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

# CYBERBULLYING DETECTION AND PREVENTION SYSTEM

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

# Author: Your Name

# Description:

# This script detects cyberbullying in text messages using

# TF-IDF + Logistic Regression.

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

import pandas as pd

import re

from sklearn.feature\_extraction.text import TfidfVectorizer

from sklearn.linear\_model import LogisticRegression

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

from sklearn.metrics import classification\_report, confusion\_matrix, accuracy\_score

from sklearn.pipeline import Pipeline

import joblib

# ---------------------------

# STEP 1: TEXT PREPROCESSING

# ---------------------------

STOPWORDS = {

"a","an","the","and","or","but","if","while","is","are","was","were","this","that",

"to","for","of","in","on","at","by","from","with","be","as","it","i","you","he","she",

"they","we","my","your","yours","me","mine","our","ours","their","theirs","just","so",

"not","no","will","can","could","would","should","do","did","does","have","has","had"

}

def simple\_preprocess(text: str) -> str:

"""Clean text by removing URLs, mentions, special chars, and stopwords."""

if not isinstance(text, str):

return ""

text = text.lower()

text = re.sub(r'http\S+|www\.\S+', ' ', text) # remove URLs

text = re.sub(r'@\w+', ' ', text) # remove mentions

text = re.sub(r'#', ' ', text) # remove #

text = re.sub(r'[^a-z\s]', ' ', text) # keep only letters

text = re.sub(r'\s+', ' ', text).strip()

tokens = [t for t in text.split() if t not in STOPWORDS and len(t) > 1]

return " ".join(tokens)

# ---------------------------

# STEP 2: SAMPLE DATASET

# ---------------------------

data = [

("You're such an idiot, nobody likes you", 1),

("I love your work, keep it up!", 0),

("Go kill yourself", 1),

("What a beautiful picture", 0),

("You're disgusting and worthless", 1),

("Great job on the project team!", 0),

("Nobody wants you here, leave", 1),

("Let's meet tomorrow at 5pm", 0),

("I hate you so much", 1),

("Thanks for the help, appreciated", 0),

("You're a pathetic loser", 1),

("Congratulations on your win!", 0)

]

df = pd.DataFrame(data, columns=["text", "label"])

df["text\_clean"] = df["text"].apply(simple\_preprocess)

# ---------------------------

# STEP 3: TRAIN-TEST SPLIT

# ---------------------------

X = df["text\_clean"]

y = df["label"]

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.25, random\_state=42, stratify=y)

# ---------------------------

# STEP 4: MODEL PIPELINE

# ---------------------------

model = Pipeline([

("tfidf", TfidfVectorizer(ngram\_range=(1,2), min\_df=1)),

("clf", LogisticRegression(max\_iter=500))

])

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

# ---------------------------

# STEP 5: EVALUATION

# ---------------------------

y\_pred = model.predict(X\_test)

print("\n================ EVALUATION RESULTS ================")

print(f"Accuracy: {accuracy\_score(y\_test, y\_pred):.3f}")

print("Confusion Matrix:\n", confusion\_matrix(y\_test, y\_pred))

print("Classification Report:\n", classification\_report(y\_test, y\_pred, target\_names=["Non-Bullying", "Bullying"]))

# ---------------------------

# STEP 6: SAVE MODEL

# ---------------------------

joblib.dump(model, "cyberbullying\_model.joblib")

print("\n✅ Model saved as 'cyberbullying\_model.joblib'")

# ---------------------------

# STEP 7: TEST NEW MESSAGES

# ---------------------------

def predict\_texts(texts):

"""Predict whether given texts are bullying or not."""

texts\_clean = [simple\_preprocess(t) for t in texts]

preds = model.predict(texts\_clean)

probs = model.predict\_proba(texts\_clean)

for i, text in enumerate(texts):

label = "Cyberbullying" if preds[i] == 1 else "Not Bullying"

confidence = probs[i][preds[i]] \* 100

print(f"\nText: {text}\n→ Prediction: {label} ({confidence:.2f}% confidence)")

# Example predictions

sample\_texts = [

"I will punch you tomorrow!",

"You are awesome and talented!",

"Nobody likes you, just disappear!"

]

print("\n================ SAMPLE PREDICTIONS ================")

predict\_texts(sample\_texts)

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

# END OF PROGRAM

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