d0stvpghi

May 30, 2023

ML Lab Assignment - 10 (Naive Bayes Classification)

Arya Chakraborty [22MSD7020]

1 Importing Dataset and the required libraries

[30]:
```
from sklearn.datasets import fetch_20newsgroups
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import accuracy_score
```

2 Here we are considering only 3 categories from 20newsgroups dataset

[31]:
```
categories = ['sci.space', 'comp.graphics', 'rec.sport.baseball']
train_data = fetch_20newsgroups(subset='train', categories=categories,␣
 ↪shuffle=True, random_state=42)
test_data = fetch_20newsgroups(subset='test', categories=categories,␣
 ↪shuffle=True, random_state=42)
```

[ ]:
```
print("Training Data:")
for i in range(5):
    print(f"Category: {train_data.target_names[train_data.target[i]]}")
    print(f"Sample:\n{train_data.data[i]}")
    print("=" * 50)
```

3 Preprocessing & Vectorotizing the training and testing data

[33]:
```
def preprocess(document):
    document = document.lower()
    tokens = word_tokenize(document)

    # Remove stopwords
    stop_words = set(stopwords.words('english'))
    tokens = [token for token in tokens if token not in stop_words]
```

```
    # Lemmatize the tokens
    lemmatizer = WordNetLemmatizer()
    tokens = [lemmatizer.lemmatize(token) for token in tokens]

    # joining the tokens
    preprocessed_document = ' '.join(tokens)

    return preprocessed_document
```

[35]:
```
# Preprocessing
X_train = [preprocess(document) for document in train_data.data]
X_test = [preprocess(document) for document in test_data.data]

# vectorizing
vectorizer = CountVectorizer()
X_train_preprocessed = vectorizer.fit_transform(train_data.data)
X_test_preprocessed = vectorizer.transform(test_data.data)
```

[37]:
```
classifier = MultinomialNB()
classifier.fit(X_train_preprocessed, train_data.target)
```

[37]:
```
MultinomialNB()
```

[38]:
```
predictions = classifier.predict(X_test_preprocessed)
```

[39]:
```
accuracy = accuracy_score(test_data.target, predictions)
print("Accuracy:", accuracy)
```

```
Accuracy: 0.9694915254237289
```

4 further future prediction

[40]:
```
new_document = """
Astronauts aboard the International Space Station conducted a spacewalk to␣
 ↪repair
a faulty solar panel. The spacewalk lasted for several hours and was successful␣
 ↪in
restoring power generation to the ISS. This mission marks an important␣
 ↪milestone in
space exploration and demonstrates the capabilities of human spaceflight.
"""
```

[17]:
```
import nltk
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
```

```
from nltk.stem import WordNetLemmatizer

nltk.download('stopwords')
nltk.download('punkt')
nltk.download('wordnet')
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data…
[nltk_data]   Unzipping corpora/stopwords.zip.
[nltk_data] Downloading package punkt to /root/nltk_data…
[nltk_data]   Unzipping tokenizers/punkt.zip.
[nltk_data] Downloading package wordnet to /root/nltk_data…
```

[17]: True

5 user defined function for preprocessing the new document and predicting it's class

[45]:
```
def new_prediction(x):
  # Preprocessing
  preprocessed_document = preprocess(x)
  feature_vector = vectorizer.transform([preprocessed_document])

  # Model prediction
  predicted_class = classifier.predict(feature_vector)
  predicted_class_index = predicted_class[0]
  predicted_class_name = train_data.target_names[predicted_class_index]
  print("Predicted Class:", predicted_class_name)
```

[46]:
```
new_prediction(new_document)
```

```
Predicted Class: sci.space
```