In [13]:

```
# Import necessary libraries
from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score

# Sample dataset - replace this with your own dataset
texts = ["This is a positive sentence.", "This is a negative sentence.", "Another positive example.", "Another negative example."]
labels = [1, 0, 1, 0]  # 1 for positive, 0 for negative

# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(texts, labels, test_size=0.2, random_state=42)

# Create a Bag of Words (BoW) vectorizer
count_vectorizer = CountVectorizer()
X_train_counts = count_vectorizer.fit_transform(X_train)
X_test_counts = count_vectorizer.transform(X_test)

# Create a TF-IDF vectorizer
tfidf_vectorizer = TfidfVectorizer()
X_train_tfidf = tfidf_vectorizer.fit_transform(X_train)
X_test_tfidf = tfidf_vectorizer.transform(X_test)

# Initialize and train a Naive Bayes classifier for BoW
nb_classifier_bow = MultinomialNB()
nb_classifier_bow.fit(X_train_counts, y_train)

# Predict using BoW model
y_pred_bow = nb_classifier_bow.predict(X_test_counts)

# Calculate accuracy for BoW
accuracy_bow = accuracy_score(y_test, y_pred_bow)
print(f"Accuracy using BoW: {accuracy_bow:.2f}")

# Initialize and train a Naive Bayes classifier for TF-IDF
nb_classifier_tfidf = MultinomialNB()
nb_classifier_tfidf.fit(X_train_tfidf, y_train)

# Predict using TF-IDF model
y_pred_tfidf = nb_classifier_tfidf.predict(X_test_tfidf)

# Calculate accuracy for TF-IDF
accuracy_tfidf = accuracy_score(y_test, y_pred_tfidf)
print(f"Accuracy using TF-IDF: {accuracy_tfidf:.2f}")
```

```
Accuracy using BoW: 0.00
Accuracy using TF-IDF: 0.00
```

In [17]:

```
pip install scikit-learn
```

```
Requirement already satisfied: scikit-learn in c:\users\pavan\anaconda3\lib\site-packages (1.2.1)
Requirement already satisfied: scipy>=1.3.2 in c:\users\pavan\anaconda3\lib\site-packages (from scikit-learn) (1.10.0)
Requirement already satisfied: numpy>=1.17.3 in c:\users\pavan\anaconda3\lib\site-packages (from scikit-learn) (1.23.5)
Requirement already satisfied: joblib>=1.1.1 in c:\users\pavan\anaconda3\lib\site-packages (from scikit-learn) (1.1.1)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\pavan\anaconda3\lib\site-packages (from scikit-learn) (2.2.0)
Note: you may need to restart the kernel to use updated packages.
```

In [18]:

```
# Import necessary libraries
from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score

# Sample dataset
corpus = [
    "This is a positive sentence.",
    "This is a negative sentence.",
    "A positive review is great.",
    "A negative review is terrible.",
]

# Labels for the dataset (1 for positive, 0 for negative)
labels = [1, 0, 1, 0]

# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(corpus, labels, test_size=0.2, random_state=42)

# Create Bag of Words (BoW) and TF-IDF representations of the text
bow_vectorizer = CountVectorizer()
tfidf_vectorizer = TfidfVectorizer()

X_train_bow = bow_vectorizer.fit_transform(X_train)
X_test_bow = bow_vectorizer.transform(X_test)

X_train_tfidf = tfidf_vectorizer.fit_transform(X_train)
X_test_tfidf = tfidf_vectorizer.transform(X_test)

# Initialize and train a Naive Bayes classifier on BoW and TF-IDF features
nb_bow = MultinomialNB()
nb_tfidf = MultinomialNB()

nb_bow.fit(X_train_bow, y_train)
nb_tfidf.fit(X_train_tfidf, y_train)

# Make predictions on the test data
y_pred_bow = nb_bow.predict(X_test_bow)
y_pred_tfidf = nb_tfidf.predict(X_test_tfidf)

# Calculate and print accuracy
accuracy_bow = accuracy_score(y_test, y_pred_bow)
accuracy_tfidf = accuracy_score(y_test, y_pred_tfidf)

print(f"Accuracy using BoW: {accuracy_bow}")
print(f"Accuracy using TF-IDF: {accuracy_tfidf}")
```

```
Accuracy using BoW: 0.0
Accuracy using TF-IDF: 0.0
```

In [ ]: