Lez16_Naive_Bias

January 6, 2023

[1]:
```
# Imports
import numpy as np
from sklearn.datasets import fetch_20newsgroups
from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer,␣
 ↪TfidfTransformer
from sklearn.naive_bayes import MultinomialNB
from sklearn import metrics

# The used dataset has 20 different classifications
# The dataset is already divided into a "train" and a "test" subset
documents_train = fetch_20newsgroups(subset='train')
documents_test = fetch_20newsgroups(subset='test')

# Print attributes of the documents
print(list(documents_train))
```

```
['data', 'filenames', 'target_names', 'target', 'DESCR']
```

[6]:
```
#For classification, a numerical representation of the documents is needed.
count_vect = CountVectorizer(analyzer = 'word') # Convert document into vector
train_data = count_vect.fit_transform(documents_train.data)
test_data = count_vect.transform(documents_test.data)

# The basic idea is, to count how often a certain word occurs within a document␣
 ↪and therefore classify the document.
# The tfidfTransformer is used to calculate the probabilities of a certain␣
 ↪classification
transformer = TfidfTransformer()
train_transformed = transformer.fit_transform(train_data)
test_transformed = transformer.transform(test_data)

# Set up classifier, MultinomialNB uses Naive Bayes algorithm for classification
classifier = MultinomialNB(alpha = 1)
#Train the classifier
classifier.fit(train_data, documents_train.target)
#Test the classifier
prediction = classifier.predict(test_transformed)
```

```
# Verify classification
accuracy = metrics.accuracy_score(prediction, documents_test.target)

print("Accuracy: ", accuracy)
```

```
Accuracy:  0.8147902283590016
```

[ ]: