EXP NO: 5a	**INFORMATION EXTRACTION USING NLTK**
DATE: 4/9/25	

Aim:

To extract meaningful information such as named entities and grammatical structures from text data using NLTK’s natural language processing tools.

Program:

Step 1: Import Required Libraries

```
import pandas as pd
import nltk import
string import re
from nltk.corpus import stopwords from
nltk.tokenize import word_tokenize Step
```

2: Download and Load the Dataset

```
import kagglehub import
os

# Download dataset
path = kagglehub.dataset_download("snap/amazon-fine-food-reviews")

# Load the CSV file
df = pd.read_csv(os.path.join(path, "Reviews.csv"))

# Select only the review text column and limit to 1000 reviews
reviews  =  df['Text'].dropna()[:1000] Step 3: Download NLTK
```

Resources

```
nltk.download('punkt') nltk.download('stopwords')
nltk.download('averaged_perceptron_tagger')
nltk.download('maxent_ne_chunker')
```

nltk.download('words') **Step 4: Preprocess Text**

```
def preprocess(text):     text = text.strip().lower()                # lowercase
+ remove spaces     text = re.sub(r'[^a-z\s]', '', text)       # remove
punctuation/special chars     return text
 reviews_cleaned = reviews.apply(preprocess)
```

Step 5: Tokenization and Stopword Removal

```
stop_words  =  set(stopwords.words("english"))
def                     tokenize(text):tokens=
word_tokenize(text)      #      split      into
wordstokens=[wforwin
tokens  if  w.isalpha()]  #  keep  only  words
tokens = [w for w in tokens if w.lower() not
in stop_words] return tokens


tokens_sample = tokenize(reviews_cleaned.iloc[50])
print(tokens_sample[:20])
```

Step 6: Part-of-Speech (POS) Tagging

```
pos_tags = nltk.pos_tag(tokens_sample)
print(pos_tags[:15])
```

Each word is labeled with its grammatical role — noun, verb, adjective, etc.

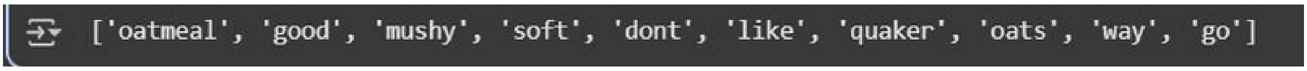
Example:

`[('oatmeal', 'NN'), ('good', 'JJ'), ('soft', 'JJ')]` **Step**

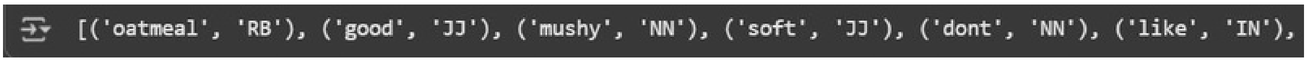
7: Named Entity Recognition (NER)

```
ner_tree = nltk.ne_chunk(pos_tags, binary=False)
print(ner_tree)
```

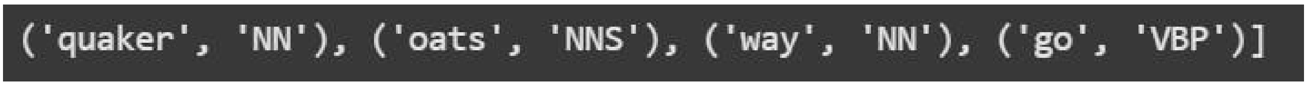
Output:



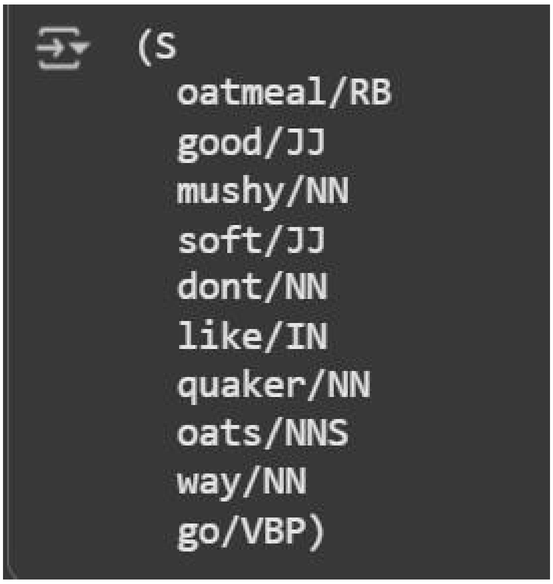
```
['oatmeal', 'good', 'mushy', 'soft', 'dont', 'like', 'quaker', 'oats', 'way', 'go']
```



```
[('oatmeal', 'RB'), ('good', 'JJ'), ('mushy', 'NN'), ('soft', 'JJ'), ('dont', 'NN'), ('like', 'IN'),
```



```
('quaker', 'NN'), ('oats', 'NNS'), ('way', 'NN'), ('go', 'VBP')]
```



```
(S
  oatmeal/RB
  good/JJ
  mushy/NN
  soft/JJ
  dont/NN
  like/IN
  quaker/NN
  oats/NNS
  way/NN
  go/VBP)
```

Result:

The text was successfully tokenized, POS-tagged, and processed for named entity recognition using NLTK, enabling structured extraction of linguistic and semantic information.