

```

import pandas as pd
import numpy as np
import re
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import classification_report, accuracy_score
from nltk.corpus import stopwords
import nltk

# dataset loading
nltk.download('stopwords')
stop_words = set(stopwords.words('english'))
data = pd.read_csv('twitter_data.csv', encoding='ISO-8859-1', on_bad_lines='skip',
quoting=3)
data.columns = ['target', 'ids', 'date', 'flag', 'user', 'source', 'text']
#data Preprocessing
def clean_text(text):
    text = str(text)
    text = re.sub(r"http\S+|www\S+|https\S+", '', text, flags=re.MULTILINE) #
Remove URLs
    text = re.sub(r'@\w+|\#', '', text) # Remove mentions and hashtags
    text = re.sub(r'[^A-Za-z\s]', '', text) # Remove special characters
    text = text.lower() # Convert to lowercase
    text = ' '.join(word for word in text.split() if word not in stop_words) #
Remove stopwords
    return text

data['cleaned_text'] = data['text'].apply(clean_text)
print(data.shape)

print(data.head(3))

# feature extraction using TF-IDF
vectorizer = TfidfVectorizer(max_features=5000) # Increased feature size for
better representation
X = vectorizer.fit_transform(data['cleaned_text']).toarray()
y = data['target']
#training and testing sets
X_train, X_test, y_train, y_test= train_test_split(X, y, test_size=0.2,
random_state=1000)
model = KNeighborsClassifier(n_neighbors=250, weights='distance')
model.fit(X_train, y_train)
# model evaluation
y_pred = model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred) * 100
print(f"Accuracy: {accuracy:.2f}%")

```