To perform sentiment analysis on the IMDB review dataset using the TF-IDF technique, follow these steps:

**1. Set Up Your Environment**

Ensure you have the necessary libraries installed. You can use Python libraries like scikit-learn, pandas, and nltk. Install them using:

bash

Copy code

pip install numpy pandas scikit-learn nltk

**2. Load the Data**

First, you need to load the IMDB dataset. If you don't have it, you can use the IMDb dataset from scikit-learn or download it from other sources. For this example, I'll assume you have the dataset in a CSV file format.

python

Copy code

import pandas as pd

# Load the dataset

data = pd.read\_csv('imdb\_reviews.csv') # Replace with the path to your dataset

print(data.head())

**3. Preprocess the Data**

Preprocessing includes text cleaning and splitting the data into training and test sets.

python

Copy code

from sklearn.model\_selection import train\_test\_split

from sklearn.feature\_extraction.text import TfidfVectorizer

# Define the features and labels

X = data['review'] # Assuming 'review' column contains the text

y = data['sentiment'] # Assuming 'sentiment' column contains the labels (positive/negative)

# Split the dataset into training and test sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Initialize TF-IDF Vectorizer

tfidf = TfidfVectorizer(stop\_words='english', max\_features=5000)

# Fit and transform the training data, transform the test data

X\_train\_tfidf = tfidf.fit\_transform(X\_train)

X\_test\_tfidf = tfidf.transform(X\_test)

**4. Train and Evaluate Models**

You can use different machine learning models to classify the sentiments. Let's use Logistic Regression, Naive Bayes, and Support Vector Machine (SVM) as examples.

python

Copy code

from sklearn.naive\_bayes import MultinomialNB

from sklearn.linear\_model import LogisticRegression

from sklearn.svm import SVC

from sklearn.metrics import classification\_report, accuracy\_score

# Initialize models

models = {

'Naive Bayes': MultinomialNB(),

'Logistic Regression': LogisticRegression(),

'SVM': SVC()

}

# Train and evaluate each model

for name, model in models.items():

model.fit(X\_train\_tfidf, y\_train)

y\_pred = model.predict(X\_test\_tfidf)

print(f"Model: {name}")

print(f"Accuracy: {accuracy\_score(y\_test, y\_pred)}")

print(classification\_report(y\_test, y\_pred))

print("-" \* 60)