

GuiDP

Code:

```
import tkinter as tk

from tkinter import messagebox

import pandas as pd

from sklearn.feature_extraction.text import TfidfVectorizer

from sklearn.model_selection import train_test_split

from sklearn.naive_bayes import MultinomialNB

from sklearn.metrics import accuracy_score, classification_report


# ฟังก์ชันสำหรับการประมวลผลและฝึกสอนโมเดล

def train_spam_filter():

    # โหลดข้อมูล (ใช้ชุดข้อมูล SMS Spam Collection)

    url = "https://raw.githubusercontent.com/mohitgupta-omg/Kaggle-SMS-spam-collection-dataset/master/spam.csv"

    data = pd.read_csv(url, encoding='latin-1')

    # แปลงข้อความสแปมเป็น 1 และไม่เป็นสแปมเป็น 0

    data['v1'] = data['v1'].map({'spam': 1, 'ham': 0})

    # แยกข้อมูลเป็นชุดฝึกสอนและชุดทดสอบ

    X_train, X_test, y_train, y_test = train_test_split(data['v2'], data['v1'],
test_size=0.2, random_state=42)

    # ทำการแปลงข้อความเป็นเวกเตอร์ด้วย TF-IDF

    vectorizer = TfidfVectorizer(stop_words='english')

    X_train_tfidf = vectorizer.fit_transform(X_train)

    X_test_tfidf = vectorizer.transform(X_test)
```

สร้างโมเดล Naive Bayes

model = MultinomialNB()

model.fit(X_train_tfidf, y_train)

ทดสอบโมเดล

y_pred = model.predict(X_test_tfidf)

accuracy = accuracy_score(y_test, y_pred)

print(f"Accuracy: {accuracy:.4f}")

return model, vectorizer

ฝึกสอนโมเดล

model, vectorizer = train_spam_filter()

ฟังก์ชันสำหรับการสแกนข้อความสแปม

def scan_spam():

message = message_entry.get("1.0", tk.END).strip()

if not message:

result_text.insert(tk.END, "กรุณาใส่ข้อความที่ต้องการตรวจสอบ\n")

return

แปลงข้อความเป็น TF-IDF เวกเตอร์

message_tfidf = vectorizer.transform([message])

ทำนายว่าข้อความเป็นสแปมหรือไม่

```
prediction = model.predict(message_tfidf)
```

```
# แสดงผลลัพธ์
```

```
if prediction == 1:
```

```
    result_text.insert(tk.END, "ข้อความนี้เป็นสแปม\n")
```

```
else:
```

```
    result_text.insert(tk.END, "ข้อความนี้ไม่เป็นสแปม\n")
```

```
# ฟังก์ชันสำหรับการล้างผลลัพธ์
```

```
def clear_results():
```

```
    result_text.delete("1.0", tk.END)
```

```
# ฟังก์ชันสำหรับการแสดงข้อความ About
```

```
def show_about():
```

```
    messagebox.showinfo("About", "โปรแกรมนี้เป็นเครื่องมือสำหรับการกรอง  
ข้อความสแปม")
```

```
# สร้างหน้าต่างหลัก
```

```
root = tk.Tk()
```

```
root.title("Spam Filter Tool")
```

```
root.geometry("600x400")
```

```
# กรอบข้อความ
```

```
message_frame = tk.Frame(root)
```

```
message_frame.pack(pady=10)
```

```
message_label = tk.Label(message_frame, text="ใส่ข้อความที่ต้องการตรวจสอบ:")
```

```
message_label.pack()
```

```
message_entry = tk.Text(message_frame, height=5, width=50)
```

```
message_entry.pack()
```

```
scan_spam_button = tk.Button(message_frame, text="ตรวจสอบข้อความ",  
command=scan_spam)
```

```
scan_spam_button.pack(pady=5)
```

```
# แสดงผลลัพธ์การสแกน
```

```
result_frame = tk.Frame(root)
```

```
result_frame.pack(pady=10)
```

```
result_label = tk.Label(result_frame, text="ผลการสแกน:")
```

```
result_label.pack()
```

```
result_text = tk.Text(result_frame, height=10, width=70)
```

```
result_text.pack()
```

```
# ปุ่มล้างผลลัพธ์
```

```
clear_button = tk.Button(root, text="ล้างผลลัพธ์", command=clear_results)
```

```
clear_button.pack(pady=5)
```

```
# เมนู About
```

```
menu = tk.Menu(root)
```

```
root.config(menu=menu)
```

```
help_menu = tk.Menu(menu)

menu.add_cascade(label="Help", menu=help_menu)

help_menu.add_command(label="About", command=show_about)


root.mainloop()
```

การทำงานของโค้ด:

1. การฝึกสอนโมเดล:

- ใช้ชุดข้อมูล SMS Spam Collection ในการฝึกสอนโมเดล Naive Bayes สำหรับการจำแนกข้อความว่าเป็นสแปมหรือไม่
- ใช้ TF-IDF Vectorizer ในการแปลงข้อความเป็นเวกเตอร์เพื่อให้ง่ายต่อการนำเข้าโมเดล
- โมเดลและเวกเตอร์ที่ฝึกสอนแล้วจะถูกนำมาใช้ใน GUI เพื่อทำการตรวจสอบข้อความ

2. การใช้งาน GUI:

- ผู้ใช้สามารถป้อนข้อความที่ต้องการตรวจสอบในช่อง message_entry
- เมื่อกดปุ่ม "ตรวจสอบข้อความ" ระบบจะทำการแปลงข้อความและทำนายว่าข้อความ เป็นสแปมหรือไม่
- ผลลัพธ์จะแสดงในกรอบ result_text ว่าข้อความนั้นเป็น "สแปม" หรือ "ไม่เป็นสแปม"

3. การแสดงผล:

- ผลลัพธ์จะแสดงในช่องกรอบข้อความ หากผู้ใช้ต้องการล้างผลลัพธ์สามารถกดปุ่ม "ล้างผลลัพธ์" ได้

โค้ดนี้จะทำงานในรูปแบบ GUI โดยผู้ใช้สามารถป้อนข้อความและทดสอบว่าเป็นสแปมหรือไม่ผ่านอินเทอร์เฟซที่ใช้งานง่าย

ปัญหาที่พบบ่อยระหว่างการทดสอบ

คือ `ModuleNotFoundError` ซึ่งเกิดจากการที่ Python ไม่สามารถหาไลบรารี `scikit-learn` (ที่มีโมดูล `sklearn`) ในระบบของคุณได้

แก้ไขโดยการ ติดตั้งไลบรารี `scikit-learn` ในสภาพแวดล้อมการทำงานของ Python

คำสั่งติดตั้ง `scikit-learn`:

```
pip install scikit-learn
```

กรณีต้องการระบุให้ติดตั้งใน Python 3 โดยเฉพาะ:

```
pip3 install scikit-learn
```

เปิดใช้งานสภาพแวดล้อมนั้นก่อนการติดตั้งแพ็คเกจ:(กรณีลืมเปิด)

```
source venv/bin/activate # สำหรับ Linux/MacOS
```

```
.\venv\Scripts\activate # สำหรับ Windows
```

ปัญหาจากการที่ Python ไม่สามารถหาไลบรารี pandas ในระบบได้:

คำสั่งติดตั้ง pandas:

```
pip install pandas
```

กรณีต้องการระบุให้ติดตั้งใน Python 3 โดยเฉพาะ:

```
pip3 install pandas
```