

```
# Sentiment Analysis Project

# Step 1: Install & Import
!pip install nltk

import nltk
nltk.download("movie_reviews")
from nltk.corpus import movie_reviews
import random

from sklearn.feature_extraction.text import CountVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.pipeline import make_pipeline
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, classification_report
```

Requirement already satisfied: nltk in /usr/local/lib/python3.12/dist-packages (3.9.1)
Requirement already satisfied: click in /usr/local/lib/python3.12/dist-packages (from nltk) (8.2.1)
Requirement already satisfied: joblib in /usr/local/lib/python3.12/dist-packages (from nltk) (1.5.2)
Requirement already satisfied: regex>=2021.8.3 in /usr/local/lib/python3.12/dist-packages (from nltk) (2024.11.6)
Requirement already satisfied: tqdm in /usr/local/lib/python3.12/dist-packages (from nltk) (4.67.1)
[nltk_data] Downloading package movie_reviews to /root/nltk_data...
[nltk_data] Unzipping corpora/movie_reviews.zip.

```
# Step 2: Load Data
documents = [(movie_reviews.raw(fileid), category)
              for category in movie_reviews.categories()
              for fileid in movie_reviews.fileids(category)]

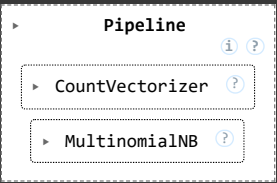
random.shuffle(documents)

texts = [doc for doc, label in documents]
labels = [label for doc, label in documents]
```

```
# Step 3: Train-Test Split
X_train, X_test, y_train, y_test = train_test_split(texts, labels, test_size=0.2, random_state=42)
```

```
# Step 4: Model Pipeline
model = make_pipeline(CountVectorizer(), MultinomialNB())
```

```
# Step 5: Train
model.fit(X_train, y_train)
```



```
# Step 6: Evaluate
y_pred = model.predict(X_test)
print("Accuracy:", accuracy_score(y_test, y_pred))
print("\nClassification Report:\n", classification_report(y_test, y_pred))
```

Accuracy: 0.805

Classification Report:

	precision	recall	f1-score	support
neg	0.78	0.83	0.80	193
pos	0.83	0.78	0.81	207
accuracy			0.81	400
macro avg	0.81	0.81	0.80	400
weighted avg	0.81	0.81	0.81	400

```
# Step 7: Test on custom text
custom_reviews = [
    "This movie was fantastic! The acting and story were brilliant.",
    "Terrible movie. Waste of time. Bad acting."
]
print("\nCustom Predictions:")
print(model.predict(custom_reviews))
```

Custom Predictions:
['pos' 'neg']

