

Sentiment Analysis model using the Naive Bayes

Importing necessary libraries

```
In [4]: import nltk
nltk.download('movie_reviews')
from nltk.corpus import movie_reviews
from nltk.classify import NaiveBayesClassifier
from nltk.classify.util import accuracy as nltk_accuracy

[nltk_data] Downloading package movie_reviews to
[nltk_data] C:\Users\Refat\AppData\Roaming\nltk_data...
[nltk_data] Unzipping corpora\movie_reviews.zip.
```

Function to extract features from words

```
In [2]: def extract_features(words):
        return dict([(word, True) for word in words])
```

Load movie reviews from NLTK

```
In [15]: fileids_pos = movie_reviews.fileids('pos')

fileids_neg = movie_reviews.fileids('neg')
```

Extract features from movie reviews

```
In [17]: features_pos = [(extract_features(movie_reviews.words(fileids=[f])), 'Positive') for f in fileids_pos]
features_neg = [(extract_features(movie_reviews.words(fileids=[f])), 'Negative') for f in fileids_neg]
```

Split the data into training and testing datasets

```
In [7]: threshold = 0.8
num_pos = int(threshold * len(features_pos))
num_neg = int(threshold * len(features_neg))
```

Create training and testing datasets

```
In [8]: features_train = features_pos[:num_pos] + features_neg[:num_neg]
features_test = features_pos[num_pos:] + features_neg[num_neg:]
```

Train a Naive Bayes classifier

```
In [9]: classifier = NaiveBayesClassifier.train(features_train)
```

Test the classifier

```
In [10]: accuracy = nltk_accuracy(classifier, features_test)
print("Accuracy:", accuracy)
```

Accuracy: 0.735

Example usage of the classifier

```
In [24]: input_reviews = [  
    "The movie was great!",  
    "The movie was terrible...",  
    "an action-packed South Indian film that seamlessly blends intense storytelling with breathtaking visuals"  
    "Very bad movies masquerading as a blockbuster! Salhaar tries to dazzle with its style but it all a glossy  
    ]  
  
for review in input_reviews:  
    print("\nReview:", review)  
    probdist = classifier.prob_classify(extract_features(review.split()))  
    pred_sentiment = probdist.max()  
  
    print("Predicted sentiment:", pred_sentiment)  
    print("Probability:", round(probdist.prob(pred_sentiment), 2))
```

Review: The movie was great!
Predicted sentiment: Negative
Probability: 0.53

Review: The movie was terrible...
Predicted sentiment: Negative
Probability: 0.53

Review: an action-packed South Indian film that seamlessly blends intense storytelling with breathtaking visuals
Predicted sentiment: Positive
Probability: 0.99

Review: Very bad movies masquerading as a blockbuster! Salhaar tries to dazzle with its style but it all a glossy facade with no real depth
Predicted sentiment: Negative
Probability: 0.51

In []: