```
import nltk
from textblob import TextBlob

# Download necessary NLTK data
nltk.download('punkt')



from nltk.corpus import movie_reviews

# Load movie review data
reviews = []
for fileid in movie_reviews.fileids():
    review = movie_reviews.raw(fileid)
    reviews.append(review)
```

```
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from nltk.stem import WordNetLemmatizer

# Preprocess data
stop_words = set(stopwords.words('english'))
lemmatizer = WordNetLemmatizer()

def preprocess_text(text):
    # Tokenize text
    tokens = word_tokenize(text)

    # Remove stop words
    filtered_tokens = [token for token in tokens if token.lower() not in stop_words]

    # Lemmatize words
    lemmatized_tokens = [lemmatizer.lemmatize(token) for token in filtered_tokens]
    # Rejoin tokens into string
    preprocessed_text = ' '.join(lemmatized_tokens)

    return preprocessed_text

# Apply preprocessing to movie review data
preprocessed_reviews = []
for review in reviews:
    preprocessed_review = preprocess_text(review)

preprocessed_reviews.append(preprocessed_review)



# Analyze sentiment of movie reviews
sentiments = []
for review in preprocessed_reviews:
    blob = TextBlob(review)
    sentiment = blob.sentiment.polarity
    if sentiment > 0:
        sentiments.append('positive')
    elif sentiment < 0:
        sentiments.append('negative')
    else:
        sentiments.append('neutral')


import matplotlib.pyplot as plt

# Count number of positive, negative, and neutral reviews
count_positive = sentiments.count('positive')
count_negative = sentiments.count('negative')
count_neutral = sentiments.count('neutral')

# Visualize results
labels = ['Positive', 'Negative', 'Neutral']
sizes = [count_positive, count_negative, count_neutral]
colors = ['yellowgreen', 'lightcoral', 'gold']

plt.pie(sizes, labels=labels, colors=colors, autopct
```