```
import pandas as pd
import re
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import classification_report, accuracy_score

# Load the dataset
df = pd.read_csv('/sentiment140dataset.csv', encoding='latin-1', header=None)
df.columns = ['target', 'id', 'date', 'flag', 'user', 'text']

# Simplify sentiment labels: 0 = Negative, 4 = Positive
df['target'] = df['target'].apply(lambda x: 0 if x == 0 else 1)

# Clean tweet text
def clean_text(text):
    text = re.sub(r'http\S+', '', text)  # remove URLs
    text = re.sub(r'@\w+', '', text)     # remove mentions
    text = re.sub(r'#\w+', '', text)     # remove hashtags
    text = re.sub(r'[^A-Za-z\s]', '', text)  # remove punctuation
    text = text.lower()  # lowercase
    return text

df['clean_text'] = df['text'].apply(clean_text)

# Split into train/test sets
X_train, X_test, y_train, y_test = train_test_split(
    df['clean_text'], df['target'], test_size=0.2, random_state=42
)

# Vectorize text
vectorizer = CountVectorizer(stop_words='english')
X_train_vec = vectorizer.fit_transform(X_train)
X_test_vec = vectorizer.transform(X_test)

# Train model
model = MultinomialNB()
model.fit(X_train_vec, y_train)

# Predict and evaluate
y_pred = model.predict(X_test_vec)
print("Accuracy:", accuracy_score(y_test, y_pred))
print("\nClassification Report:\n", classification_report(y_test, y_pred, target_names=['Negative', 'Positive']))
```

```
Accuracy: 0.72

Classification Report:
              precision    recall  f1-score   support

    Negative       0.47      0.54      0.50        26
    Positive       0.83      0.78      0.81        74

    accuracy                           0.72       100
   macro avg       0.65      0.66      0.65       100
weighted avg       0.73      0.72      0.73       100
```