โค้ดการสร้าง ระบบกรองสแปม ด้วยการใช้ Machine Learning และการประมวลผลภาษาธรรมชาติ (NLP) โดยทำงานผ่านการใช้ชุดข้อมูลข้อความที่เป็นสแปมและไม่เป็นสแปม (ham) และสร้างโมเดลสำหรับจำแนกข้อความว่าเป็นสแปมหรือไม่

ขั้นตอนการทำงานของโค้ด:

1. โหลดข้อมูล

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

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

โค้ดทำการโหลดข้อมูลจาก URL ซึ่งเป็นชุดข้อมูลข้อความที่ประกอบไปด้วยข้อความ SMS และระบุว่าข้อความนั้นเป็น "สแปม" (spam) หรือ "ไม่เป็นสแปม" (ham)

ใช้ pandas ในการอ่านข้อมูลจากไฟล์ CSV และใช้ encoding='latin-1' เพื่อจัดการกับอักขระพิเศษในข้อความ

---

2. แปลงประเภทข้อมูล

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

คอลัมน์ v1 ในชุดข้อมูลจะมีค่าเป็น "spam" หรือ "ham"

ระบบทำการแปลงข้อมูลนี้ให้เป็นตัวเลขเพื่อให้โมเดลสามารถประมวลผลได้ง่าย โดยกำหนดค่าให้กับ "spam" เป็น 1 และ "ham" เป็น 0

---

3. แบ่งข้อมูลเป็นชุดฝึกสอนและชุดทดสอบ

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

v2 คือคอลัมน์ที่เก็บข้อความ SMS ส่วน v1 คือคอลัมน์ที่เก็บป้ายกำกับ (spam หรือ ham)

ข้อมูลถูกแบ่งออกเป็นสองชุด:

- **ชุดฝึกสอน (Training Set): ใช้ในการฝึกสอนโมเดล (80% ของข้อมูล)**
- **ชุดทดสอบ (Test Set): ใช้ในการประเมินผลโมเดล (20% ของข้อมูล)**

---

4. แปลงข้อความเป็นเวกเตอร์ด้วย TF-IDF

```
vectorizer = TfidfVectorizer(stop_words='english')

X_train_tfidf = vectorizer.fit_transform(X_train)

X_test_tfidf = vectorizer.transform(X_test)
```

ข้อความในรูปแบบตัวอักษรไม่สามารถถูกประมวลผลได้โดยตรงในโมเดลการเรียนรู้ของเครื่อง ดังนั้นจึงต้องแปลงข้อความเป็นเวกเตอร์เชิงตัวเลข

เทคนิค TF-IDF (Term Frequency-Inverse Document Frequency) ถูกใช้เพื่อแปลงข้อความ โดย TF-IDF จะให้ค่าความสำคัญกับคำที่ปรากฏในข้อความขึ้นอยู่กับความถี่ของมัน

fit_transform() ใช้กับข้อมูลฝึกสอนเพื่อสร้างเวกเตอร์ และ transform() ใช้กับข้อมูลทดสอบเพื่อให้ข้อมูลทดสอบมีรูปแบบที่สอดคล้องกับข้อมูลฝึกสอน

---

5. สร้างโมเดล Naive Bayes

```
model = MultinomialNB()

model.fit(X_train_tfidf, y_train)
```

สร้างโมเดล Naive Bayes (MultinomialNB) ซึ่งเป็นโมเดลสำหรับการจำแนกข้อความ โดยเฉพาะในงานที่เกี่ยวข้องกับการจำแนกข้อความสแปม

โมเดลนี้ถูกฝึกสอนด้วยชุดข้อมูลฝึกสอน (X_train_tfidf และ y_train)

---

6. ทำการทดสอบโมเดลกับข้อมูลทดสอบ

```
y_pred = model.predict(X_test_tfidf)
```

โมเดลที่ถูกฝึกสอนแล้วจะถูกใช้ทำนายข้อมูลทดสอบ โดยการทำนายว่าข้อความในชุดทดสอบเป็นสแปมหรือไม่

ผลลัพธ์ของการทำนายจะถูกบันทึกในตัวแปร y_pred

---

7. แสดงผลลัพธ์ความแม่นยำและรายงานการจำแนกประเภท

```
print(f"Accuracy: {accuracy_score(y_test, y_pred):.4f}")

print("Classification Report:")

print(classification_report(y_test, y_pred))
```

ใช้เมตริก accuracy_score เพื่อวัดความแม่นยำของโมเดล โดยคำนวณจากผลลัพธ์ที่โมเดลทำนาย (y_pred) เทียบกับค่าจริง (y_test)

ใช้ classification_report เพื่อแสดงรายละเอียดเพิ่มเติม เช่น precision, recall, และ f1-score ของการจำแนกข้อความสแปมและไม่เป็นสแปม

สรุป:

โค้ดนี้ใช้เทคนิค NLP และ Machine Learning เพื่อสร้างระบบกรองสแปม โดยทำการฝึกสอนโมเดล Naive Bayes จากข้อมูลข้อความที่เป็นสแปมและไม่เป็นสแปม โมเดลถูกทดสอบและประเมินความแม่นยำด้วยข้อมูลทดสอบ