In [20]:
```
import re
import nltk
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
import string

nltk.download('punkt_tab')
nltk.download('stopwords')
```

```
[nltk_data] Downloading package punkt_tab to
[nltk_data]     C:\Users\vdm\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt_tab is already up-to-date!
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\vdm\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

Out[20]: True

In [21]:
```
# task1: data exploration
import pandas as pd
# from google.colab import drive
# from google.colab import files
# files=files.upload()

# I extended data to 100 rows using chatgpt as only 7 rows are in old text_class
# df = pd.read_csv("/content/text_class_extended.csv") #in google colab
df = pd.read_csv("text_class_extended.csv")
df.head(25)
```

Out[21]:

| | text | label |
|---|---|---|
| **0** | It arrived late and in bad condition. | negative |
| **1** | It arrived late and in bad condition. | negative |
| **2** | Terrible service, I will never shop here again. | negative |
| **3** | Product was damaged when it arrived, very disa... | negative |
| **4** | The item broke within a week of use. | negative |
| **5** | Service was standard, nothing to highlight. | neutral |
| **6** | Excellent value for money, I'm impressed. | positive |
| **7** | Impressive performance and great service. | positive |
| **8** | The quality is good, but the delivery was late. | neutral |
| **9** | Absolutely wonderful experience, highly recomm... | positive |
| **10** | Delivery time was average, no major issues. | neutral |
| **11** | Very happy with the purchase, will buy again. | positive |
| **12** | The packaging was great and product exceeded e... | positive |
| **13** | Absolutely wonderful experience, highly recomm... | positive |
| **14** | Neither good nor bad, just acceptable. | neutral |
| **15** | The product works flawlessly, highly satisfied. | positive |
| **16** | Product was damaged when it arrived, very disa... | negative |
| **17** | Neither good nor bad, just acceptable. | neutral |
| **18** | It arrived late and in bad condition. | negative |
| **19** | Delivery time was average, no major issues. | neutral |
| **20** | The product matches the description, no surpri... | neutral |
| **21** | Poor quality and unresponsive customer support. | negative |
| **22** | Great experience overall, five stars. | positive |
| **23** | The packaging was great and product exceeded e... | positive |
| **24** | Top-notch quality and fast delivery. | positive |

In [22]:
```
df.describe()
```

Out[22]:

| | text | label |
|---|---|---|
| **count** | 100 | 100 |
| **unique** | 28 | 3 |
| **top** | It arrived late and in bad condition. | positive |
| **freq** | 9 | 34 |

In [23]:
```
#total rows
print(f"total rows: {len(df)}")

# count of labels
print("\n", df['label'].value_counts())
```

```
total rows: 100

 label
positive    34
negative    33
neutral     33
Name: count, dtype: int64
```

In [24]:
```
# all stopwords from library
stop_words = set(stopwords.words('english'))
print(stop_words)

def clean_text(text):

    text = text.lower()

    # removin g punctuation
    text = text.translate(str.maketrans('', '', string.punctuation))

    return text
```

```
{'only', 'by', "she'd", "needn't", 'too', "we've", 'm', 'how', 'what', 'other',
'for', 'shouldn', 'just', 'no', "he'd", 'your', 'their', 'than', "we'll", 'won',
'those', 'are', "couldn't", "should've", "shouldn't", "aren't", 'themselves', 'th
ere', 'when', 'you', 'then', "weren't", "i'd", "it'd", 'll', 'nor', "they'd", 'u
p', "doesn't", 'wasn', 'y', 'that', "won't", 'before', "it'll", 'why', 'after',
'in', 'and', 'same', 'more', 'ours', "mustn't", "i'll", 'himself', "shan't", "mig
htn't", 'wouldn', 'has', 'while', 'below', 'few', 's', 'to', "wasn't", 'we', 'm
e', 'theirs', 'being', "he'll", 'if', 'they', 'any', 'at', "i've", 'ain', 'its',
'about', 'she', 'do', "she'll", 'be', 'is', 'have', 'most', "you'll", 'should',
"we're", "hasn't", 'here', 'myself', 'shan', "he's", 'his', 'or', "they've", 'o
f', 'this', 'had', 'under', "she's", 'needn', "isn't", 'o', 'mightn', 'as', 'doe
s', 'd', 'very', 'which', "they're", 'because', 'both', 'through', 'hasn', "tha
t'll", 'i', "you've", 'it', 'into', "i'm", 'further', "you're", 'am', 'between',
'isn', 'her', 'yourselves', 'such', "you'd", 'so', 'itself', 'them', 'doing', 'ab
ove', 'weren', 'not', 've', 'where', 'who', 'once', 'hers', 'down', 'each', 'my',
'been', 'on', 'haven', "wouldn't", 'don', 'out', "we'd", 'can', 'from', 'did', 'h
im', 'but', 'against', 'mustn', 'the', 'these', "didn't", 'herself', "don't", 'ha
ving', 'ma', 'with', "it's", 'couldn', 'own', 'were', 'didn', 'now', 'again', 'r
e', 'during', 'yours', 'aren', 'ourselves', 'was', 'yourself', 'until', 'our', 'a
n', "hadn't", 'hadn', 'he', 'a', "they'll", 'will', 't', 'all', 'off', 'over', 's
ome', "haven't", 'whom', 'doesn'}
```

In [25]:
```
# now applying preprocessing
df['cleaned_step_i'] = df['text'].apply(clean_text)
#cleaned rows
df[['text', 'cleaned_step_i']].head()
```

Out[25]:

| | text | cleaned_step_i |
|---|---|---|
| 0 | It arrived late and in bad condition. | it arrived late and in bad condition |
| 1 | It arrived late and in bad condition. | it arrived late and in bad condition |
| 2 | Terrible service, I will never shop here again. | terrible service i will never shop here again |
| 3 | Product was damaged when it arrived, very disa... | product was damaged when it arrived very disap... |
| 4 | The item broke within a week of use. | the item broke within a week of use |

In [26]:
```
df['cleaned_step_ii'] = [word_tokenize(text) for text in df['cleaned_step_i']]
df[['cleaned_step_i', 'cleaned_step_ii']].head()
```

Out[26]:

| | cleaned_step_i | cleaned_step_ii |
|---|---|---|
| 0 | it arrived late and in bad condition | [it, arrived, late, and, in, bad, condition] |
| 1 | it arrived late and in bad condition | [it, arrived, late, and, in, bad, condition] |
| 2 | terrible service i will never shop here again | [terrible, service, i, will, never, shop, here... |
| 3 | product was damaged when it arrived very disap... | [product, was, damaged, when, it, arrived, ver... |
| 4 | the item broke within a week of use | [the, item, broke, within, a, week, of, use] |

In [27]:
```
df['cleaned_step_iii'] = [[word for word in text if word not in stop_words] for
df[['cleaned_step_ii', 'cleaned_step_iii']].head()
```

Out[27]:

| | cleaned_step_ii | cleaned_step_iii |
|---|---|---|
| 0 | [it, arrived, late, and, in, bad, condition] | [arrived, late, bad, condition] |
| 1 | [it, arrived, late, and, in, bad, condition] | [arrived, late, bad, condition] |
| 2 | [terrible, service, i, will, never, shop, here... | [terrible, service, never, shop] |
| 3 | [product, was, damaged, when, it, arrived, ver... | [product, damaged, arrived, disappointed] |
| 4 | [the, item, broke, within, a, week, of, use] | [item, broke, within, week, use] |

In [28]:
```
df['final_cleaned_text'] = [' '.join(text) for text in df['cleaned_step_iii']]
df[['cleaned_step_iii', 'final_cleaned_text']].head()
```

Out[28]:

| | cleaned_step_iii | final_cleaned_text |
|---|---|---|
| 0 | [arrived, late, bad, condition] | arrived late bad condition |
| 1 | [arrived, late, bad, condition] | arrived late bad condition |
| 2 | [terrible, service, never, shop] | terrible service never shop |
| 3 | [product, damaged, arrived, disappointed] | product damaged arrived disappointed |
| 4 | [item, broke, within, week, use] | item broke within week use |

In [29]:
```
# spliting the data between test and train sets
from sklearn.model_selection import train_test_split

X_input_data = df['final_cleaned_text']

y_input_label = df['label']
print(df[['final_cleaned_text','label']])

X_train, X_test, y_train, y_test = train_test_split(X_input_data, y_input_label,
print(f"\n\nX_train: {X_train}")
print(f"\n\nX_test: {X_test}")
print(f"\n\ny_train: {y_train}")
print(f"\n\ny_test: {y_test}")
```

```
                     final_cleaned_text     label
0             arrived late bad condition  negative
1             arrived late bad condition  negative
2            terrible service never shop  negative
3   product damaged arrived disappointed  negative
4             item broke within week use  negative
..                                   ...       ...
95  impressive performance great service  positive
96                 loved product amazing  positive
97            arrived late bad condition  negative
98            arrived late bad condition  negative
99  product damaged arrived disappointed  negative

[100 rows x 2 columns]


X_train: 55                 customer support helpful polite
88                    worst purchase ive ever made
26            impressive performance great service
42                              happy purchase buy
69            impressive performance great service
                          ...
60    received defective item complete waste money
71           product matches description surprises
14                     neither good bad acceptable
92                      arrived late bad condition
51                     neither good bad acceptable
Name: final_cleaned_text, Length: 80, dtype: object


X_test: 83             impressive performance great service
53                       quality good delivery late
70    packaging great product exceeded expectations
45     received defective item complete waste money
44               great experience overall five stars
39              product damaged arrived disappointed
22               great experience overall five stars
80                       color size completely wrong
10                delivery time average major issues
0                         arrived late bad condition
18                        arrived late bad condition
30                delivery time average major issues
73                     packaging fine could improved
33        product works flawlessly highly satisfied
90       poor quality unresponsive customer support
4                         item broke within week use
76                       neither good bad acceptable
77                       quality good delivery late
12    packaging great product exceeded expectations
31                service standard nothing highlight
Name: final_cleaned_text, dtype: object


y_train: 55    positive
88    negative
26    positive
42    positive
69    positive
        ...
60    negative
```

```
71     neutral
14     neutral
92    negative
51     neutral
Name: label, Length: 80, dtype: object


y_test: 83    positive
53     neutral
70    positive
45    negative
44    positive
39    negative
22    positive
80    negative
10     neutral
0     negative
18    negative
30     neutral
73     neutral
33    positive
90    negative
4     negative
76     neutral
77     neutral
12    positive
31     neutral
Name: label, dtype: object
```

In [30]:
```python
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.linear_model import LogisticRegression

#vectorizing
vectorizer = CountVectorizer()

X_train_vec = vectorizer.fit_transform(X_train)
# print(f"X_train_vec: {X_train_vec}")

X_test_vec = vectorizer.transform(X_test)
# print(f"X_test_vec: {X_test_vec}")
#training model
model = LogisticRegression()

model.fit(X_train_vec, y_train)
```

Out[30]:
```
LogisticRegression()
```

In [31]:
```python
from sklearn.metrics import accuracy_score

y_pred = model.predict(X_test_vec)

acc = accuracy_score(y_test, y_pred)

print("accuracy:", acc)

# increasing rows in the dataset improved accuracy to 0.95 from 0.5 previously
```

```
accuracy: 0.95
```

In [32]:
```
from sklearn.metrics import confusion_matrix, classification_report

print("confusion-matrix:")
print(confusion_matrix(y_test, y_pred))

print("\n\nreport:")
print(classification_report(y_test, y_pred))
```

```
confusion-matrix:
[[7 0 0]
 [0 6 1]
 [0 0 6]]


report:
              precision    recall  f1-score   support

    negative       1.00      1.00      1.00         7
     neutral       1.00      0.86      0.92         7
    positive       0.86      1.00      0.92         6

    accuracy                           0.95        20
   macro avg       0.95      0.95      0.95        20
weighted avg       0.96      0.95      0.95        20
```

In [ ]:
```
```