

Project: Sentiment Analysis on Product Reviews

✓ 1) Install and Import Required libraries

```
# Install necessary libraries if not already installed
!pip install nltk scikit-learn pandas matplotlib

# Import modules
import pandas as pd
import numpy as np
import nltk
import re
import string
import matplotlib.pyplot as plt

from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import accuracy_score, confusion_matrix,
classification_report

# Download NLTK resources
nltk.download('stopwords')
nltk.download('punkt')
nltk.download('punkt_tab') # Added this line to download the missing resource
from nltk.corpus import stopwords
```

```
Requirement already satisfied: nltk in /usr/local/lib/python3.12/dist-packages (3.9.1)
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.12/dist-packages (1.6.1)
Requirement already satisfied: pandas in /usr/local/lib/python3.12/dist-packages (2.2.2)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.12/dist-packages (3.10.0)
Requirement already satisfied: click in /usr/local/lib/python3.12/dist-packages (from nltk) (8.3.0)
Requirement already satisfied: joblib in /usr/local/lib/python3.12/dist-packages (from nltk) (1.5.2)
Requirement already satisfied: regex<=2021.8.3 in /usr/local/lib/python3.12/dist-packages (from nltk) (2024.11.6)
Requirement already satisfied: tqdm in /usr/local/lib/python3.12/dist-packages (from nltk) (4.67.1)
Requirement already satisfied: numpy>=1.19.5 in /usr/local/lib/python3.12/dist-packages (from scikit-learn) (2.0.2)
Requirement already satisfied: scipy>=1.6.0 in /usr/local/lib/python3.12/dist-packages (from scikit-learn) (1.16.2)
Requirement already satisfied: threadpoolctl>=3.1.0 in /usr/local/lib/python3.12/dist-packages (from scikit-learn) (3.6.0)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.12/dist-packages (from pandas) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.12/dist-packages (from pandas) (2025.2)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.12/dist-packages (from pandas) (2025.2)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (1.3.3)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (4.60.1)
Requirement already satisfied: kiwisolver>=1.3.1 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (1.4.9)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (25.0)
Requirement already satisfied: pillow>=8 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (11.3.0)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (3.2.5)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.12/dist-packages (from python-dateutil>=2.8.2->pandas) (1.17)
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data] Package stopwords is already up-to-date!
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data] Package punkt is already up-to-date!
[nltk_data] Downloading package punkt_tab to /root/nltk_data...
[nltk_data] Package punkt_tab is already up-to-date!
```

✓ 2) Sample Dataset

```
# Example dataset
data = {
    'review': [
        "I love this phone, the battery life is amazing!",
        "Worst purchase ever. It stopped working in 2 days.",
        "Great sound quality and very comfortable to wear.",
        "The product was damaged when delivered. Totally disappointed.",
        "Excellent quality and fast delivery. Worth every penny!",
        "Battery drains quickly and camera quality is poor.",
        "Absolutely fantastic! Exceeded my expectations.",
    ]
}
```

```

    "Not worth the money. Poor build quality.",
    "Very satisfied with the performance.",
    "Terrible experience. The device overheats constantly."
],
'sentiment': [
    'positive', 'negative', 'positive', 'negative', 'positive',
    'negative', 'positive', 'negative', 'positive', 'negative'
]
}

df = pd.DataFrame(data)
print(df.head())

```

```

          review sentiment
0  I love this phone, the battery life is amazing!  positive
1  Worst purchase ever. It stopped working in 2 d...  negative
2  Great sound quality and very comfortable to wear.  positive
3  The product was damaged when delivered. Total...  negative
4  Excellent quality and fast delivery. Worth eve...  positive

```

3) Text Preprocessing

```

stop_words = set(stopwords.words('english'))

def clean_text(text):
    text = text.lower()
    text = re.sub(r'http\S+|www\S+', '', text) # remove links
    text = re.sub(r'[^\w\s]', '', text) # remove punctuation/numbers
    tokens = nltk.word_tokenize(text)
    tokens = [w for w in tokens if w not in stop_words]
    return " ".join(tokens)

df['clean_review'] = df['review'].apply(clean_text)
df.head()

```

	review	sentiment	clean_review
0	I love this phone, the battery life is amazing!	positive	love phone battery life amazing
1	Worst purchase ever. It stopped working in 2 d...	negative	worst purchase ever stopped working days
2	Great sound quality and very comfortable to wear.	positive	great sound quality comfortable wear
3	The product was damaged when delivered. Total...	negative	product damaged delivered totally disappointed
4	Excellent quality and fast delivery. Worth eve...	positive	excellent quality fast delivery worth every penny

4) Feature Extraction (TF-IDF)

```

vectorizer = TfidfVectorizer(max_features=1000)
X = vectorizer.fit_transform(df['clean_review'])
y = df['sentiment']

```

5) Split Data & Train Model

```

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

model = MultinomialNB()
model.fit(X_train, y_train)

y_pred = model.predict(X_test)

```

6) Model Evaluation

```
print("✅ Accuracy:", accuracy_score(y_test, y_pred))
print("\nClassification Report:\n", classification_report(y_test, y_pred))

# Confusion matrix
cm = confusion_matrix(y_test, y_pred)
print("\nConfusion Matrix:\n", cm)

plt.imshow(cm, cmap='Blues')
plt.title("Confusion Matrix")
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.show()
```

✅ Accuracy: 0.3333333333333333

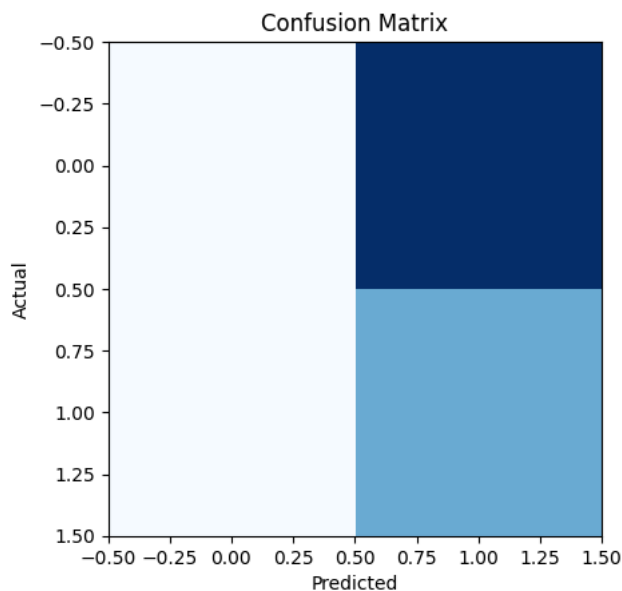
```
Classification Report:
              precision    recall  f1-score   support

   negative         0.00      0.00      0.00         2
   positive         0.33      1.00      0.50         1

   accuracy          0.17      0.50      0.33         3
  macro avg          0.17      0.50      0.25         3
 weighted avg          0.11      0.33      0.17         3
```

Confusion Matrix:

```
[[0 2]
 [0 1]]
/usr/local/lib/python3.12/dist-packages/sklearn/metrics/_classification.py:1565: UndefinedMetricWarning: Precision is ill-defined for
_warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
/usr/local/lib/python3.12/dist-packages/sklearn/metrics/_classification.py:1565: UndefinedMetricWarning: Precision is ill-defined for
_warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
/usr/local/lib/python3.12/dist-packages/sklearn/metrics/_classification.py:1565: UndefinedMetricWarning: Precision is ill-defined for
_warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
```



7) Predict Sentiment for New Reviews

```
def predict_sentiment(new_review):
    cleaned = clean_text(new_review)
    vector = vectorizer.transform([cleaned])
    prediction = model.predict(vector)[0]
    return prediction

# Try a few examples
examples = [
```

```
"This product is fantastic and works perfectly!",  
"Completely useless, I want a refund.",  
"Average quality but acceptable for the price."  
]  
  
for review in examples:  
    print(f"Review: {review}\nPredicted Sentiment: {predict_sentiment(review)}\n")
```

```
Review: This product is fantastic and works perfectly!  
Predicted Sentiment: positive
```

```
Review: Completely useless, I want a refund.  
Predicted Sentiment: positive
```

```
Review: Average quality but acceptable for the price.  
Predicted Sentiment: positive
```