```
import nltk
from nltk.corpus import movie_reviews
import random
```

```
# Load movie review dataset (positive and negative reviews)
nltk.download('movie_reviews')
```

```
[nltk_data] Downloading package movie_reviews to /root/nltk_data...
[nltk_data]   Unzipping corpora/movie_reviews.zip.
True
```

```
# Prepare dataset
documents = [(list(movie_reviews.words(fileid)), category)
             for category in movie_reviews.categories()
             for fileid in movie_reviews.fileids(category)]

# Shuffle data
random.shuffle(documents)

# Feature extractor: presence of words
all_words = nltk.FreqDist(w.lower() for w in movie_reviews.words())
word_features = list(all_words)[:2000]

def document_features(document):
    words = set(document)
    features = {}
    for word in word_features:
        features[f'contains({word})'] = (word in words)
    return features

# Prepare feature sets
featuresets = [(document_features(d), c) for (d, c) in documents]

# Split into train and test
train_set, test_set = featuresets[:1900], featuresets[1900:]

# Train Naive Bayes Classifier (applies Bayesian estimation)
classifier = nltk.NaiveBayesClassifier.train(train_set)

# Evaluate
accuracy = nltk.classify.accuracy(classifier, test_set)
print(f"Accuracy: {accuracy:.4f}")

# Show most informative features
classifier.show_most_informative_features(5)
```

```
Accuracy: 0.8300
Most Informative Features
   contains(outstanding) = True              pos : neg    =     11.3 : 1.0
   contains(wonderfully) = True              pos : neg    =      9.9 : 1.0
         contains(mulan) = True              pos : neg    =      9.0 : 1.0
        contains(seagal) = True              neg : pos    =      8.2 : 1.0
        contains(poorly) = True              neg : pos    =      6.1 : 1.0
```