

```

import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
import nltk
nltk.download('stopwords')
nltk.download('punkt')
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from nltk.stem.porter import PorterStemmer

[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data] Unzipping corpora/stopwords.zip.
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data] Unzipping tokenizers/punkt.zip.

train_data = pd.read_csv('/content/twitter_training.csv.zip')
valid_data = pd.read_csv('/content/twitter_validation.csv')

print(train_data.head())
print(valid_data.head())

      2401  Borderlands  Positive  \
0   2401  Borderlands  Positive
1   2401  Borderlands  Positive
2   2401  Borderlands  Positive
3   2401  Borderlands  Positive
4   2401  Borderlands  Positive

      im getting on borderlands and i will murder you all ,
0   I am coming to the borders and I will kill you...
1   im getting on borderlands and i will kill you ...
2   im coming on borderlands and i will murder you...
3   im getting on borderlands 2 and i will murder ...
4   im getting into borderlands and i can murder y...
3364  Facebook Irrelevant  \
0   352   Amazon   Neutral
1  8312  Microsoft  Negative
2  4371    CS-GO   Negative
3  4433   Google   Neutral
4  6273    FIFA   Negative

      I mentioned on Facebook that I was struggling for motivation to go for a run the other day, which has been translated by Tom's great a
0   BBC News - Amazon boss Jeff Bezos rejects clai...
1   @Microsoft Why do I pay for WORD when it funct...
2   CSGO matchmaking is so full of closet hacking,...
3   Now the President is slapping Americans in the...
4   Hi @EAHelp I've had Madeleine McCann in my cel...

```



```

train_data.columns = ['ID', 'Game', 'Reviews', 'Comments']
valid_data.columns = ['ID', 'Game', 'Reviews', 'Comments']

print(train_data.columns)

# Check the column names of the DataFrame
print(valid_data.columns)

Index(['ID', 'Game', 'Reviews', 'Comments'], dtype='object')
Index(['ID', 'Game', 'Reviews', 'Comments'], dtype='object')

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

import numpy as np

def preprocess(text):

    # Convert null/NaN values to empty strings
    if isinstance(text, float) and np.isnan(text):
        text = ''

    # Convert text to lowercase
    text = text.lower()

    # Tokenize text

```

```

words = word_tokenize(text)

# Remove stop words
words = [word for word in words if word not in stop_words]

# Stem words
words = [stemmer.stem(word) for word in words]

# Join words
text = ' '.join(words)

return text

train_data['Comments'] = train_data['Comments'].apply(preprocess)
valid_data['Comments'] = valid_data['Comments'].apply(preprocess)

tfidf = TfidfVectorizer(max_features=10000)
X_train = tfidf.fit_transform(train_data['Comments'])
X_valid = tfidf.transform(valid_data['Comments'])
y_train = train_data['Reviews']
y_valid = valid_data['Reviews']

# Train a logistic regression model on the training data
lr = LogisticRegression(max_iter=1000)
lr.fit(X_train, y_train)

# Predict the sentiment of the validation data using the trained model
y_pred = lr.predict(X_valid)

# Calculate the accuracy of the model on the validation data
accuracy = accuracy_score(y_valid, y_pred)

# Print the accuracy score
print("Accuracy:", accuracy)

    Accuracy: 0.8588588588588588

from sklearn.tree import DecisionTreeClassifier

# Create a decision tree classifier
dt = DecisionTreeClassifier()

# Train the model on the training data
dt.fit(X_train, y_train)

# Predict the sentiment of the validation data using the trained model
y_pred = dt.predict(X_valid)

# Calculate the accuracy of the model on the validation data
accuracy = accuracy_score(y_valid, y_pred)

# Print the accuracy score
print("Accuracy:", accuracy)

    Accuracy: 0.8948948948948949

from sklearn.ensemble import RandomForestClassifier

# Create a random forest classifier with 100 trees
rf = RandomForestClassifier(n_estimators=100)

# Train the model on the training data
rf.fit(X_train, y_train)

# Predict the sentiment of the validation data using the trained model
y_pred = rf.predict(X_valid)

# Calculate the accuracy of the model on the validation data
accuracy = accuracy_score(y_valid, y_pred)

# Print the accuracy score
print("Accuracy:", accuracy)

    Accuracy: 0.943943943943944

```

[Colab paid products](#) - [Cancel contracts here](#)

