Coding Preprocessing

```python
import pandas as pd
import re
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.decomposition import PCA
from sklearn.preprocessing import MinMaxScaler
from collections import Counter

# ฟังก์ชันทำความสะอาดข้อความ
Codeium: Refactor | Explain | Generate Docstring | ×
def clean_text(text):
    text = re.sub(r'http\S+', '', text)  # ลบลิงก์
    text = re.sub(r'@\w+', '', text)  # ลบการกล่าวถึง (@mention)
    text = re.sub(r'#\w+', '', text)  # ลบแฮชแท็ก (#hashtag)
    text = re.sub(r'\d+', '', text)  # ลบตัวเลข
    text = re.sub(r'[^\w\s]', '', text)  # ลบอักขระพิเศษ
    text = re.sub(r'\s+', ' ', text)  # ลบช่องว่างที่ไม่จำเป็น
    return text.strip()
```

```python
# ฟังก์ชันเพิ่มฟีเจอร์
Codeium: Refactor | Explain | Generate Docstring | ×
def add_empirical_features(data):
    data['tweet_length'] = data['OriginalTweet'].apply(len)
    data['word_count'] = data['OriginalTweet'].apply(lambda x: len(x.split()))
    data['sentence_count'] = data['OriginalTweet'].apply(lambda x: len(re.split(r'[.!?]+', x)) - 1)
    return data[['tweet_length', 'word_count', 'sentence_count']]

# ฟังก์ชันโหลดและเตรียมข้อมูล
Codeium: Refactor | Explain | Generate Docstring | ×
def load_and_prepare_data(file_path, is_train=True):
    data = pd.read_csv(file_path, encoding='latin1')  # โหลดข้อมูลจากไฟล์ CSV ด้วย encoding 'latin1'
    data.dropna(subset=['OriginalTweet'], inplace=True)  # ลบแถวที่มีค่าหายไปในคอลัมน์ 'OriginalTweet'
    if is_train:
        data.dropna(subset=['Sentiment'], inplace=True)  # ลบแถวที่มีค่าหายไปในคอลัมน์ 'Sentiment'
    sentiment_mapping = {'Extremely Positive': 2, 'Positive': 1, 'Neutral': 0, 'Negative': -1, 'Extremely Negative': -2}  # mapping ค่าของ Sentiment
    เป็นตัวเลข
    data['Sentiment'] = data['Sentiment'].map(sentiment_mapping)  # แปลงค่า Sentiment เป็นตัวเลข
    data['OriginalTweet'] = data['OriginalTweet'].apply(clean_text).str.lower()  # ทำความสะอาดข้อความและแปลงเป็นตัวพิมพ์เล็กทั้งหมด
    return data
```

```python
# ฟังก์ชันแปลงข้อความเป็น TF-IDF features
Codeium: Refactor | Explain | Generate Docstring | ×
def transform_to_tfidf(train_data, test_data):
    vectorizer = TfidfVectorizer(
        max_features=1000,  # กำหนดจำนวนฟีเจอร์สูงสุดเป็น 1000
        ngram_range=(1, 2),  # ใช้ทั้ง 1-grams และ 2-grams
        norm='l2',  # ใช้การนอร์ม L2
        use_idf=True,  # ใช้ IDF
        smooth_idf=True,
        sublinear_tf=True,
        min_df=2,  # คำต้องปรากฏอย่างน้อยใน 2 เอกสาร
        max_df=0.5  # คำต้องปรากฏไม่เกิน 50% ของเอกสารทั้งหมด
    )
    X_train = vectorizer.fit_transform(train_data['OriginalTweet'])  # แปลงข้อความในข้อมูลฝึกเป็น TF-IDF features
    X_test = vectorizer.transform(test_data['OriginalTweet'])  # แปลงข้อความในข้อมูลทดสอบเป็น TF-IDF features
    return X_train, X_test, vectorizer
```

```python
# ฟังก์ชันลดมิติของข้อมูลโดยใช้ PCA
Codeium: Refactor | Explain | Generate Docstring | ×
def apply_pca(X_train, X_test, n_components=100):
    pca = PCA(n_components=n_components)
    X_train_pca = pca.fit_transform(X_train.toarray())  # ลดมิติของข้อมูลฝึกโดยใช้ PCA
    X_test_pca = pca.transform(X_test.toarray())  # ลดมิติของข้อมูลทดสอบโดยใช้ PCA
    return X_train_pca, X_test_pca

# ฟังก์ชันปรับขนาดข้อมูลให้อยู่ในช่วง 0-1
Codeium: Refactor | Explain | Generate Docstring | ×
def scale_data(X_train, X_test):
    X_train.columns = X_train.columns.astype(str)
    X_test.columns = X_test.columns.astype(str)
    scaler = MinMaxScaler()
    X_train_scaled = scaler.fit_transform(X_train)
    X_test_scaled = scaler.transform(X_test)
    return X_train_scaled, X_test_scaled
```

```python
# ฟังก์ชันบันทึกข้อมูลลงไฟล์ CSV
Codeium: Refactor | Explain | Generate Docstring | ✕
def save_to_csv(X, y, output_path):
    df = pd.DataFrame(X)  # สร้าง DataFrame จาก TF-IDF features ที่ถูกลดมิติแล้ว
    if y is not None:
        df['Sentiment'] = y.values  # เพิ่มคอลัมน์ 'Sentiment' ลงใน DataFrame
    df.to_csv(output_path, index=False)  # บันทึก DataFrame ลงในไฟล์ CSV โดยไม่ใส่ index
    print(f'TF-IDF features และค่าที่ได้ถูกบันทึกใน {output_path}')  # แสดงข้อความยืนยันการบันทึกไฟล์

# เส้นทางไฟล์ข้อมูลและไฟล์ผลลัพธ์
train_file_path = r'C:\Users\Windows 11\Desktop\CoronaML\Corona_NLP_train.csv'
test_file_path = r'C:\Users\Windows 11\Desktop\CoronaML\Corona_NLP_test.csv'
train_output_path = r'C:\Users\Windows 11\Desktop\CoronaML\preprocessed_Corona_NLP_train.csv'
test_output_path = r'C:\Users\Windows 11\Desktop\CoronaML\preprocessed_Corona_NLP_test.csv'

# โหลดและเตรียมข้อมูล
train_data = load_and_prepare_data(train_file_path, is_train=True)
test_data = load_and_prepare_data(test_file_path, is_train=False)
```

```python
# เพิ่มฟีเจอร์
train_empirical_features = add_empirical_features(train_data)
test_empirical_features = add_empirical_features(test_data)

# ตรวจสอบการกระจายของคำในข้อความฝึก
all_text = ' '.join(train_data['OriginalTweet'])
word_counts = Counter(all_text.split())
print("คำที่พบมากที่สุด 10 อันดับ:")
print(word_counts.most_common(10))

# ดูตัวอย่างข้อความ
print("ข้อความตัวอย่าง:")
print(train_data['OriginalTweet'].head())
```

```python
# แปลงข้อความเป็น TF-IDF features
X_train_tfidf, X_test_tfidf, vectorizer = transform_to_tfidf(train_data, test_data)
# ลดมิติของข้อมูลโดยใช้ PCA
X_train_pca, X_test_pca = apply_pca(X_train_tfidf, X_test_tfidf)

# รวมฟีเจอร์กับ TF-IDF features ที่ผ่านการลดมิติแล้ว
X_train_combined = pd.concat([pd.DataFrame(X_train_pca), train_empirical_features.reset_index(drop=True)], axis=1)
X_test_combined = pd.concat([pd.DataFrame(X_test_pca), test_empirical_features.reset_index(drop=True)], axis=1)
# ปรับขนาดข้อมูลให้อยู่ในช่วง 0-1
X_train_scaled, X_test_scaled = scale_data(X_train_combined, X_test_combined)

# ตรวจสอบจำนวนฟีเจอร์ที่ถูกสร้างขึ้น
print("จำนวนฟีเจอร์ที่ถูกสร้างขึ้น:", len(vectorizer.get_feature_names_out()))
```

```python
# แสดงตัวอย่างของ TF-IDF features และฟีเจอร์
print("TF-IDF features ตัวอย่าง (หลังจากทำ PCA และการปรับขนาด) ในข้อมูล:")
print(X_train_scaled[:5])
print("บันทึก Sentiment:")
print(train_data['Sentiment'].head())

# บันทึก TF-IDF features และค่าที่ได้ลงในไฟล์ CSV
save_to_csv(X_train_scaled, train_data['Sentiment'], train_output_path)
save_to_csv(X_test_scaled, test_data['Sentiment'], test_output_path)
```

อธิบายเพิ่มเติม

- ไฟล์ CSV อาจมีอักขระที่ไม่ได้เข้ารหัสเป็น UTF-8 ซึ่งเป็น encoding เริ่มต้นของ pandas.read_csv การใช้ encoding='latin1' ช่วยหลีกเลี่ยงปัญหานี้
- การใช้ encoding ที่ไม่ถูกต้องอาจทำให้เกิดข้อผิดพลาด UnicodeDecodeError ซึ่งการใช้ latin1 สามารถช่วยหลีกเลี่ยงปัญหานี้ได้
- มีการเพิ่ม Feature MinMax แล้ว Scop ให้อยู่ในช่วง 0-1 เพื่อไม่ให้ data มีค่าเป็นลบ

Result

```
ค่าที่พบมากที่สุด 10 อันดับ:
[('the', 44708), ('to', 38328), ('and', 23976), ('of', 21509), ('a', 19326), ('in', 19121), ('for', 14033), ('is', 12241), ('are', 11335), ('covid', 10426)]
ข้อความตัวอย่าง:
0                                          and and
1    advice talk to your neighbours family to excha...
2    coronavirus australia woolworths to give elder...
3    my food stock is not the only one which is emp...
4    me ready to go at supermarket during the outbr...
Name: OriginalTweet, dtype: object
```


```
1    advice talk to your neighbours family to excha...
2    coronavirus australia woolworths to give elder...
3    my food stock is not the only one which is emp...
4    me ready to go at supermarket during the outbr...
Name: OriginalTweet, dtype: object
```

```
1    advice talk to your neighbours family to excha...
2    coronavirus australia woolworths to give elder...
3    my food stock is not the only one which is emp...
4    me ready to go at supermarket during the outbr...
1    advice talk to your neighbours family to excha...
2    coronavirus australia woolworths to give elder...
3    my food stock is not the only one which is emp...
4    me ready to go at supermarket during the outbr...
1    advice talk to your neighbours family to excha...
2    coronavirus australia woolworths to give elder...
1    advice talk to your neighbours family to excha...
2    coronavirus australia woolworths to give elder...
1    advice talk to your neighbours family to excha...
2    coronavirus australia woolworths to give elder...
3    my food stock is not the only one which is emp...
1    advice talk to your neighbours family to excha...
2    coronavirus australia woolworths to give elder...
1    advice talk to your neighbours family to excha...
1    advice talk to your neighbours family to excha...
2    coronavirus australia woolworths to give elder...
1    advice talk to your neighbours family to excha...
2    coronavirus australia woolworths to give elder...
3    my food stock is not the only one which is emp...
3    my food stock is not the only one which is emp...
4    me ready to go at supermarket during the outbr...
Name: OriginalTweet, dtype: object
```

```
จำนวนฟีเจอร์ที่ถูกสร้างขึ้น: 1000
TF-IDF features ตัวอย่าง (หลังจากทำ PCA และการปรับขนาด) ในข้อมูล:
[[0.22079328 0.41231168 0.19476408 0.31502716 0.37344406 0.29284488
  0.24150958 0.57617493 0.54816629 0.43134665 0.17363326 0.56342281
  0.31707355 0.34142744 0.91685689 1.         0.65416114 0.1869991
  1.         0.54478328 0.50823498 0.52821538 0.37273319 0.50005963
  0.43443637 0.27119677 0.3723108  0.28949967 0.68088854 0.35535673
  0.4579903  0.36976935 0.38700104 0.55490245 0.51449351 0.58057953
  0.19310351 0.27659623 0.45512692 0.19576599 0.42609926 0.42315708
  0.32686599 0.34266912 0.59762009 0.59828296 0.45800438 0.47180308
  0.51011733 0.44437547 0.50000505 0.55399891 0.5061283  0.41015127
  0.363387   0.51993238 0.29392812 0.29178219 0.42328283 0.52427997
  0.42083689 0.44540377 0.4171807  0.44626998 0.43698816 0.57641441
  0.48277417 0.550313   0.48914272 0.52455911 0.59855963 0.47245925
  0.31368305 0.41148755 0.43117447 0.37828196 0.42371005 0.40936108
  0.44605428 0.43651879 0.48379518 0.6095743  0.56209044 0.5349985
  0.43093155 0.45837476 0.53508734 0.57013335 0.54169115 0.33795823
  0.58737792 0.36549284 0.55917652 0.57610832 0.39420099 0.49590217
```

```
บันทึก Sentiment:
0    0
1    1
2    1
3    1
4   -2
Name: Sentiment, dtype: int64
TF-IDF features และค่าที่ได้ถูกบันทึกใน C:\Users\Windows 11\Desktop\CoronaML\preprocessed_Corona_NLP_train.csv
TF-IDF features และค่าที่ได้ถูกบันทึกใน C:\Users\Windows 11\Desktop\CoronaML\preprocessed_Corona_NLP_test.csv
```

Coding model

```python
import pandas as pd
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import classification_report, accuracy_score

# ฟังก์ชันทดสอบโมเดล
Codeium: Refactor | Explain | Generate Docstring | ×
def test_model(model, X_train, y_train, X_test, y_test):
    model.fit(X_train, y_train)  # ฝึกโมเดลด้วยข้อมูล
    y_pred = model.predict(X_test)  # ทำนายผลด้วยข้อมูลทดสอบ
    print(classification_report(y_test, y_pred))  # แสดงรายงานการทำนาย
    print("ความแม่นยำ:", accuracy_score(y_test, y_pred))  # แสดงค่าความแม่นยำ

# โหลดข้อมูลฝึกและทดสอบที่ผ่านการ preprocessing แล้ว
train_file_path = 'C:\\Users\\Windows 11\\Desktop\\CoronaML\\preprocessed_Corona_NLP_train.csv'
test_file_path = 'C:\\Users\\Windows 11\\Desktop\\CoronaML\\preprocessed_Corona_NLP_test.csv'

train_data = pd.read_csv(train_file_path)
test_data = pd.read_csv(test_file_path)
```

```python
# แยกฟีเจอร์และค่าเป้าหมาย
X_train = train_data.drop('Sentiment', axis=1)
y_train = train_data['Sentiment']
X_test = test_data.drop('Sentiment', axis=1)
y_test = test_data['Sentiment']

# สร้างโมเดล Logistic Regression พร้อมกับ Grid Search
param_grid = {'C': [1, 10, 100, 1000, 3792]}
grid_search = GridSearchCV(LogisticRegression(max_iter=1000), param_grid, cv=5, scoring='accuracy')
grid_search.fit(X_train, y_train)

print("Best parameters:", grid_search.best_params_)

# ทดสอบโมเดลที่ดีที่สุด
best_model = grid_search.best_estimator_
test_model(best_model, X_train, y_train, X_test, y_test)
```

Result

```
Best parameters: {'C': 1}
              precision    recall  f1-score   support

          -2       0.45      0.31      0.37       592
          -1       0.41      0.34      0.37      1041
           0       0.45      0.59      0.51       619
           1       0.34      0.47      0.39       947
           2       0.48      0.29      0.36       599

    accuracy                           0.40      3798
   macro avg       0.42      0.40      0.40      3798
weighted avg       0.41      0.40      0.40      3798

ความแม่นยำ: 0.40179041600842547
```

อธิบายเพิ่มเติม

- มีการTrain model ด้วย วิธี LogisticRegression เนื่องจาก Logistic Regression สามารถจัดการกับข้อมูลที่มีความไม่สมดุลระหว่างคลาสได้ดี(Class weight)