yes'])

        sick = st.selectbox('Sick', options=['', 'No', 'Yes'])

        I131\_treatment = st.selectbox('I131 Treatment', options=['', 'No', 'Yes'])

        lithium = st.selectbox('Lithium', options=['', 'No', 'Yes'])

        hypopituitary = st.selectbox('Hypopituitary', options=['', 'No', 'Yes'])

        T3 = st.number\_input('T3', value=None)

        FTI = st.number\_input('FTI', value=None)

    # Detect button

    with col2:

        detect\_button = st.button('Detect', key='predict\_button')

        detect\_button\_container = st.container()

        with detect\_button\_container:

            detect\_button\_css = f"""

                <style>

                    .stButton > button:first-child {{

                        width: 100%;

                        color: white;

                        border-color: {detect\_button\_color};

                        border-radius: 5px;

                        padding: 10px;

                    }}

                </style>

            """

            st.markdown(detect\_button\_css, unsafe\_allow\_html=True)

        if detect\_button:

            # Preprocess inputs

            inputs = preprocess\_inputs(age, sex, on\_thyroxine, query\_on\_thyroxine, on\_antithyroid\_meds, sick,

                                       pregnant,

                                       thyroid\_surgery, I131\_treatment, query\_hypothyroid, query\_hyperthyroid,

                                       lithium,

                                       goitre, tumor, hypopituitary, psych, TSH, T3, TT4, T4U, FTI)

            # Get prediction

            diagnosis\_num = predict\_diagnosis(inputs)

            diagnosis\_label = diagnoses.get(diagnosis\_num, 'Unknown')

            st.markdown(

                f"<h1 style='text-align: center; color: {diagnosis\_color};'>{diagnosis\_label}</h1>",

                unsafe\_allow\_html=True)

if \_\_name\_\_ == '\_\_main\_\_':

    main()