import streamlit as st

import pandas as pd

import joblib

from datetime import datetime, timedelta

import spacy

import os

# ===========================

# Load Models and Encoders

# ===========================

MODEL\_DIR = "models"

priority\_model = joblib.load(os.path.join(MODEL\_DIR, "newpriority\_predictor.pkl"))

resolution\_model = joblib.load(os.path.join(MODEL\_DIR, "resolution\_time\_predictorr.pkl"))

le\_area = joblib.load(os.path.join(MODEL\_DIR, "labelencoder\_area.pkl"))

le\_type = joblib.load(os.path.join(MODEL\_DIR, "labelencoder\_type.pkl"))

le\_priority\_output = joblib.load(os.path.join(MODEL\_DIR, "newpriority\_label\_encoder.pkl"))

scaler = joblib.load(os.path.join(MODEL\_DIR, "scalerr.pkl"))

X\_columns = joblib.load(os.path.join(MODEL\_DIR, "x\_columns.pkl"))

nlp = spacy.load("en\_core\_web\_sm")

# ===========================

# Department Mapping

# ===========================

type\_to\_department = {

"Garbage Issue": "Municipal Waste Management Department",

"Electricity Issue": "Electricity Distribution Department",

"Illegal Parking": "Traffic and Transportation Department",

"Tree Falling": "Urban Forestry and Landscaping Department",

"Road Damage": "Public Works Department",

"Animal Nuisance": "Animal Control and Welfare Department",

"Streetlight Issue": "Urban Lighting and Infrastructure Department",

"Noise Pollution": "Environmental Protection Department",

"Water Issue": "Water Supply and Sanitation Department",

"Sewage Problem": "Sewerage and Drainage Department"

}

# ===========================

# UI Styling & Layout

# ===========================

st.set\_page\_config(page\_title="Smart Complaint Management System", page\_icon="📣", layout="wide")

# Custom CSS to enhance the page styling

st.markdown("""

<style>

.big-font {

font-size: 24px !important;

font-weight: bold;

color: #ff6347;

}

.header-font {

font-size: 28px !important;

color: #008080;

}

.section-header {

font-size: 22px !important;

color: #4169e1;

}

.card {

background-color: #f0f8ff;

padding: 20px;

border-radius: 10px;

margin-top: 20px;

box-shadow: 0 4px 8px rgba(0, 0, 0, 0.2);

}

</style>

""", unsafe\_allow\_html=True)

# Title

st.title("📣 \*\*Smart Complaint Management System\*\*")

st.markdown("### \*\*Submit Your Complaint and Predict Resolution Time\*\*")

st.markdown("---")

# Step 1: Complaint Filing

st.header("📝 \*\*Step 1: File a Complaint\*\*")

st.markdown("---")

with st.form("complaint\_form"):

complaint\_text = st.text\_area("• Complaint Description", "Uncollected garbage near XYZ lane in Hadapsar, causing health hazard.")

area = st.selectbox("• Area", le\_area.classes\_)

problem\_type = st.selectbox("• Problem Type", le\_type.classes\_)

filing\_date = st.date\_input("• Filing Date", datetime.today())

submitted = st.form\_submit\_button("Submit Complaint")

if submitted:

st.session\_state['complaint\_text'] = complaint\_text

st.session\_state['area'] = area

st.session\_state['problem\_type'] = problem\_type

st.session\_state['filing\_date'] = filing\_date

department = type\_to\_department.get(problem\_type, "General Complaints Department")

input\_df = pd.DataFrame([{

"complaint\_text": complaint\_text,

"type": problem\_type,

"area": area,

"type\_emphasis": problem\_type

}]).fillna('missing').astype(str)

priority\_encoded = priority\_model.predict(input\_df)[0]

priority = le\_priority\_output.inverse\_transform([priority\_encoded])[0]

st.session\_state['department'] = department

st.session\_state['priority'] = priority

st.subheader("📍 \*\*Step 2: Classification Results\*\*")

st.write(f"• 🏢 \*\*Assigned Department\*\*: {department}")

st.write(f"• 🔺 \*\*Predicted Priority\*\*: {priority}")

st.markdown("---")

st.markdown("<h4 class='big-font'>Your complaint has been successfully filed! 🎉</h4>", unsafe\_allow\_html=True)

# Step 3: ETA Prediction

if 'department' in st.session\_state and st.button("▶️ \*\*Predict Estimated Resolution Time (ETA)\*\*"):

st.subheader("⏳ \*\*Step 3: Predicted Resolution Time\*\*")

st.markdown("---")

eta\_input = pd.DataFrame([{

"area": st.session\_state['area'],

"problem\_type": st.session\_state['problem\_type'],

"department": st.session\_state['department'],

"filing\_year": st.session\_state['filing\_date'].year,

"filing\_month": st.session\_state['filing\_date'].month,

"filing\_day": st.session\_state['filing\_date'].day

}])

eta\_input = pd.get\_dummies(eta\_input)

eta\_input = eta\_input.reindex(columns=X\_columns, fill\_value=0).astype(float)

eta\_scaled = scaler.transform(eta\_input)

predicted\_days = resolution\_model.predict(eta\_scaled)[0]

expected\_completion = st.session\_state['filing\_date'] + timedelta(days=predicted\_days)

st.write(f"📅 \*\*Predicted Resolution Time\*\*: {predicted\_days:.1f} days")

st.write(f"✅ \*\*Expected Completion Date\*\*: {expected\_completion.strftime('%d %B %Y')}")

st.success("🎉 \*\*Complaint processed successfully!\*\*")

st.info("💰 \*\*Check Budget Allocation: Go to Budget\_Analysis.py\*\*")

# Add Call-to-Action Button

st.markdown("""

<div style='text-align:center;'>

<a href="https://your-link-here.com" target="\_blank">

<button style="background-color:#008080; color:white; padding:10px 20px; border-radius:5px; font-size:16px;">

Visit Budget Analysis

</button>

</a>

</div>

""", unsafe\_allow\_html=True)