EX. NO: 4

Text preprocessing and Analysis pipeline

AIM:

To clean and preprocess amazon reviews tools for analysis.

CODE:

```
import pandas as pd
import re
import spacy

nlp = spacy.load("en_core_web_sm")

df = pd.read_csv('amazon_reviews.csv')

print(df['reviews.text'].head())

def clean(text):
    if pd.isnull(text):
        return ()
    text = text.lower()
    text = result[re[^\w\s]::text)
    text = text.encode['ascii:,ignore,
                       'code')

    doc = nlp(p(text))

    if not text is-stop and not token
        is_punct:
    return tokens.
```

cleaned_token :

0 [got , gps, respond , otr , road ...

1 [m, professional , ofr , mek ...

2 [duck , dogs , prior , ...

3 [going, write , thought , ...

4 [we , gor, recks , got , Africa , ...

Top 15 frequent words in Amazon reviews :

[(1, 3203), ('root' , 1947), ('it' , 962),

('books' , 005), ('kindle', 501), , ...

(great , A22) , ('we', 120), ...] .

```
df['cleaned'] = df['review_text'].apply(
    lambda text: clean_text_spacy(text))

print(df[['review_text', 'cleaned']].head())

all_tokens = [token for tokens in df['cleaned_tokens']
              for token in tokens]

from collections import Counter
word_freq = Counter(all_tokens)
print('word_freq', word_freq)
```

RESULT:

Cleaned token contains meaningful words without notes and frequent words reflect from the review.