```
import pandas as pd
import numpy as np
import re
import string

import nltk
from nltk.corpus import stopwords
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, classification_report



import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

# Input data files are available in the read-only "../input/" directory
# For example, running this (by clicking run or pressing Shift+Enter) will list all files under the input directory

import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))



news=pd.read_csv("/content/FA-KES-Dataset 2.csv", encoding='latin1')


news.head()
```

	unit_id	article_title	article_content	source	date	location	labels
0	1914947530	Syria attack symptoms consistent with nerve ag...	Wed 05 Apr 2017 Syria attack symptoms consiste...	nna	4/5/2017	idlib	0
1	1914947532	Homs governor says U.S. attack caused deaths b...	Fri 07 Apr 2017 at 0914 Homs governor says U.S...	nna	4/7/2017	homs	0
2	1914947533	Death toll from Aleppo bomb attack at least 112	Sun 16 Apr 2017 Death toll from Aleppo bomb at...	nna	4/16/2017	aleppo	0
3	1914947534	Aleppo bomb blast kills six Syrian state TV	Wed 19 Apr 2017 Aleppo bomb blast kills six Sy...	nna	4/19/2017	aleppo	0
4	1914947535	29 Syria Rebels Dead in Fighting for Key Alepp...	Sun 10 Jul 2016 29 Syria Rebels Dead in Fighti...	nna	7/10/2016	aleppo	0

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

```
news.shape
news.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 804 entries, 0 to 803
Data columns (total 7 columns):
 #   Column          Non-Null Count  Dtype 
---  ------          --------------  ----- 
 0   unit_id         804 non-null    int64 
 1   article_title   804 non-null    object
 2   article_content 804 non-null    object
 3   source          804 non-null    object
 4   date            804 non-null    object
 5   location        804 non-null    object
 6   labels          804 non-null    int64 
dtypes: int64(2), object(5)
memory usage: 44.1+ KB
```

```
duplicate_news=news[news.duplicated()].sum()
print(duplicate_news)
```

```
unit_id                                                3852958098
article_title      6 citizens killed 22 injured in terrorist atta...
article_content    18 November 2016 6 citizens killed 22 injured ...
source                                                   sanasana
date                                          11/18/201611/18/2016
location                                           aleppoaleppo
labels                                                        0
dtype: object
```

```
news.drop_duplicates(keep=False, inplace=True)
duplicate_news = news[news.duplicated()]
print(duplicate_news)
```

```
Empty DataFrame
Columns: [unit_id, article_title, article_content, source, date, location, labels]
Index: []
```

```
news.drop(['unit_id'], axis=1, inplace=True)
```

```
news['article']=news.article_title+news.article_content
```

```
news.drop(['article_title', 'article_content'], axis=1, inplace=True)
```

```
def cleaned_data(mess):
    nonpunc = [char for char in mess if char not in punctuation]
    nonpunc = ''.join(nonpunc)
    return ' '.join(word for word in nonpunc.split() if word.lower() not in stop)
```

```
news['cleaned_msg'] = news.article.apply(cleaned_data)
news.head()
```

	source	date	location	labels	article	cleaned_msg
0	nna	4/5/2017	idlib	0	Syria attack symptoms consistent with nerve ag...	Syria attack symptoms consistent nerve agent u...
1	nna	4/7/2017	homs	0	Homs governor says U.S. attack caused deaths b...	Homs governor says US attack caused deaths doe...
2	nna	4/16/2017	aleppo	0	Death toll from Aleppo bomb attack at least 11...	Death toll Aleppo bomb attack least 112Sun 16 ...
[illegible]					Aleppo bomb blast kills six Syrian state [illegible]	

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

```
import nltk
nltk.download('stopwords')
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
True
```

```
import string
punctuation = list(string.punctuation)
from nltk.corpus import stopwords
stop = set(stopwords.words('english'))
print(punctuation)
```

```
['!', '"', '#', '$', '%', '&', "'", '(', ')', '*', '+', ',', '-', '.', '/', ':', ';', '<', '=', '>', '?', '@', '[',
```

```
news.to_csv('cleaned_msg.csv', index=False)
news.head()
```

	source	date	location	labels	article	cleaned_msg
0	nna	4/5/2017	idlib	0	Syria attack symptoms consistent with nerve ag...	Syria attack symptoms consistent nerve agent u...
1	nna	4/7/2017	homs	0	Homs governor says U.S. attack caused deaths b...	Homs governor says US attack caused deaths doe...
2	nna	4/16/2017	aleppo	0	Death toll from Aleppo bomb attack at least 11...	Death toll Aleppo bomb attack least 112Sun 16 ...
[illegible]					Aleppo bomb blast kills six Syrian state [illegible]	

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

```
pd.to_datetime(news['date'])
news.head()
```

	source	date	location	labels	article	cleaned_msg
0	nna	4/5/2017	idlib	0	Syria attack symptoms consistent with nerve ag...	Syria attack symptoms consistent nerve agent u...
1	nna	4/7/2017	homs	0	Homs governor says U.S. attack caused deaths b...	Homs governor says US attack caused deaths doe...
2	nna	4/16/2017	aleppo	0	Death toll from Aleppo bomb attack at least 11...	Death toll Aleppo bomb attack least 112Sun 16 ...
3	nna	4/19/2017	aleppo	0	Aleppo bomb blast kills six Syrian state TVWed...	Aleppo bomb blast kills six Syrian state TVWed...

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

```
from nltk.sentiment import SentimentIntensityAnalyzer

data = ['cource', 'date', 'location', 'labels', 'artcile', 'cleaned_msg']
df = pd.DataFrame(data)

import nltk
nltk.download('vader_lexicon')

from nltk.sentiment import SentimentIntensityAnalyzer

sia = SentimentIntensityAnalyzer()
```

```
[nltk_data] Downloading package vader_lexicon to /root/nltk_data...
```

```
def get_sentiment(text):
    scores = sia.polarity_scores(text)
    compound_score = scores['compound']
    if compound_score >= 0.05:
        return 'positive'
    elif compound_score <= -0.05:
        return 'negative'
    else:
        return 'neutral'

news['sentiment'] = news['cleaned_msg'].apply(get_sentiment)

news['sentiment'].value_counts()
```

sentiment	count
negative	797
positive	3
neutral	1

dtype: int64

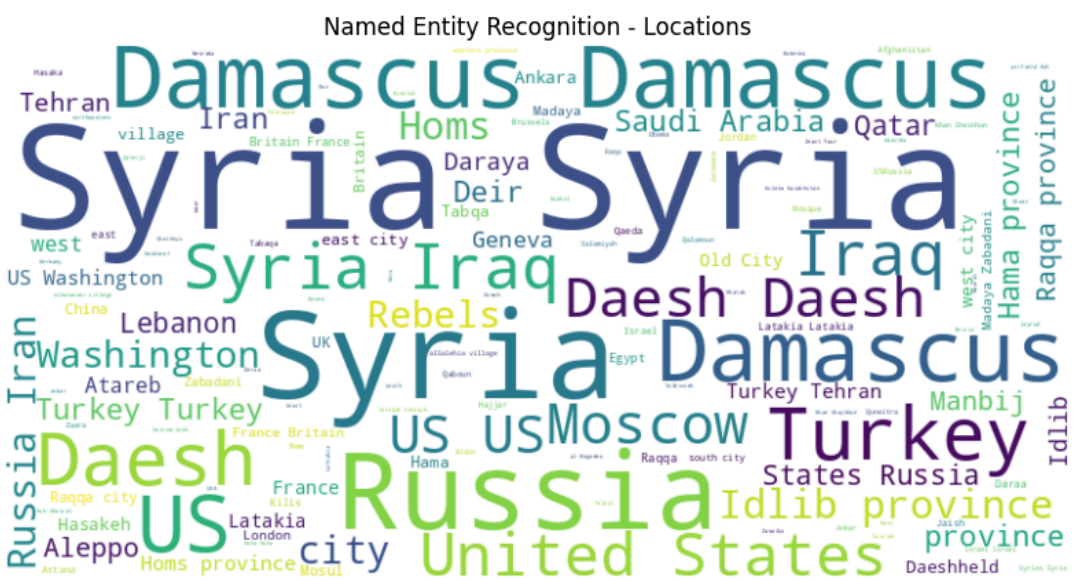
```
news.head()
```

	source	date	location	labels	article	cleaned_msg	sentiment
0	nna	4/5/2017	idlib	0	Syria attack symptoms consistent with nerve ag...	Syria attack symptoms consistent nerve agent u...	negative
1	nna	4/7/2017	homs	0	Homs governor says U.S. attack caused deaths b...	Homs governor says US attack caused deaths doe...	negative
2	nna	4/16/2017	aleppo	0	Death toll from Aleppo bomb attack at least 11...	Death toll Aleppo bomb attack least 112Sun 16 ...	negative
3	nna	4/19/2017	aleppo	0	Aleppo bomb blast kills six Syrian state TVWed...	Aleppo bomb blast kills six Syrian state TVWed...	negative
4	nna	7/10/2016	aleppo	0	29 Syria Rebels Dead in Fighting for Key Alepp...	29 Syria Rebels Dead Fighting Key Aleppo RoadS...	negative

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

```
import matplotlib.pyplot as plt
import seaborn as sns
```

```
import spacy
from wordcloud import WordCloud


nlp = spacy.load("en_core_web_sm")


data = ['cource', 'date', 'location', 'labels', 'artcile', 'cleaned_msg']


df = pd.DataFrame(data)
location_entities = []
for doc in nlp.pipe(news['cleaned_msg']):
    location_entities.extend([ent.text for ent in doc.ents if ent.label_ == 'GPE'])


wordcloud = WordCloud(width=800, height=400, background_color='white').generate(' '.join(location_entities))
plt.figure(figsize=(10, 6))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.title('Named Entity Recognition - Locations')
plt.show()
```

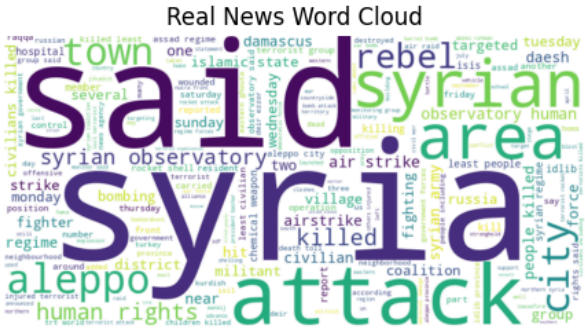



```
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize


fake_news = news[news['labels'] == 0]['article'].str.cat(sep=' ')
real_news = news[news['labels'] == 1]['article'].str.cat(sep=' ')


import nltk
nltk.download('punkt_tab') # Download the 'punkt_tab' data package
```

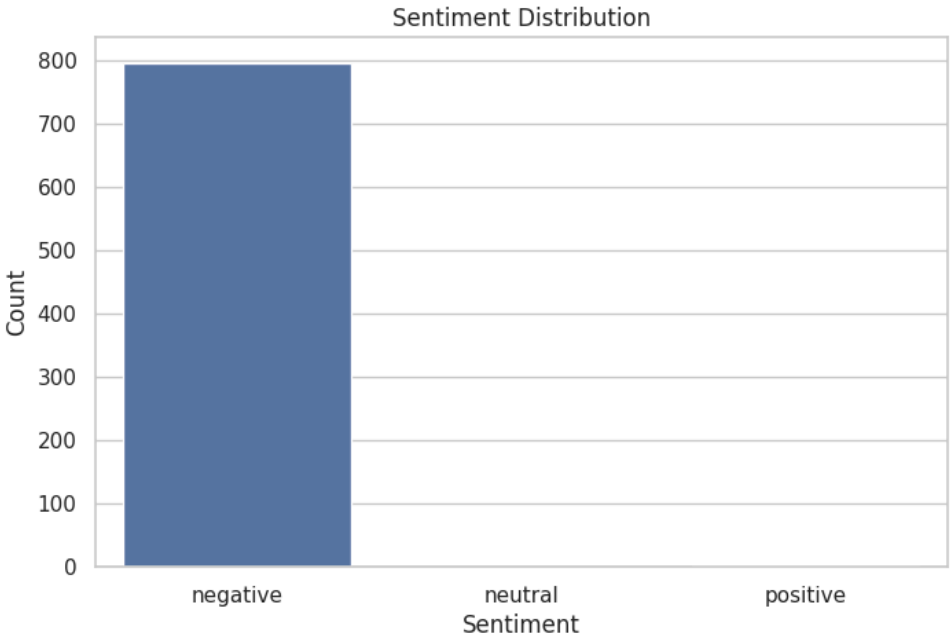
```
[nltk_data] Downloading package punkt_tab to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt_tab.zip.
True
```

```
stop_words = set(stopwords.words('english'))
fake_tokens = word_tokenize(fake_news)
real_tokens = word_tokenize(real_news)
fake_tokens = [word.lower() for word in fake_tokens if word.isalpha() and word.lower() not in stop_words]
real_tokens = [word.lower() for word in real_tokens if word.isalpha() and word.lower() not in stop_words]


fake_wordcloud = WordCloud(width=800, height=400, background_color='white').generate(' '.join(fake_tokens))
real_wordcloud = WordCloud(width=800, height=400, background_color='white').generate(' '.join(real_tokens))


plt.figure(figsize=(12, 6))

plt.subplot(1, 2, 1)
plt.imshow(fake_wordcloud, interpolation='bilinear')
plt.title('Fake News Word Cloud')
plt.axis('off')

plt.subplot(1, 2, 2)
plt.imshow(real_wordcloud, interpolation='bilinear')
```

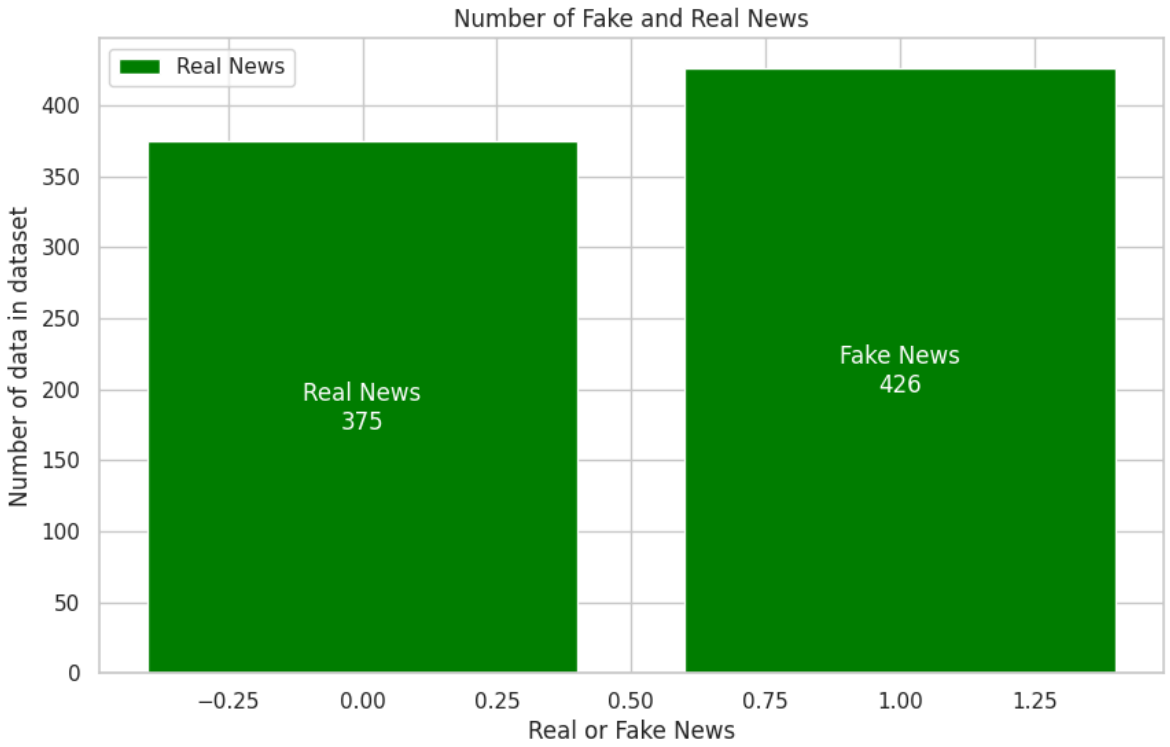
```
plt.title('Real News Word Cloud')
plt.axis('off')

plt.show()
```



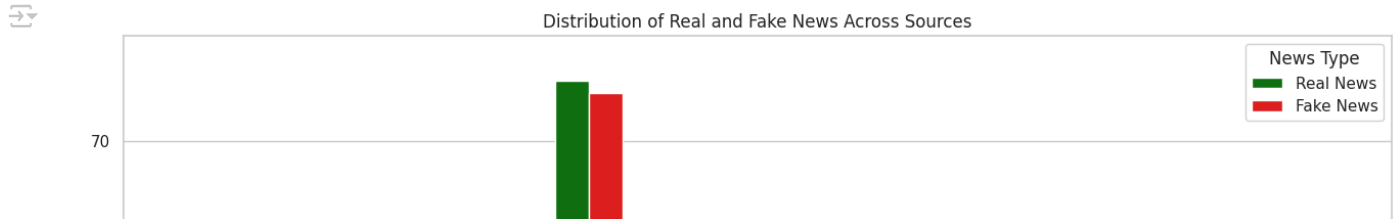

```
sns.set(style="whitegrid")
plt.figure(figsize=(8, 5))
sns.countplot(x="sentiment", data=news)
plt.title("Sentiment Distribution")
plt.xlabel("Sentiment")
plt.ylabel("Count")
plt.show()
```




```
real_count = (news.labels == 0).sum()
fake_count = (news.labels == 1).sum()

plt.figure(figsize=(10, 6))
plt.hist(news.labels, color='green', bins=[-0.5, 0.5, 1.5], rwidth=0.8, align='mid')

plt.text(0, real_count // 2, f'Real News\n{real_count}', ha='center', va='center', color='white', fontsize=12)
plt.text(1, fake_count // 2, f'Fake News\n{fake_count}', ha='center', va='center', color='white', fontsize=12)

plt.ylabel('Number of data in dataset')
plt.xlabel('Real or Fake News')
plt.title('Number of Fake and Real News')

plt.legend(['Real News', 'Fake News'])
plt.show()
```




```
plt.figure(figsize=(16, 12))
palette = {0: 'green', 1: 'red'}
sns.countplot(x='source', hue='labels', data=news, palette=palette)
plt.xlabel('Source')
plt.ylabel('Count')
plt.title('Distribution of Real and Fake News Across Sources')

plt.legend(['Real News', 'Fake News'], title='News Type')
plt.xticks(rotation=45, ha='right')
plt.show()
```




```
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
```

```
X_train, X_test, y_train, y_test = train_test_split(news['cleaned_msg'], news['labels'], test_size=0.2, random_state=42)

# Initialize the CountVectorizer to convert text data into a bag-of-words representation
vectorizer = CountVectorizer()
X_train_bow = vectorizer.fit_transform(X_train)
X_test_bow = vectorizer.transform(X_test)

# Initialize the Random Forest classifier
rf_classifier = RandomForestClassifier(n_estimators=100, random_state=42)

# Train the classifier
rf_classifier.fit(X_train_bow, y_train)

# Make predictions on the test set
predictions = rf_classifier.predict(X_test_bow)

# Evaluate the model
accuracy = accuracy_score(y_test, predictions)
classification_rep = classification_report(y_test, predictions)
conf_matrix = confusion_matrix(y_test, predictions)

# Display the evaluation metrics
print(f"Accuracy: {accuracy}")
print("\nClassification Report:\n", classification_rep)
print("\nConfusion Matrix:\n", conf_matrix)
```

```
Accuracy: 0.546583850931677

Classification Report:
              precision    recall  f1-score   support

           0       0.45      0.34      0.39        68
           1       0.59      0.70      0.64        93

    accuracy                           0.55       161
   macro avg       0.52      0.52      0.51       161
weighted avg       0.53      0.55      0.53       161


Confusion Matrix:
 [[23 45]
 [28 65]]
```