

Lab11.2_LogisticRegression_N.Rukmini_2403A52402.ipynb

STEP 2 — Import Libraries

```
# Data handling
import numpy as np
import pandas as pd

# Text preprocessing
import re
import nltk
from nltk.corpus import stopwords

# Feature extraction
from sklearn.feature_extraction.text import TfidfVectorizer

# Model training
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split

# Evaluation
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix

# Visualization
import matplotlib.pyplot as plt
import seaborn as sns
```

STEP 3 — Load and Preprocess Data

```
# Example loading
data = pd.read_csv("/content/IMDB Dataset.csv", engine='python', on_bad_lines='warn')

# Download stopwords (first time only)
nltk.download('stopwords')
stop_words = set(stopwords.words('english'))

def clean_text(text):
    text = text.lower()
    text = re.sub(r'[^a-zA-Z]', ' ', text)
    words = text.split()
    words = [word for word in words if word not in stop_words]
    return " ".join(words)

data['clean_review'] = data['review'].apply(clean_text)

print(data[['review', 'clean_review']].head())
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
```

```
review \
0 One of the other reviewers has mentioned that ...
1 A wonderful little production. <br /><br />The...
2 I thought this was a wonderful way to spend ti...
3 Basically there's a family where a little boy ...
4 Petter Mattei's "Love in the Time of Money" is...

clean_review
0 one reviewers mentioned watching oz episode ho...
1 wonderful little production br br filming tech...
```

```

2 thought wonderful way spend time hot summer we...
3 basically family little boy jake thinks zombie...
4 petter mattei love time money visually stunnin...

```

STEP 4 — Feature Extraction (TF-IDF)

```

tfidf = TfidfVectorizer(max_features=5000)

X = tfidf.fit_transform(data['clean_review'])
y = data['sentiment'] # Changed 'label' to 'sentiment'

print("Vocabulary Size:", len(tfidf.vocabulary_))

```

Vocabulary Size: 5000

STEP 5 — Train Logistic Regression Model

```

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42
)

model = LogisticRegression(max_iter=200)
model.fit(X_train, y_train)

```

▼ LogisticRegression ⓘ ?
 LogisticRegression(max_iter=200)

STEP 6 — Model Evaluation

```

y_pred = model.predict(X_test)

print("Accuracy:", accuracy_score(y_test, y_pred))
print(classification_report(y_test, y_pred))

cm = confusion_matrix(y_test, y_pred)

sns.heatmap(cm, annot=True, fmt='d')
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.title("Confusion Matrix")
plt.show()

```

Accuracy: 0.8921

	precision	recall	f1-score	support
negative	0.90	0.88	0.89	4961
positive	0.88	0.91	0.89	5039
accuracy			0.89	10000
macro avg	0.89	0.89	0.89	10000
weighted avg	0.89	0.89	0.89	10000

