NAME.   **:** *ASHOK K*

Email   ***ID :*** *ashokk2712004@gmail.com@gmail.com*

TASK -1 :

_____

PROGRAM:
```
import random
import nltk
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import accuracy_score, classification_report
from nltk.corpus import movie_reviews

# Download NLTK data
nltk.download('movie_reviews')

# Load and shuffle dataset
documents = [(list(movie_reviews.words(fileid)), category)
        for category in movie_reviews.categories()
        for fileid in movie_reviews.fileids(category)]
random.shuffle(documents)

# Prepare data
train_docs, test_docs = train_test_split(documents, test_size=0.25, random_state=42)
train_texts = [' '.join(doc) for doc, _ in train_docs]
train_labels = [label for _, label in train_docs]
test_texts = [' '.join(doc) for doc, _ in test_docs]
test_labels = [label for _, label in test_docs]

# Feature extraction and model training
vectorizer = CountVectorizer()
X_train = vectorizer.fit_transform(train_texts)
X_test = vectorizer.transform(test_texts)
classifier = MultinomialNB().fit(X_train, train_labels)
```

```
# Predictions and evaluation
predictions = classifier.predict(X_test)
print(f'Accuracy: {accuracy_score(test_labels, predictions)}')
print('Classification Report:')
print(classification_report(test_labels, predictions))
```

 EXPECTED OUTPUT:

Using built-in dataset as an alternative

Overall Sentiment Distribution:

Accuracy: 0.782

```
Classification Report:
              precision    recall  f1-score   support

         neg       0.76      0.80      0.78       241
         pos       0.80      0.76      0.78       259

    accuracy                           0.78       500
   macro avg       0.78      0.78      0.78       500
weighted avg       0.78      0.78      0.78
```

****