In [1]:
```
import numpy as np
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
```

In [2]:
```
texts = [
    "This is a positive text.",
    "Negative sentiment here.",
    "Positive reviews are great!",
    "I don't like negative comments.",
]

labels = [1, 0, 1, 0]  # 1 for positive, 0 for negative
```

In [3]:
```
X_train, X_test, y_train, y_test = train_test_split(texts, labels, test_size=0.2, random_state=42)
```

In [4]:
```
count_vectorizer = CountVectorizer()
X_train_bow = count_vectorizer.fit_transform(X_train)
X_test_bow = count_vectorizer.transform(X_test)
```

In [5]:
```
tfidf_vectorizer = TfidfVectorizer()
X_train_tfidf = tfidf_vectorizer.fit_transform(X_train)
X_test_tfidf = tfidf_vectorizer.transform(X_test)
```

In [6]:
```
nb_classifier_bow = MultinomialNB()
nb_classifier_tfidf = MultinomialNB()

nb_classifier_bow.fit(X_train_bow, y_train)
nb_classifier_tfidf.fit(X_train_tfidf, y_train)
```

Out[6]:
```
▾ MultinomialNB
MultinomialNB()
```

In [7]:
```
y_pred_bow = nb_classifier_bow.predict(X_test_bow)
y_pred_tfidf = nb_classifier_tfidf.predict(X_test_tfidf)

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: 1.0
Accuracy using TF-IDF: 0.0
```