CSCI 544 - NLP Assignment 1:

Name: Nilakshi Nagrale
USC ID: 2403347301

Import necessary libraries and packages.

1. <u>Dataset Preparation:</u>

Using pandas

- Download the dataset and load it into a Pandas DataFrame.
- Keep only the "review_body" and "star_rating" columns.
- Handle data inconsistency, remove NaN values
- Filter ratings and map sentiment
  Ratings > 3 → Positive (`1`)
  Ratings ≤ 2 → Negative (`0`)
  Drop ratings = 3.
- Randomly sample 100,000 positive and 100,000 negative reviews.
- Then, do an 80-20 split for training and testing.

2. <u>Data Cleaning:</u>

Using regex expressions to match and replace the below items with empty strings:

- change all to lower case - string methods
- URLs - using BeautifulSoup parser
- Emails - using Regex
- HTML tags - using Regex
- Punctuations - using Regex
- extra spaces - using Regex
- special / non-alphabetical characters - using Regex

Output Avg length before/after data cleaning.

3. <u>Data Preprocessing:</u>

- Remove stop words - using nltk.corpus and stopwords
- Handle negative words
- Perfrom lemmatization - using nltk.stem and WordNetLemmatizer
- Extract features using TfidfVectorizer

Output Avg length before/after data processing.

<u>4. Perceptron Model:</u>

- Use Perceptron() and GridSearchCV() from sklearn library
- Perform hyperparameter tuning:
  max_iter - shows number of epochs

alpha - intensity of regualarization in case of penalty
penalty - controls model's penalty in case of larger weights

- Train model on train dataset
- Run model on test data
- Output Train/test metrics

5. SVM Model:

- Use LinearSVC() and GridSearchCV() from sklearn library
- Perform hyperparameter tuning:
  max_iter - shows number of epochs
  C - Regularization intensity, to help balance overfitting/underfitting
  loss - loss functions
- Train model on train dataset
- Run model on test data
- Output Train/test metrics

6. Logistic Regression Model:

- Use LogisticRegression() from sklearn library
- Train model on train dataset
- Run model on test data
- Output Train/test metrics

7. Naive Bayes Model:

- Use MultinomialNB() and GridSearchCV() from sklearn library
- Perform hyperparameter tuning:
  Alpha parameter
- Train model on train dataset
- Run model on test data
- Output Train/test metrics

PROGRAM OUTPUT:

```
Positive reviews: 2001052
Negative reviews: 445348
Neutral reviews (discarded): 193680

Average length before cleaning: 318.0072
Average length after cleaning: 301.1237
Average length before cleaning + processing: 318.0072
Average length after cleaning + processing: 194.0846

Perceptron - Training Data Accuracy: 0.8523
Perceptron - Training Data Precision: 0.8615
Perceptron - Training Data Recall: 0.8391
Perceptron - Training Data F1-Score: 0.8501
Perceptron - Testing Data Accuracy: 0.8490
Perceptron - Testing Data Precision: 0.8585
Perceptron - Testing Data Recall: 0.8376
Perceptron - Testing Data F1-Score: 0.8479

LinearSVC - Training Data Accuracy: 0.9247
LinearSVC - Training Data Precision: 0.9278
LinearSVC - Training Data Recall: 0.9208
LinearSVC - Training Data F1-Score: 0.9243
LinearSVC - Testing Data Accuracy: 0.9124
LinearSVC - Testing Data Precision: 0.9144
LinearSVC - Testing Data Recall: 0.9109
LinearSVC - Testing Data F1-Score: 0.9126

Logistic Reg - Training Data Accuracy: 0.9212
Logistic Reg - Training Data Precision: 0.9236
Logistic Reg - Training Data Recall: 0.9182
Logistic Reg - Training Data F1-Score: 0.9209
Logistic Reg - Testing Data Accuracy: 0.9122
Logistic Reg - Testing Data Precision: 0.9134
Logistic Reg - Testing Data Recall: 0.9116
Logistic Reg - Testing Data F1-Score: 0.9125

NB - Training Data Accuracy: 0.8834
NB - Training Data Precision: 0.8923
NB - Training Data Recall: 0.8718
NB - Training Data F1-Score: 0.8819
NB - Testing Data Accuracy: 0.8692
NB - Testing Data Precision: 0.8810
NB - Testing Data Recall: 0.8552
NB - Testing Data F1-Score: 0.8679
```