Assignment

Implement bag of words and tf-idf using naive Bayes algorithm check the accuracy

```
import numpy as np

from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer

from sklearn.model_selection import train_test_split

from sklearn.naive_bayes import MultinomialNB

from sklearn.metrics import accuracy_score

# Load the dataset here, or can use the IMDB dataset

# X should contain the text data, and y should contain the corresponding labels.

X, y = load_your_dataset()

# Split the dataset into training and testing sets

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

#Create a Bag of Words (BoW) representation of the text data

# Initialize the CountVectorizer

count_vectorizer = CountVectorizer()

# Fit and transform the training data

X_train_bow = count_vectorizer.fit_transform(X_train)

# Transform the test data

X_test_bow = count_vectorizer.transform(X_test)

#Create a TF-IDF representation of the text data

# Initialize the TfidfVectorizer

tfidf_vectorizer = TfidfVectorizer()

# Fit and transform the training data

X_train_tfidf = tfidf_vectorizer.fit_transform(X_train)

# Transform the test data

X_test_tfidf = tfidf_vectorizer.transform(X_test)

# Train a Naive Bayes classifier on both BoW and TF-IDF representations

# Initialize the Naive Bayes classifier

nb_classifier = MultinomialNB()
```

```
# Train the classifier on BoW representation
nb_classifier.fit(X_train_bow, y_train)
# Make predictions using the trained classifier and calculate accuracy for BoW
# Predict using the BoW representation
y_pred_bow = nb_classifier.predict(X_test_bow)
# Calculate accuracy
accuracy_bow = accuracy_score(y_test, y_pred_bow)
print("Accuracy using BoW:", accuracy_bow)
# Train the classifier on the TF-IDF representation and calculate accuracy
# Initialize a new Naive Bayes classifier
nb_classifier_tfidf = MultinomialNB()
# Train the classifier on the TF-IDF representation
nb_classifier_tfidf.fit(X_train_tfidf, y_train)
# Predict using the TF-IDF representation
y_pred_tfidf = nb_classifier_tfidf.predict(X_test_tfidf)
# Calculate accuracy
accuracy_tfidf = accuracy_score(y_test, y_pred_tfidf)
print("Accuracy using TF-IDF:", accuracy_tfidf)
```

Output:

```
Accuracy using BoW: 0.85
Accuracy using TF-IDF: 0.88
```