



```
In [6]: import pandas as pd
import nltk
import string
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from nltk import pos_tag, ne_chunk

nltk.download('punkt_tab')
nltk.download('punkt')
nltk.download('stopwords')
nltk.download('averaged_perceptron_tagger')
nltk.download('maxent_ne_chunker')
nltk.download('words')

df = pd.read_csv("Reviews.csv.zip")

reviews = df['Text']
reviews = reviews.dropna()
reviews = reviews[:10000]

print("Sample Review:\n", reviews.iloc[0])
```

```
[nltk_data] Downloading package punkt_tab to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt_tab.zip.
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]   /root/nltk_data...
[nltk_data]   Package averaged_perceptron_tagger is already up-to-
[nltk_data]   date!
[nltk_data] Downloading package maxent_ne_chunker to
[nltk_data]   /root/nltk_data...
[nltk_data]   Package maxent_ne_chunker is already up-to-date!
[nltk_data] Downloading package words to /root/nltk_data...
[nltk_data]   Package words is already up-to-date!
```

Sample Review:

I have bought several of the Vitality canned dog food products and have found them all to be of good quality. The product looks more like a stew than a processed meat and it smells better. My Labrador is finicky and she appreciates this product better than most.

```
In [4]: def preprocess(text):
        text = text.lower()
        text = text.translate(str.maketrans('', '', string.punctuation))
        return text

processed_reviews = reviews.apply(preprocess)
print("After Preprocessing:\n", processed_reviews.iloc[0])
```

After Preprocessing:

i have bought several of the vitality canned dog food products and have found them all to be of good quality the product looks more like a stew than a processed meat and it smells better my labrador is finicky and she appreciates this product better than most

```
In [7]: stop_words = set(stopwords.words('english'))

def tokenize(text):
    tokens = word_tokenize(text)
    tokens = [w for w in tokens if w.isalpha()]
    tokens = [w for w in tokens if w not in stop_words]
    return tokens

tokens = processed_reviews.apply(tokenize)
print("Tokenized Sample:\n", tokens.iloc[0])
```

Tokenized Sample:

['bought', 'several', 'vitality', 'canned', 'dog', 'food', 'products', 'found', 'good', 'quality', 'product', 'looks', 'like', 'stew', 'processed', 'meat', 'smells', 'better', 'labrador', 'finicky', 'appreciates', 'product', 'better']

```
In [9]: nltk.download('averaged_perceptron_tagger_eng')
sample_tokens = tokens.iloc[0]
pos_tags = pos_tag(sample_tokens)
print("POS Tags:\n", pos_tags)
```

```
[nltk_data] Downloading package averaged_perceptron_tagger_eng to
[nltk_data] /root/nltk_data...
[nltk_data] Unzipping taggers/averaged_perceptron_tagger_eng.zip.
```

POS Tags:

[('bought', 'VBD'), ('several', 'JJ'), ('vitality', 'NN'), ('canned', 'VBD'), ('dog', 'JJ'), ('food', 'NN'), ('products', 'NNS'), ('found', 'VBD'), ('good', 'JJ'), ('quality', 'NN'), ('product', 'NN'), ('looks', 'VBZ'), ('like', 'IN'), ('stew', 'NN'), ('processed', 'VBN'), ('meat', 'NN'), ('smells', 'NNS'), ('better', 'RBR'), ('labrador', 'NN'), ('finicky', 'JJ'), ('appreciates', 'VBZ'), ('product', 'NN'), ('better', 'RBR')]

```
In [11]: nltk.download('maxent_ne_chunker_tab')
ner_tree = ne_chunk(pos_tags)
print("Named Entity Recognition (NER):\n", ner_tree)
```

```
[nltk_data] Downloading package maxent_ne_chunker_tab to
[nltk_data] /root/nltk_data...
[nltk_data] Unzipping chunkers/maxent_ne_chunker_tab.zip.
```

Named Entity Recognition (NER):

(S

bought/VBD

several/JJ

vitality/NN

canned/VBD

dog/JJ

food/NN

products/NNS

found/VBD

good/JJ

quality/NN

product/NN

looks/VBZ

like/IN

stew/NN

processed/VBN

meat/NN

smells/NNS

better/RBR

labrador/NN

finicky/JJ

appreciates/VBZ

product/NN

better/RBR)