

8. Naive Bayes Classification: Sentiment analysis on a twitter dataset.

from sklearn.model_selection import train_test_split

from sklearn.feature_extraction.text import CountVectorizer

from sklearn.naive_bayes import MultinomialNB

from sklearn.metrics import classification_report, accuracy_score

url = download('stopwords')

from nltk.corpus import stopwords
import string

data = {

'tweet': [

"I love this phone!",

"This movie is terrible...",

"Had an awesome day today :)",

"I have waiting in traffic!",

"Such a boring game."

"What a great experience!",

"worst customer service."

"feeling happy & blessed!"

]

'sentiment': [

'positive', 'negative', 'positive',

'negative', 'negative',

'positive', 'negative', 'positive',

'negative', 'positive'

3

df = pd.DataFrame(data)

df.preprocess_text(text):

text = text.lower()

text = "".join([char for char in text if char not in string.punctuation])

tokens = text.split()

tokens = [word for word in tokens if word not in stopwords.words('English')]

return ' '.join(tokens)

df['clean_tweet'] = df['tweet'].apply(preprocess_text)

vectorizer = CountVectorizer()

X = vectorizer.fit_transform(df['clean_tweet'])

Y = df['sentiment']

~~X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.2, random_state=42)~~

nb_classifier = MultinomialNB()

nb_classifier.fit(X_train, y_train)

y_pred = nb_classifier.predict(X_test)


```
print("Accuracy:", accuracy_score  
      (y_dec, y_pred))
```

```
print("In Classification Report: ")
```

```
print(classification_report(y_dec,  
                             y_pred))
```

OUTPUT :

Accuracy : 0.0

Classification report:

	Precision	recall	F1-Score	Support
negative	0.00	0.00	0.00	2.0
positive	0.00	0.00	0.00	0.0
Accuracy			0.00	2.0
macro avg	0.00	0.00	0.00	2.0
weighted avg	0.00	0.00	0.00	2.0