slhysbixa

November 28, 2025

[3]:

```
from datasets import load_dataset
from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, f1_score, recall_score,␣
 ↪precision_score, log_loss
```

[2]:

```
# Load IMDb
dataset = load_dataset("stanfordnlp/imdb")

train_dataset = dataset["train"]
test_dataset = dataset["test"]

print("Train label counts:")
print(train_dataset.features["label"].names)
print({
    0: sum(1 for x in train_dataset["label"] if x == 0),
    1: sum(1 for x in train_dataset["label"] if x == 1)
})



X_train = train_dataset["text"]
y_train = train_dataset["label"]

X_test = test_dataset["text"]
y_test = test_dataset["label"]

print("\nTest label counts:")
print({
    0: sum(1 for x in test_dataset["label"] if x == 0),
    1: sum(1 for x in test_dataset["label"] if x == 1)
})

len(X_train), len(X_test)
```

```
Train label counts:
['neg', 'pos']
{0: 12500, 1: 12500}
```

```
Test label counts:
{0: 12500, 1: 12500}
```

[2]:

```
(25000, 25000)
```

[6]:

```python
# Bag-of-words vectorizer
bow_vectorizer = CountVectorizer(max_features=20000)
X_train_bow = bow_vectorizer.fit_transform(X_train)
X_test_bow = bow_vectorizer.transform(X_test)

bow_model = LogisticRegression(max_iter=1000)
bow_model.fit(X_train_bow, y_train)

y_pred_bow = bow_model.predict(X_test_bow)
y_proba_bow = bow_model.predict_proba(X_test_bow)

print("\n=== Bag-of-Words + Logistic Regression ===")
print(f"Accuracy:  {accuracy_score(y_test, y_pred_bow):.4f}")
print(f"F1:        {f1_score(y_test, y_pred_bow):.4f}")
print(f"Precision: {precision_score(y_test, y_pred_bow):.4f}")
print(f"Recall:    {recall_score(y_test, y_pred_bow):.4f}")
print(f"Loss:      {log_loss(y_test, y_proba_bow):.4f}")
```

```
=== Bag-of-Words + Logistic Regression ===
Accuracy:  0.8618
F1:        0.8606
Precision: 0.8683
Recall:    0.8531
Loss:      0.4338
```

[7]:

```python
tfidf_vectorizer = TfidfVectorizer(max_features=20000)
X_train_tfidf = tfidf_vectorizer.fit_transform(X_train)
X_test_tfidf = tfidf_vectorizer.transform(X_test)

tfidf_model = LogisticRegression(max_iter=2000)
tfidf_model.fit(X_train_tfidf, y_train)

y_pred_tfidf = tfidf_model.predict(X_test_tfidf)
y_proba_tfidf = tfidf_model.predict_proba(X_test_tfidf)

print("\n=== TF-IDF + Logistic Regression ===")
print(f"Accuracy:  {accuracy_score(y_test, y_pred_tfidf):.4f}")
print(f"F1:        {f1_score(y_test, y_pred_tfidf):.4f}")
print(f"Precision: {precision_score(y_test, y_pred_tfidf):.4f}")
print(f"Recall:    {recall_score(y_test, y_pred_tfidf):.4f}")
print(f"Loss:      {log_loss(y_test, y_proba_tfidf):.4f}")
```

```
=== TF-IDF + Logistic Regression ===
Accuracy:  0.8834
F1:        0.8832
Precision: 0.8847
Recall:    0.8818
Loss:      0.3181
```

```
[ ]:
```