```
data_news.head()
```

	category	headline	links	short_description	keywords
0	WELLNESS	143 Miles in 35 Days: Lessons Learned	https://www.huffingtonpost.com/entry/running-l...	Resting is part of training. I've confirmed wh...	running-lessons
1	WELLNESS	Talking to Yourself: Crazy or Crazy Helpful?	https://www.huffingtonpost.com/entry/talking-t...	Think of talking to yourself as a tool to coac...	talking-to-yourself-crazy
2	WELLNESS	Crenezumab: Trial Will Gauge Whether Alzheimer...	https://www.huffingtonpost.com/entry/crenezuma...	The clock is ticking for the United States to ...	crenezumab-alzheimers-disease-drug
3	WELLNESS	Oh, What a Difference She Made	https://www.huffingtonpost.com/entry/meaningfu...	If you want to be busy, keep trying to be perf...	meaningful-life
4	WELLNESS	Green Superfoods	https://www.huffingtonpost.com/entry/green-sup...	First, the bad news: Soda bread, corned beef a...	green-superfoods

Next steps: Generate code with data_news | View recommended plots | New interactive sheet

```
import pandas as pd
import re
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
import nltk
nltk.download('punkt_tab')
nltk.download('stopwords')
from sklearn.feature_extraction.text import TfidfVectorizer


data_news = pd.read_csv('data_news.csv')
print(data_news.head())
print(data_news.info())
print(data_news.isnull().sum())

# Check the distribution of categories
print(data_news['category'].value_counts())


def clean_text(text):
    text = re.sub(r'<.*?>', '', text)  # Remove HTML tags
    text = re.sub(r'[^a-zA-Z\s]', '', text)  # Remove non-letters and non-spaces
    text = text.strip().lower()  # Remove leading/trailing spaces and convert
    to lower case
    return text

data_news['clean_text'] = data_news['short_description'].apply(clean_text)


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

# Tokenize and remove stop words
data_news['tokens'] = data_news['clean_text'].apply(lambda x: [word for word in
word_tokenize(x) if word not in stop_words])
```

```
[nltk_data] Downloading package punkt_tab to /root/nltk_data...
[nltk_data]   Package punkt_tab is already up-to-date!
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
   category                                          headline  \
0  WELLNESS              143 Miles in 35 Days: Lessons Learned
1  WELLNESS       Talking to Yourself: Crazy or Crazy Helpful?
2  WELLNESS  Crenezumab: Trial Will Gauge Whether Alzheimer...
3  WELLNESS                     Oh, What a Difference She Made
4  WELLNESS                                   Green Superfoods

                                               links  \
0  https://www.huffingtonpost.com/entry/running-l...
1  https://www.huffingtonpost.com/entry/talking-t...
2  https://www.huffingtonpost.com/entry/crenezuma...
3  https://www.huffingtonpost.com/entry/meaningfu...
4  https://www.huffingtonpost.com/entry/green-sup...

                                   short_description  \
0  Resting is part of training. I've confirmed wh...
1  Think of talking to yourself as a tool to coac...
2  The clock is ticking for the United States to ...
3  If you want to be busy, keep trying to be perf...
4  First, the bad news: Soda bread, corned beef a...
```

```
                             keywords
0                     running-lessons
1           talking-to-yourself-crazy
2  crenezumab-alzheimers-disease-drug
3                     meaningful-life
4                    green-superfoods
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 50000 entries, 0 to 49999
Data columns (total 5 columns):
 #   Column             Non-Null Count  Dtype 
---  ------             --------------  ----- 
 0   category           50000 non-null  object
 1   headline           50000 non-null  object
 2   links              50000 non-null  object
 3   short_description  50000 non-null  object
 4   keywords           47332 non-null  object
dtypes: object(5)
memory usage: 1.9+ MB
None
category                0
headline                0
links                   0
short_description       0
keywords             2668
dtype: int64
category
WELLNESS          5000
POLITICS          5000
ENTERTAINMENT     5000
TRAVEL            5000
STYLE & BEAUTY    5000
PARENTING         5000
```

```
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer

# Create TF-IDF features
tfidf = TfidfVectorizer(max_features=1000)  # Limiting to the top 1000 features
features = tfidf.fit_transform(data_news['clean_text'].apply(lambda x: ''.join(x)))
features.shape

# Labels
labels = data_news['category']

# Split the data
X_train, X_test, y_train, y_test = train_test_split(features, labels, test_size=0.2, random_state=42)
```

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report




# Initialize and train the logistic regression model
model = LogisticRegression()
model.fit(X_train, y_train)

# Predict on the test set
y_pred = model.predict(X_test)

# Evaluate the model
print(classification_report(y_test, y_pred))
```

```
                precision    recall  f1-score   support

      BUSINESS       0.49      0.50      0.50       955
 ENTERTAINMENT       0.42      0.40      0.41       985
  FOOD & DRINK       0.56      0.58      0.57      1021
     PARENTING       0.63      0.57      0.60      1030
      POLITICS       0.53      0.44      0.48      1034
        SPORTS       0.50      0.57      0.53       995
STYLE & BEAUTY       0.61      0.56      0.58       986
        TRAVEL       0.57      0.56      0.56      1008
      WELLNESS       0.52      0.59      0.55      1009
    WORLD NEWS       0.49      0.56      0.52       977

      accuracy                           0.53     10000
     macro avg       0.53      0.53      0.53     10000
  weighted avg       0.53      0.53      0.53     10000

/usr/local/lib/python3.11/dist-packages/sklearn/linear_model/_logistic.py:465: ConvergenceWarning: lbfgs failed to converge (status=
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.
```

```
Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
```