```python
import pandas as pd
import numpy as np
import re
import string
import nltk
import matplotlib.pyplot as plt
import seaborn as sns

from nltk.corpus import stopwords
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split
from sklearn.svm import LinearSVC
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix

nltk.download('stopwords')
stop_words = set(stopwords.words('english'))
```

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

```python
df = pd.read_csv("news_dataset.csv")  # Make sure this file exists
print("Dataset shape:", df.shape)
df = df[['text', 'label']].dropna()

```

```
Dataset shape: (3729, 2)
```

```python
df.head()
```

	text	label
0	Payal has accused filmmaker Anurag Kashyap of ...	REAL
1	A four-minute-long video of a woman criticisin...	FAKE
2	Republic Poll, a fake Twitter account imitatin...	FAKE
3	Delhi teen finds place on UN green list, turns...	REAL
4	Delhi: A high-level meeting underway at reside...	REAL

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

```python
def preprocess_text(text):
    text = str(text).lower()  # lowercase
    text = re.sub(r"http\S+|www\S+|https\S+", '', text)  # remove links
    text = re.sub(r"[^\w\s]", '', text)  # remove punctuation
    text = re.sub(r"\d+", '', text)  # remove numbers
```

```
    text = ' '.join([word for word in text.split() if word not in stop_words])
    return text

```

```
df['cleaned_text'] = df['text'].apply(preprocess_text)
print("\nSample cleaned text:\n", df['cleaned_text'].head())
```

```
Sample cleaned text:
 0    payal accused filmmaker anurag kashyap behavin...
1    fourminutelong video woman criticising governm...
2    republic poll fake twitter account imitating a...
3    delhi teen finds place un green list turns gla...
4    delhi highlevel meeting underway residence raj...
Name: cleaned_text, dtype: object
```

```
tfidf = TfidfVectorizer(max_features=5000)
X = tfidf.fit_transform(df['cleaned_text'])
y = df['label']
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

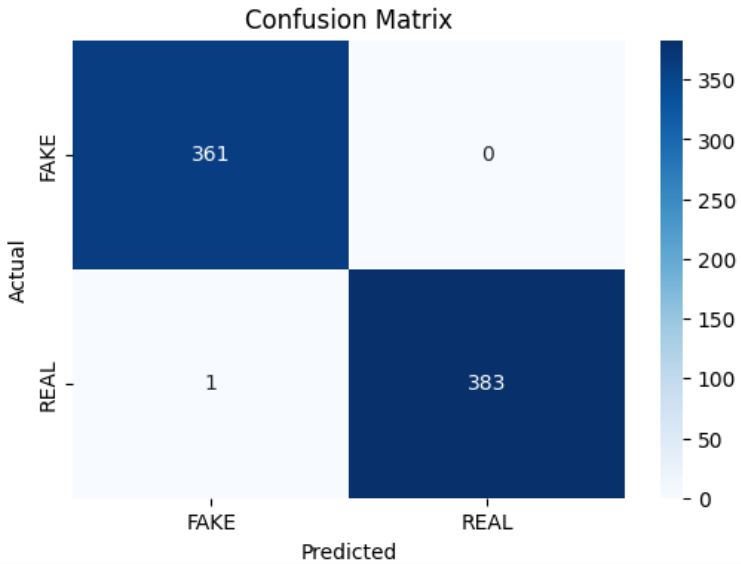
```
model = LinearSVC()
model.fit(X_train, y_train)
```

```
▾ LinearSVC
LinearSVC()
```

```
y_pred = model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print("\n model accuracy:", accuracy)
print("\n classification report:\n", classification_report(y_test, y_pred))
```

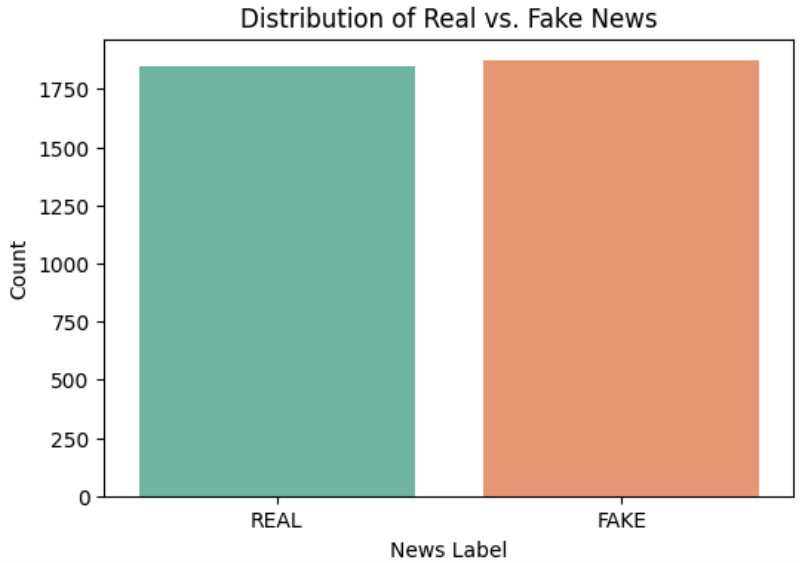
```
 model accuracy: 0.9986577181208054

 classification report:
               precision    recall  f1-score   support

        FAKE       1.00      1.00      1.00       361
        REAL       1.00      1.00      1.00       384

    accuracy                           1.00       745
   macro avg       1.00      1.00      1.00       745
weighted avg       1.00      1.00      1.00       745
```

```
cm = confusion_matrix(y_test, y_pred)
plt.figure(figsize=(6,4))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', xticklabels=['FAKE', 'REAL'], yticklabels=['FAKE', 'REAL'])
plt.title('Confusion Matrix')
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.show()
```

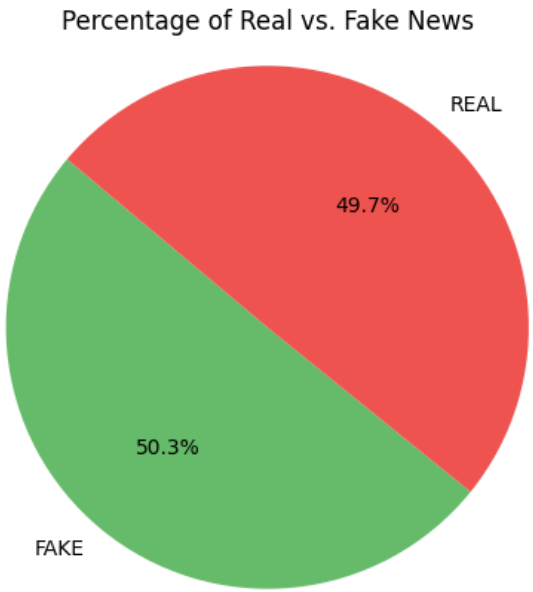



```
import matplotlib.pyplot as plt
import seaborn as sns

# Count plot
plt.figure(figsize=(6,4))
sns.countplot(x='label', data=df, palette='Set2')
plt.title('Distribution of Real vs. Fake News')
plt.xlabel('News Label')
plt.ylabel('Count')
plt.show()

```

```
/tmp/ipython-input-11-1299514214.py:6: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

  sns.countplot(x='label', data=df, palette='Set2')
```




```
# Pie chart
label_counts = df['label'].value_counts()
plt.figure(figsize=(5,5))
plt.pie(label_counts, labels=label_counts.index, autopct='%1.1f%%', startangle=140, colors=['#66bb6a', '#ef5350'])
plt.title('Percentage of Real vs. Fake News')
plt.axis('equal')
plt.show()

```


Percentage of Real vs. Fake News
REAL
49.7%
50.3%
FAKE