import nltk

import pandas as pd

from sklearn.feature\_extraction.text import TfidfVectorizer

from sklearn.preprocessing import LabelEncoder

from nltk.corpus import stopwords

from nltk.tokenize import word\_tokenize

from nltk.stem import WordNetLemmatizer

# Sample data

data = {

'text': ["This is the first document.",

"This document is the second document.",

"And this is the third one.",

"Is this the first document?"],

'label': ['A', 'B', 'C', 'A']

}

# Convert data to DataFrame

df = pd.DataFrame(data)

# Text Cleaning, Lemmatization, and Stop Words Removal

lemmatizer = WordNetLemmatizer()

stop\_words = set(stopwords.words('english'))

def clean\_text(text):

tokens = word\_tokenize(text.lower()) # Tokenize and convert to lowercase

tokens = [token for token in tokens if token.isalnum()] # Remove non-alphanumeric characters

tokens = [lemmatizer.lemmatize(token) for token in tokens] # Lemmatize

tokens = [token for token in tokens if token not in stop\_words] # Remove stop words

return ' '.join(tokens)

df['clean\_text'] = df['text'].apply(clean\_text)

# Label Encoding

label\_encoder = LabelEncoder()

df['encoded\_label'] = label\_encoder.fit\_transform(df['label'])

# TF-IDF Representation

tfidf\_vectorizer = TfidfVectorizer()

tfidf\_matrix = tfidf\_vectorizer.fit\_transform(df['clean\_text'])

tfidf\_df = pd.DataFrame(tfidf\_matrix.toarray(), columns=tfidf\_vectorizer.get\_feature\_names\_out())

# Save Outputs

df[['clean\_text', 'label', 'encoded\_label']].to\_csv('cleaned\_data.csv', index=False)

tfidf\_df.to\_csv('tfidf\_representation.csv', index=False)

print("Outputs saved successfully.")