In [21]:

```
import pandas as pd
import numpy as np
import nltk
nltk.download('wordnet')
nltk.download('stopwords')
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer
import re
from bs4 import BeautifulSoup
```

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

Read Data

In [3]:

```
tsv_file = 'amazon_reviews_us_Office_Products_v1_00.tsv.gz'
df_full = pd.read_csv(tsv_file,compression='gzip',sep='\t',on_bad_lines='warn')
```

```
C:\Users\ravin\AppData\Local\Temp\ipykernel_30772\3481158619.py:2: ParserWarning: Skipping line 20773: expected
15 fields, saw 22
Skipping line 39834: expected 15 fields, saw 22
Skipping line 52957: expected 15 fields, saw 22
Skipping line 54540: expected 15 fields, saw 22

  df_full = pd.read_csv(tsv_file,compression='gzip',sep='\t',on_bad_lines='warn')
C:\Users\ravin\AppData\Local\Temp\ipykernel_30772\3481158619.py:2: ParserWarning: Skipping line 80276: expected
15 fields, saw 22
Skipping line 96168: expected 15 fields, saw 22
Skipping line 96866: expected 15 fields, saw 22
Skipping line 98175: expected 15 fields, saw 22
Skipping line 112539: expected 15 fields, saw 22
Skipping line 119377: expected 15 fields, saw 22
Skipping line 120065: expected 15 fields, saw 22
Skipping line 124703: expected 15 fields, saw 22

  df_full = pd.read_csv(tsv_file,compression='gzip',sep='\t',on_bad_lines='warn')
C:\Users\ravin\AppData\Local\Temp\ipykernel_30772\3481158619.py:2: ParserWarning: Skipping line 134024: expecte
d 15 fields, saw 22
Skipping line 153938: expected 15 fields, saw 22
Skipping line 156225: expected 15 fields, saw 22
Skipping line 168603: expected 15 fields, saw 22
Skipping line 187002: expected 15 fields, saw 22

  df_full = pd.read_csv(tsv_file,compression='gzip',sep='\t',on_bad_lines='warn')
C:\Users\ravin\AppData\Local\Temp\ipykernel_30772\3481158619.py:2: ParserWarning: Skipping line 200397: expecte
d 15 fields, saw 22
Skipping line 203809: expected 15 fields, saw 22
Skipping line 207680: expected 15 fields, saw 22
Skipping line 223421: expected 15 fields, saw 22
Skipping line 244032: expected 15 fields, saw 22

  df_full = pd.read_csv(tsv_file,compression='gzip',sep='\t',on_bad_lines='warn')
C:\Users\ravin\AppData\Local\Temp\ipykernel_30772\3481158619.py:2: ParserWarning: Skipping line 270329: expecte
d 15 fields, saw 22
Skipping line 276484: expected 15 fields, saw 22
Skipping line 304755: expected 15 fields, saw 22

  df_full = pd.read_csv(tsv_file,compression='gzip',sep='\t',on_bad_lines='warn')
C:\Users\ravin\AppData\Local\Temp\ipykernel_30772\3481158619.py:2: ParserWarning: Skipping line 379449: expecte
d 15 fields, saw 22
Skipping line 386191: expected 15 fields, saw 22
Skipping line 391811: expected 15 fields, saw 22

  df_full = pd.read_csv(tsv_file,compression='gzip',sep='\t',on_bad_lines='warn')
C:\Users\ravin\AppData\Local\Temp\ipykernel_30772\3481158619.py:2: ParserWarning: Skipping line 414348: expecte
d 15 fields, saw 22
Skipping line 414773: expected 15 fields, saw 22
Skipping line 417572: expected 15 fields, saw 22
Skipping line 419496: expected 15 fields, saw 22
Skipping line 430528: expected 15 fields, saw 22
Skipping line 442230: expected 15 fields, saw 22
Skipping line 450931: expected 15 fields, saw 22

  df_full = pd.read_csv(tsv_file,compression='gzip',sep='\t',on_bad_lines='warn')
C:\Users\ravin\AppData\Local\Temp\ipykernel_30772\3481158619.py:2: ParserWarning: Skipping line 465377: expecte
d 15 fields, saw 22
Skipping line 467685: expected 15 fields, saw 22
Skipping line 485055: expected 15 fields, saw 22
Skipping line 487220: expected 15 fields, saw 22
Skipping line 496076: expected 15 fields, saw 22
Skipping line 512269: expected 15 fields, saw 22

  df_full = pd.read_csv(tsv_file,compression='gzip',sep='\t',on_bad_lines='warn')
C:\Users\ravin\AppData\Local\Temp\ipykernel_30772\3481158619.py:2: ParserWarning: Skipping line 529505: expecte
d 15 fields, saw 22
Skipping line 531286: expected 15 fields, saw 22
Skipping line 535424: expected 15 fields, saw 22
Skipping line 569898: expected 15 fields, saw 22
Skipping line 586293: expected 15 fields, saw 22

  df_full = pd.read_csv(tsv_file,compression='gzip',sep='\t',on_bad_lines='warn')
C:\Users\ravin\AppData\Local\Temp\ipykernel_30772\3481158619.py:2: ParserWarning: Skipping line 593880: expecte
d 15 fields, saw 22
Skipping line 599274: expected 15 fields, saw 22
Skipping line 607961: expected 15 fields, saw 22
Skipping line 612413: expected 15 fields, saw 22
Skipping line 615913: expected 15 fields, saw 22

  df_full = pd.read_csv(tsv_file,compression='gzip',sep='\t',on_bad_lines='warn')
C:\Users\ravin\AppData\Local\Temp\ipykernel_30772\3481158619.py:2: ParserWarning: Skipping line 677580: expecte
d 15 fields, saw 22
Skipping line 687191: expected 15 fields, saw 22
Skipping line 710819: expected 15 fields, saw 22

  df_full = pd.read_csv(tsv_file,compression='gzip',sep='\t',on_bad_lines='warn')
C:\Users\ravin\AppData\Local\Temp\ipykernel_30772\3481158619.py:2: ParserWarning: Skipping line 728692: expecte
d 15 fields, saw 22
Skipping line 730216: expected 15 fields, saw 22
Skipping line 758397: expected 15 fields, saw 22
Skipping line 760061: expected 15 fields, saw 22
Skipping line 768935: expected 15 fields, saw 22
Skipping line 769483: expected 15 fields, saw 22

  df_full = pd.read_csv(tsv_file,compression='gzip',sep='\t',on_bad_lines='warn')
C:\Users\ravin\AppData\Local\Temp\ipykernel_30772\3481158619.py:2: ParserWarning: Skipping line 822725: expecte
d 15 fields, saw 22
Skipping line 823621: expected 15 fields, saw 22

  df_full = pd.read_csv(tsv_file,compression='gzip',sep='\t',on_bad_lines='warn')
C:\Users\ravin\AppData\Local\Temp\ipykernel_30772\3481158619.py:2: ParserWarning: Skipping line 857041: expecte
d 15 fields, saw 22
Skipping line 857320: expected 15 fields, saw 22
Skipping line 858565: expected 15 fields, saw 22
Skipping line 860629: expected 15 fields, saw 22
Skipping line 864033: expected 15 fields, saw 22
Skipping line 868673: expected 15 fields, saw 22
Skipping line 869189: expected 15 fields, saw 22

  df_full = pd.read_csv(tsv_file,compression='gzip',sep='\t',on_bad_lines='warn')
C:\Users\ravin\AppData\Local\Temp\ipykernel_30772\3481158619.py:2: ParserWarning: Skipping line 938605: expecte
d 15 fields, saw 22
Skipping line 940100: expected 15 fields, saw 22
Skipping line 975137: expected 15 fields, saw 22
Skipping line 976314: expected 15 fields, saw 22

  df_full = pd.read_csv(tsv_file,compression='gzip',sep='\t',on_bad_lines='warn')
C:\Users\ravin\AppData\Local\Temp\ipykernel_30772\3481158619.py:2: ParserWarning: Skipping line 985597: expecte
d 15 fields, saw 22
Skipping line 990873: expected 15 fields, saw 22
Skipping line 991806: expected 15 fields, saw 22
Skipping line 1019808: expected 15 fields, saw 22
Skipping line 1021526: expected 15 fields, saw 22
Skipping line 1023905: expected 15 fields, saw 22
Skipping line 1044207: expected 15 fields, saw 22

  df_full = pd.read_csv(tsv_file,compression='gzip',sep='\t',on_bad_lines='warn')
C:\Users\ravin\AppData\Local\Temp\ipykernel_30772\3481158619.py:2: ParserWarning: Skipping line 1084683: expect
ed 15 fields, saw 22
Skipping line 1093288: expected 15 fields, saw 22

  df_full = pd.read_csv(tsv_file,compression='gzip',sep='\t',on_bad_lines='warn')
C:\Users\ravin\AppData\Local\Temp\ipykernel_30772\3481158619.py:2: ParserWarning: Skipping line 1136430: expect
ed 15 fields, saw 22
Skipping line 1139815: expected 15 fields, saw 22

  df_full = pd.read_csv(tsv_file,compression='gzip',sep='\t',on_bad_lines='warn')
C:\Users\ravin\AppData\Local\Temp\ipykernel_30772\3481158619.py:2: ParserWarning: Skipping line 1179821: expect
ed 15 fields, saw 22
Skipping line 1195351: expected 15 fields, saw 22
Skipping line 1202007: expected 15 fields, saw 22
Skipping line 1224868: expected 15 fields, saw 22
Skipping line 1232490: expected 15 fields, saw 22
Skipping line 1238697: expected 15 fields, saw 22

  df_full = pd.read_csv(tsv_file,compression='gzip',sep='\t',on_bad_lines='warn')
C:\Users\ravin\AppData\Local\Temp\ipykernel_30772\3481158619.py:2: ParserWarning: Skipping line 1258654: expect
ed 15 fields, saw 22
Skipping line 1279948: expected 15 fields, saw 22
Skipping line 1294360: expected 15 fields, saw 22
Skipping line 1302240: expected 15 fields, saw 22

  df_full = pd.read_csv(tsv_file,compression='gzip',sep='\t',on_bad_lines='warn')
C:\Users\ravin\AppData\Local\Temp\ipykernel_30772\3481158619.py:2: ParserWarning: Skipping line 1413654: expect
ed 15 fields, saw 22

  df_full = pd.read_csv(tsv_file,compression='gzip',sep='\t',on_bad_lines='warn')
C:\Users\ravin\AppData\Local\Temp\ipykernel_30772\3481158619.py:2: ParserWarning: Skipping line 1687095: expect
ed 15 fields, saw 22

  df_full = pd.read_csv(tsv_file,compression='gzip',sep='\t',on_bad_lines='warn')
C:\Users\ravin\AppData\Local\Temp\ipykernel_30772\3481158619.py:2: ParserWarning: Skipping line 1805966: expect
ed 15 fields, saw 22

  df_full = pd.read_csv(tsv_file,compression='gzip',sep='\t',on_bad_lines='warn')
C:\Users\ravin\AppData\Local\Temp\ipykernel_30772\3481158619.py:2: ParserWarning: Skipping line 1892134: expect
ed 15 fields, saw 22

  df_full = pd.read_csv(tsv_file,compression='gzip',sep='\t',on_bad_lines='warn')
C:\Users\ravin\AppData\Local\Temp\ipykernel_30772\3481158619.py:2: DtypeWarning: Columns (7) have mixed types.
Specify dtype option on import or set low_memory=False.
  df_full = pd.read_csv(tsv_file,compression='gzip',sep='\t',on_bad_lines='warn')
```

Keep Reviews and Ratings

In [4]:

```
df = df_full[['review_body','star_rating']].copy()
df.rename(columns={'review_body': 'Review', 'star_rating': 'Rating'}, inplace=True)
print(df.head())
print(df.shape)
df['Rating'] = pd.to_numeric(df['Rating'], errors='coerce')
```

```
                                              Review Rating
0                                     Great product.      5
1  What's to say about this commodity item except...      5
2    Haven't used yet, but I am sure I will like it.      5
3  Although this was labeled as &#34;new&#34; the...      1
4                     Gorgeous colors and easy to use      4
(2640254, 2)
```

We form three classes and select 20000 reviews randomly from each class.

In [5]:

```
count_negative = (df['Rating'] <= 2).sum()
count_neutral  = (df['Rating'] == 3).sum()
count_positive = (df['Rating'] > 3).sum()
print("Negative reviews:", count_negative)
print("Neutral reviews:", count_neutral)
print("Positive reviews:", count_positive)

df = df[df['Rating'] != 3]
print("\nData shape after discarding rating=3:", df.shape)
df['Sentiment'] = df['Rating'].apply(lambda x: 1 if x > 3 else 0)

df_neg = df[df['Sentiment'] == 0]
df_pos = df[df['Sentiment'] == 1]

df_neg_sample = df_neg.sample(n=20000, random_state=42)
df_pos_sample = df_pos.sample(n=20000, random_state=42)
df_downsized = pd.concat([df_neg_sample, df_pos_sample], ignore_index=True)
```

```
Negative reviews: 445363
Neutral reviews: 193691
Positive reviews: 2001183

Data shape after discarding rating=3: (2446563, 2)
```

Data Cleaning

In [ ]:

```
df_downsized.dropna(subset=['Review'], inplace=True)
df_downsized['Review'] = df_downsized['Review'].astype(str)

avg_length_before_cleaning = df_downsized['Review'].apply(len).mean()
print("Average length (in characters) before cleaning:", avg_length_before_cleaning)

df_downsized['Review'] = df_downsized['Review'].str.lower()

def remove_html_and_urls(text):
    # Remove HTML tags using BeautifulSoup
    text_no_html = BeautifulSoup(text, "html.parser").get_text(separator=" ")

    # Remove URLs using regex
    # This pattern matches http://, https://, or www. links
    text_no_url = re.sub(r'(https?://\S+|www\.\S+)', '', text_no_html)
    return text_no_url

df_downsized['Review'] = df_downsized['Review'].apply(remove_html_and_urls)

df_downsized['Review'] = df_downsized['Review'].str.replace('[^a-z]', ' ', regex=True)

df_downsized['Review'] = df_downsized['Review'].str.split().str.join(' ')

#did my best to add as much as possible
contractions_dict = {
    "won't": "will not",
    "can't": "cannot",
    "don't": "do not",
    "didn't": "did not",
    "i'm": "i am",
    "it's": "it is",
    "he's": "he is",
    "she's": "she is",
    "that's": "that is",
    "aren't": "are not",
    "weren't": "were not",
    "haven't": "have not",
    "hasn't": "has not",
    "shouldn't": "should not",
    "wouldn't": "would not",
    "couldn't": "could not",
    "isn't": "is not",
    "what's": "what is",
    "where's": "where is",
    "who's": "who is",
    "you'd": "you would",
    "you'll": "you will",
    "you're": "you are",
    "they're": "they are",
    "they've": "they have",
    "we're": "we are",
    "we've": "we have",
    "there's": "there is"
}

contractions_pattern = re.compile(r'\b(' + '|'.join(contractions_dict.keys()) + r')\b')

def expand_contractions(text, pattern=contractions_pattern):
    def replace(match):
        return contractions_dict[match.group(0)]
    return pattern.sub(replace, text)

df_downsized['Review'] = df_downsized['Review'].apply(expand_contractions)

avg_length_after_cleaning = df_downsized['Review'].apply(len).mean()
print("Average length (in characters) after cleaning:", avg_length_after_cleaning)
```

```
Average length (in characters) before cleaning: 317.63445672283615
C:\Users\ravin\AppData\Local\Temp\ipykernel_30772\2816853360.py:11: MarkupResemblesLocatorWarning: The input lo
oks more like a filename than markup. You may want to open this file and pass the filehandle into Beautiful Sou
p.
  text_no_html = BeautifulSoup(text, "html.parser").get_text(separator=" ")
Average length (in characters) after cleaning: 301.5544777238862
```

Pre-processing

In [7]:

```
stop_words = set(stopwords.words('english'))
lemmatizer = WordNetLemmatizer()

def preprocess_text(text):
    tokens = text.split()
    tokens = [word for word in tokens if word not in stop_words]
    tokens = [lemmatizer.lemmatize(word) for word in tokens]
    processed_text = " ".join(tokens)
    return processed_text

sample_indices = df_downsized.sample(3, random_state=42).index

print("SAMPLE REVIEWS BEFORE PREPROCESSING:")
for idx in sample_indices:
    print(f"Review {idx}:\n{df_downsized.loc[idx, 'Review']}")
    print("-"*80)

avg_length_before_preprocessing = df_downsized['Review'].apply(len).mean()
print("Average length (in characters) before preprocessing:", avg_length_before_preprocessing)

df_downsized['Review'] = df_downsized['Review'].apply(preprocess_text)

print("SAMPLE REVIEWS AFTER PREPROCESSING:")
for idx in sample_indices:
    print(f"Review {idx}:\n{df_downsized.loc[idx, 'Review']}")
    print("-"*80)

avg_length_after_preprocessing = df_downsized['Review'].apply(len).mean()
print("Average length (in characters) after preprocessing:", avg_length_after_preprocessing)

print("Average length (in characters) before preprocessing:", avg_length_before_preprocessing)
```

```
SAMPLE REVIEWS BEFORE PREPROCESSING:
Review 7516:
poor sound quality i exchanged it for the philips id which is a very nice phone system read my review for the p
hilips id
--------------------------------------------------------------------------------
Review 13706:
it burned immediately and i cant return it don t buy
--------------------------------------------------------------------------------
Review 21103:
very cute and unique
--------------------------------------------------------------------------------
Average length (in characters) before preprocessing: 301.5544777238862
SAMPLE REVIEWS AFTER PREPROCESSING:
Review 7516:
poor sound quality exchanged philip id nice phone system read review philip id
--------------------------------------------------------------------------------
Review 13706:
burned immediately cant return buy
--------------------------------------------------------------------------------
Review 21103:
cute unique
--------------------------------------------------------------------------------
Average length (in characters) after preprocessing: 186.31031551577578
Average length (in characters) before preprocessing: 301.5544777238862
```

TF-IDF Feature Extraction

In [16]:

```
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split
#init the vectorizer, used hyperparameters to increase the accuracy
vectorizer = TfidfVectorizer(
    ngram_range=(1, 2),
    min_df=5,
    max_df=0.8,
    sublinear_tf=True
)

X = vectorizer.fit_transform(df_downsized['Review'])
y = df_downsized['Sentiment'].values

X_train, X_test, y_train, y_test = train_test_split(
    X, y,
    test_size=0.2,
    random_state=42
)

print("X_train shape:", X_train.shape)
print("X_test shape :", X_test.shape)
print("y_train shape:", y_train.shape)
print("y_test shape :", y_test.shape)
```

```
X_train shape: (31998, 37291)
X_test shape : (8000, 37291)
y_train shape: (31998,)
y_test shape : (8000,)
```

Perceptron

In [17]:

```
from sklearn.linear_model import Perceptron
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score

perceptron_model = Perceptron()#init to train the model
perceptron_model.fit(X_train, y_train)

y_train_pred = perceptron_model.predict(X_train)
y_test_pred = perceptron_model.predict(X_test)

acc_train = accuracy_score(y_train, y_train_pred)
prec_train = precision_score(y_train, y_train_pred)
recall_train = recall_score(y_train, y_train_pred)
f1_train = f1_score(y_train, y_train_pred)

acc_test = accuracy_score(y_test, y_test_pred)
prec_test = precision_score(y_test, y_test_pred)
recall_test = recall_score(y_test, y_test_pred)
f1_test = f1_score(y_test, y_test_pred)
```

```
print(acc_train)
print(prec_train)
print(recall_train)
print(f1_train)
print(acc_test)
print(prec_test)
print(recall_test)
print(f1_test)
```

```
0.9910931933245828
0.9902107494700088
0.992004497470173
0.9911068118700659
0.860875
0.859714928732183
0.8616541353383459
0.8606834397296282
```

SVM

In [18]:

```
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score

svm_model = SVC(kernel='linear', random_state=42)#init SVM model

svm_model.fit(X_train, y_train)

y_train_pred = svm_model.predict(X_train)
y_test_pred = svm_model.predict(X_test)

acc_train = accuracy_score(y_train, y_train_pred)
prec_train = precision_score(y_train, y_train_pred)
recall_train = recall_score(y_train, y_train_pred)
f1_train = f1_score(y_train, y_train_pred)

acc_test = accuracy_score(y_test, y_test_pred)
prec_test = precision_score(y_test, y_test_pred)
recall_test = recall_score(y_test, y_test_pred)
f1_test = f1_score(y_test, y_test_pred)

print(acc_train)
print(prec_train)
print(recall_train)
print(f1_train)
print(acc_test)
print(prec_test)
print(recall_test)
print(f1_test)
```

```
0.9595912244515282
0.9611431436450238
0.9579611468548941
0.9595495072735805
0.893
0.8943130347257172
0.8907268170426065
0.89251632345555
```

Logistic Regression

In [19]:

```
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score

logreg_model = LogisticRegression(random_state=42) #init logistic regression model

logreg_model.fit(X_train, y_train)

y_train_pred = logreg_model.predict(X_train)
y_test_pred = logreg_model.predict(X_test)

acc_train = accuracy_score(y_train, y_train_pred)
prec_train = precision_score(y_train, y_train_pred)
recall_train = recall_score(y_train, y_train_pred)
f1_train = f1_score(y_train, y_train_pred)

acc_test = accuracy_score(y_test, y_test_pred)
prec_test = precision_score(y_test, y_test_pred)
recall_test = recall_score(y_test, y_test_pred)
f1_test = f1_score(y_test, y_test_pred)

print(acc_train)
print(prec_train)
print(recall_train)
print(f1_train)
print(acc_test)
print(prec_test)
print(recall_test)
print(f1_test)
```

```
0.9285267829239328
0.9343504684730312
0.9219189206071585
0.9280930671278101
0.884625
0.8887198986058301
0.8786967418546366
0.8836798991808443
```

Naive Bayes

In [20]:

```
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score

nb_model = MultinomialNB() #naive bayes model
nb_model.fit(X_train, y_train)

y_train_pred = nb_model.predict(X_train)

acc_train = accuracy_score(y_train, y_train_pred)
prec_train = precision_score(y_train, y_train_pred)
recall_train = recall_score(y_train, y_train_pred)
f1_train = f1_score(y_train, y_train_pred)

acc_test = accuracy_score(y_test, y_test_pred)
prec_test = precision_score(y_test, y_test_pred)
recall_test = recall_score(y_test, y_test_pred)
f1_test = f1_score(y_test, y_test_pred)

#training metrics
print(acc_train)
print(prec_train)
print(recall_train)
print(f1_train)

#testing metrics
print(acc_test)
print(prec_test)
print(recall_test)
print(f1_test)
```

```
0.9127757984874054
0.9079126033822985
0.9188581422949591
0.9133525814162864
0.884625
0.8887198986058301
0.8786967418546366
0.8836798991808443
```