import nltk

from nltk.stem import WordNetLemmatizer

from sklearn.feature\_extraction.text import TfidfVectorizer

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

from sklearn.linear\_model import LogisticRegression

import pandas as pd

def load\_data(file\_path):

"""Loads data from a CSV file.

Args:

file\_path (str): Path to the CSV file.

Returns:

pandas.DataFrame: Dataframe containing the loaded data.

"""

data = pd.read\_csv('/content/ vehical.csv')

return data

def preprocess\_text(text):

"""Preprocesses text data for NLP.

Args:

text (str): Text to preprocess.

Returns:

list: Preprocessed list of tokens.

"""

lemmatizer = WordNetLemmatizer()

tokens = nltk.word\_tokenize(text.lower())

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

return lemmas

def train\_model(data):

"""Trains an NLP model for intent classification.

Args:

data (pandas.DataFrame): Dataframe containing the data.

Returns:

tuple: A tuple containing the trained model and vectorizer.

"""

# Replace 'query' with the actual name of the column

# containing user queries in your CSV file

X = data['user\_query'].apply(preprocess\_text)

y = data['intent']

vectorizer = TfidfVectorizer()

X = vectorizer.fit\_transform(X)

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

model = LogisticRegression()

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

return model, vectorizer

def get\_response(query, model, vectorizer):

"""Generates a response based on the given query.

Args:

query (str): The user's query.

model (sklearn.linear\_model.LogisticRegression): The trained NLP model.

vectorizer (sklearn.feature\_extraction.text.TfidfVectorizer): The TF-IDF vectorizer.

Returns:

str: The generated response.

"""

preprocessed\_query = preprocess\_text(query)

predicted\_intent = model.predict(vectorizer.transform([preprocessed\_query]))[0]

# Replace this with your actual response generation logic based on the intent

response = f"Predicted intent: {predicted\_intent}"

return response

if \_\_name\_\_ == "\_\_main\_\_":

file\_path = "your\_dataset.csv" # Replace with your dataset path

data = load\_data(file\_path)

model, vectorizer = train\_model(data)

while True:

query = input("Enter your query: ")

if query.lower() == "quit":

break

response = get\_response(query, model, vectorizer)

print("Response:", response)