Lab Assignment 4- NLP Preprocessing And Text Classification

Group Members Name-

1) Vivek Borade
   Prn - 202201040216
2) Nirmal Chaturvedi
   Prn- 202201040210
3) Abhijeet Jadhav
   Prn- 202201040122

Batch : T4

Subject : Deep Learning LAB

Dataset- http://thinknook.com/wp-content/uploads/2012/09/Sentiment-Analysis-Dataset.zip

Colab Link- https://colab.research.google.com/drive/12_71qeO8DGVRcldKbi4i_FTHpffo978D?usp=sharing

Github Link-

https://github.com/nirmalchaturvedi/DL_A4/tree/main

```python
# Define the text preprocessing function
def preprocess_text(text):
    text = str(text).lower()  # Convert to lowercase
    tokens = word_tokenize(text)  # Tokenize the text
    tokens = [word for word in tokens if word.isalpha()]  # Remove punctuation/numbers
    tokens = [word for word in tokens if word not in stop_words]  # Remove stopwords
    tokens = [stemmer.stem(word) for word in tokens]  # Apply stemming
    tokens = [lemmatizer.lemmatize(word) for word in tokens]  # Apply lemmatization
    return ' '.join(tokens)  # Join tokens back into a string

# Apply preprocessing function to the SentimentText column
df['cleaned_text'] = df['SentimentText'].apply(preprocess_text)

# Display a sample of cleaned text
print("\n--- Cleaned Text Sample ---")
print(df[['SentimentText', 'cleaned_text']].head())

# ----------------------------
# Step 4: Vectorization
# ----------------------------
# CountVectorizer (Bag of Words)
count_vectorizer = CountVectorizer()
X_count = count_vectorizer.fit_transform(df['cleaned_text'])

# TF-IDF Vectorizer
tfidf_vectorizer = TfidfVectorizer()
X_tfidf = tfidf_vectorizer.fit_transform(df['cleaned_text'])

# Display vectorization completion and shapes
print("\n--- Vectorization Complete ---")
print(f"CountVectorizer shape: {X_count.shape}")
print(f"TF-IDF shape: {X_tfidf.shape}")
```

Lab Assignment 4- NLP Preprocessing And Text Classification

```
                                      SentimentText
0                       is so sad for my APL frie...
1                     I missed the New Moon trail...
2                            omg its already 7:30 :O
3             .. Omgaga. Im sooo  im gunna CRy. I'...
4           i think mi bf is cheating on me!!!   ...

--- Missing Values ---
ItemID             0
Sentiment          0
SentimentSource    0
SentimentText      0
dtype: int64

--- Label Distribution ---
Sentiment
1    790177
0    788435
Name: count, dtype: int64

--- Cleaned Text Sample ---
                                      SentimentText  \
0                       is so sad for my APL frie...
1                     I missed the New Moon trail...
2                            omg its already 7:30 :O
3             .. Omgaga. Im sooo  im gunna CRy. I'...
4           i think mi bf is cheating on me!!!   ...

                                        cleaned_text
0                                    sad apl friend
1                               miss new moon trailer
2                                         omg alreadi
3  omgaga im sooo im gunna cri dentist sinc supos...
4                                  think mi bf cheat

--- Vectorization Complete ---
CountVectorizer shape: (1578612, 459583)
TF-IDF shape: (1578612, 459583)
```

Lab Assignment 4- NLP Preprocessing And Text Classification

```
--- Dataset Preview ---
   ItemID  Sentiment SentimentSource  \
0       1          0    Sentiment140
1       2          0    Sentiment140
2       3          1    Sentiment140
3       4          0    Sentiment140
4       5          0    Sentiment140

                                       SentimentText
0                        is so sad for my APL frie...
1                      I missed the New Moon trail...
2                            omg its already 7:30 :O
3              .. Omgaga. Im sooo  im gunna CRy. I'...
4           i think mi bf is cheating on me!!!   ...

--- Missing Values ---
ItemID             0
Sentiment          0
SentimentSource    0
SentimentText      0
dtype: int64

--- Label Distribution ---
Sentiment
1    790177
0    788435
Name: count, dtype: int64
```

```python
import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer, CountVectorizer
from sklearn.model_selection import train_test_split



# Step 2: Text Vectorization
df['cleaned'] = df['SentimentText'].str.lower().str.replace(r'[^\w\s]', '', regex=True)

# Step 3: Vectorization

# TF-IDF Vectorization
tfidf_vectorizer = TfidfVectorizer()
X_tfidf = tfidf_vectorizer.fit_transform(df['cleaned'])

# Count Vectorizer
count_vectorizer = CountVectorizer()
X_count = count_vectorizer.fit_transform(df['cleaned'])

# Target variable (Sentiment): 1 = Positive, 0 = Negative
y = df['Sentiment']

# Train-test split (80% train, 20% test)
X_train_tfidf, X_test_tfidf, y_train, y_test = train_test_split(X_tfidf, y, test_size=0.2, random_state=42)
X_train_count, X_test_count, _, _ = train_test_split(X_count, y, test_size=0.2, random_state=42)

# Check shapes
print(f"TF-IDF Matrix Shape: {X_tfidf.shape}")
print(f"Count Vectorizer Matrix Shape: {X_count.shape}")
```

```
TF-IDF Matrix Shape: (1578612, 852392)
Count Vectorizer Matrix Shape: (1578612, 852392)
```

Lab Assignment 4- NLP Preprocessing And Text Classification

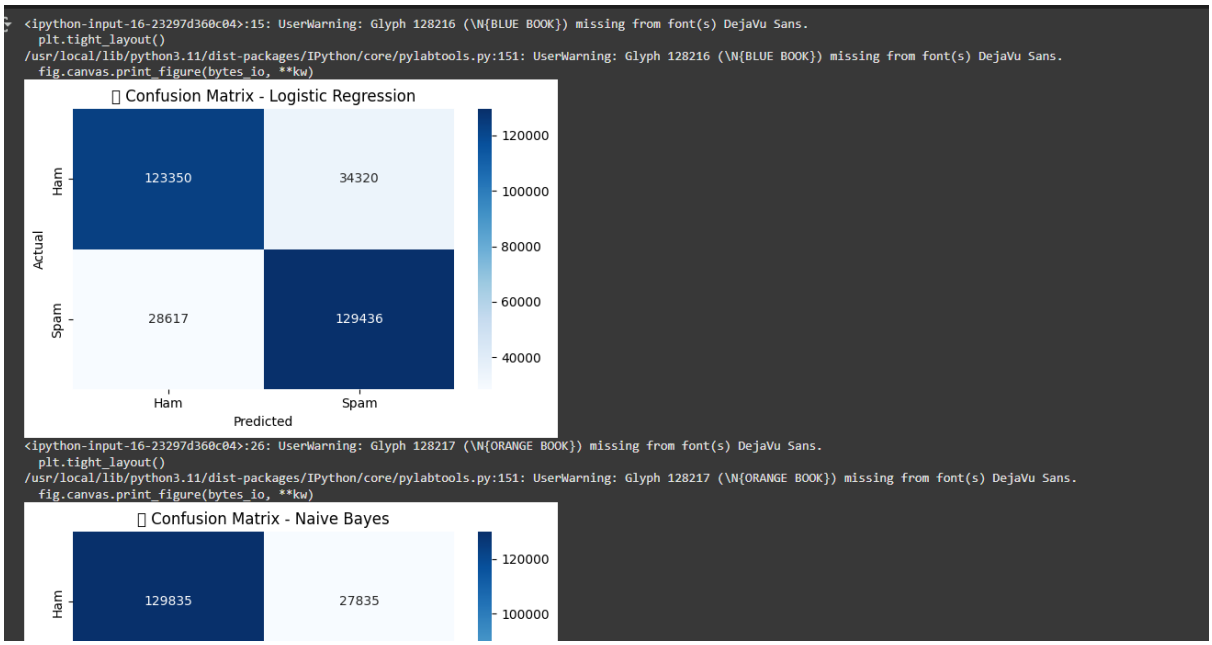
```
df['cleaned'] = df['SentimentText'].str.lower().str.replace(r'[^\w\s]', '', regex=True)


tfidf_vectorizer = TfidfVectorizer()
X_tfidf = tfidf_vectorizer.fit_transform(df['cleaned'])

# Target variable
y = df['Sentiment']

# Train-Test Split
X_train_tfidf, X_test_tfidf, y_train, y_test = train_test_split(
    X_tfidf, y, test_size=0.2, random_state=42
)



# Logistic Regression
log_reg_model = LogisticRegression()
log_reg_model.fit(X_train_tfidf, y_train)

# Multinomial Naive Bayes
nb_model = MultinomialNB()
nb_model.fit(X_train_tfidf, y_train)

print("✅ Logistic Regression Model Trained")
print("✅ Naive Bayes Model Trained")
```

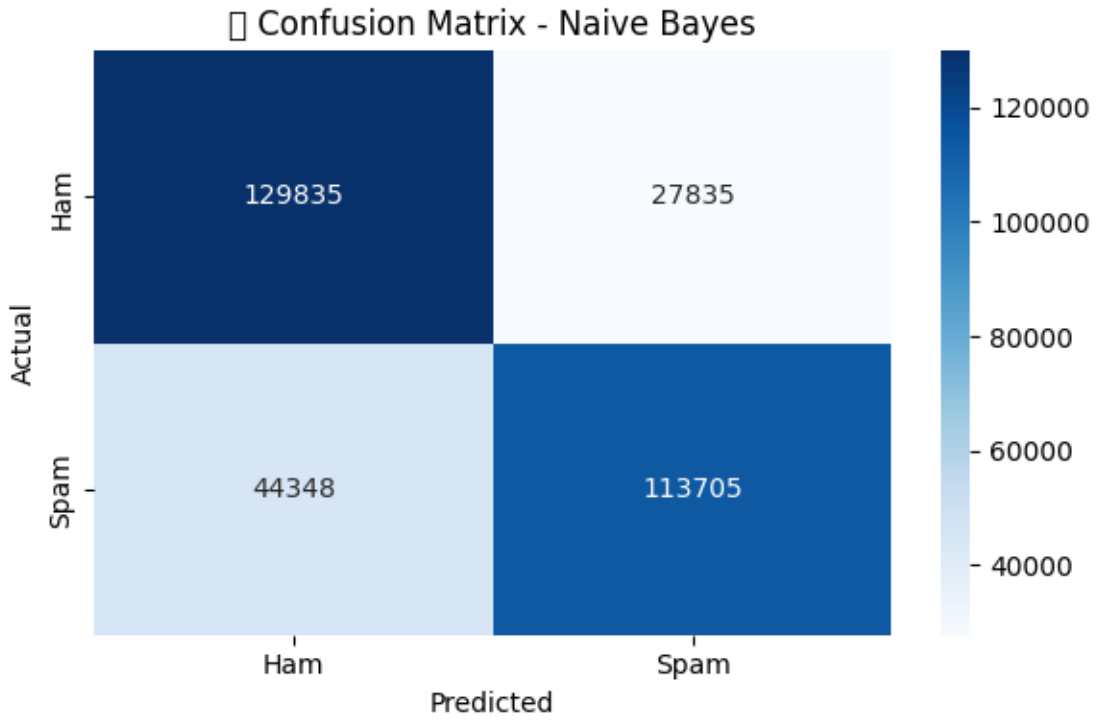
```
✅ Logistic Regression Model Trained
✅ Naive Bayes Model Trained
```

Lab Assignment 4- NLP Preprocessing And Text Classification

```
==================================================
🔍 Logistic Regression Evaluation
==================================================
✅ Accuracy: 0.8006575384118357

📋 Classification Report:
              precision    recall  f1-score   support

           0       0.81      0.78      0.80    157670
           1       0.79      0.82      0.80    158053

    accuracy                           0.80    315723
   macro avg       0.80      0.80      0.80    315723
weighted avg       0.80      0.80      0.80    315723

📊 Confusion Matrix:
 [[123350  34320]
 [ 28617 129436]]

==================================================
🔍 Naive Bayes Evaluation
==================================================
✅ Accuracy: 0.7713723738847027

📋 Classification Report:
              precision    recall  f1-score   support

           0       0.75      0.82      0.78    157670
           1       0.80      0.72      0.76    158053

    accuracy                           0.77    315723
   macro avg       0.77      0.77      0.77    315723
weighted avg       0.77      0.77      0.77    315723

📊 Confusion Matrix:
 [[129835  27835]
 [ 44348 113705]]
```

Lab Assignment 4- NLP Preprocessing And Text Classification





Lab Assignment 4- NLP Preprocessing And Text Classification

```
---- Sentiment Predictions on New Texts ----

Text 1: 'I just got a promotion at work, feeling amazing!'
  Logistic Regression Prediction: 0 (Negative)
  Naive Bayes Prediction: 0 (Negative)
------------------------------------------------------------
Text 2: 'Ugh, this day keeps getting worse.'
  Logistic Regression Prediction: 0 (Negative)
  Naive Bayes Prediction: 0 (Negative)
------------------------------------------------------------
Text 3: 'So happy to be back home with family.'
  Logistic Regression Prediction: 1 (Positive)
  Naive Bayes Prediction: 1 (Positive)
------------------------------------------------------------
Text 4: 'I'm really frustrated with how things turned out.'
  Logistic Regression Prediction: 1 (Positive)
  Naive Bayes Prediction: 0 (Negative)
------------------------------------------------------------
Text 5: 'The weather is beautiful today!'
  Logistic Regression Prediction: 1 (Positive)
  Naive Bayes Prediction: 1 (Positive)
------------------------------------------------------------
Text 6: 'Nothing ever goes my way...'
  Logistic Regression Prediction: 1 (Positive)
  Naive Bayes Prediction: 1 (Positive)
------------------------------------------------------------
Text 7: 'Enjoyed the concert so much, best night ever!'
  Logistic Regression Prediction: 1 (Positive)
  Naive Bayes Prediction: 1 (Positive)
------------------------------------------------------------
Text 8: 'I'm tired and completely out of energy.'
  Logistic Regression Prediction: 0 (Negative)
  Naive Bayes Prediction: 0 (Negative)
------------------------------------------------------------
Text 9: 'Life is good when you're surrounded by positivity.'
  Logistic Regression Prediction: 1 (Positive)
  Naive Bayes Prediction: 1 (Positive)
------------------------------------------------------------
Text 10: 'I'm done trying. Nothing works.'
  Logistic Regression Prediction: 1 (Positive)
  Naive Bayes Prediction: 1 (Positive)
------------------------------------------------------------
```