In [19]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib
import seaborn as sns
import itertools
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.linear_model import PassiveAggressiveClassifier
from sklearn.metrics import accuracy_score, confusion_matrix
```

In [20]:
```
#default theme
plt.style.use('ggplot')
sns.color_palette("tab10")
sns.set(context='notebook', style='darkgrid', font='sans-serif', font_scale=1, rc=None)
matplotlib.rcParams['figure.figsize'] =[20,8]
matplotlib.rcParams.update({'font.size': 15})
matplotlib.rcParams['font.family'] = 'sans-serif'
```

In [21]:
```
#Read the data
df=pd.read_csv('fake_or_real_news.csv')

#Get shape and head
print(df.shape)
df.head()
```

```
(6335, 4)
```

Out[21]:

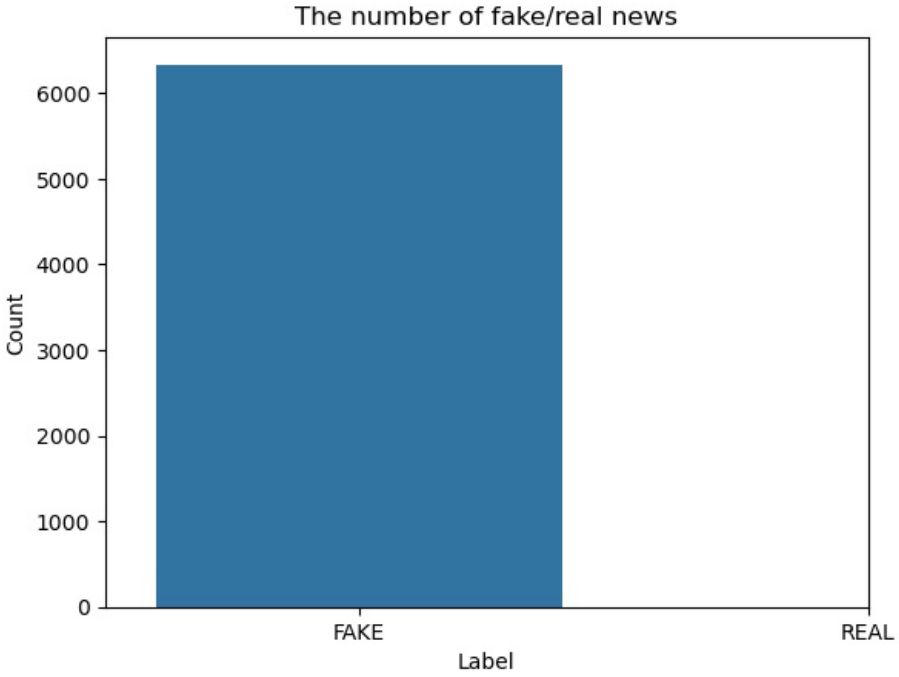
	Unnamed: 0	title	text	label
0	8476	You Can Smell Hillary's Fear	Daniel Greenfield, a Shillman Journalism Fello...	FAKE
1	10294	Watch The Exact Moment Paul Ryan Committed Pol...	Google Pinterest Digg Linkedin Reddit Stumbleu...	FAKE
2	3608	Kerry to go to Paris in gesture of sympathy	U.S. Secretary of State John F. Kerry said Mon...	REAL
3	10142	Bernie supporters on Twitter erupt in anger ag...	— Kaydee King (@KaydeeKing) November 9, 2016 T...	FAKE
4	875	The Battle of New York: Why This Primary Matters	It's primary day in New York and front-runners...	REAL

In [22]:
```
#DataFlair - Get the labels
labels=df.label
labels.head()
```

Out[22]:
```
0    FAKE
1    FAKE
2    REAL
3    FAKE
4    REAL
Name: label, dtype: object
```

In [19]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.preprocessing import LabelEncoder

# Read the CSV file into a DataFrame
df = pd.read_csv('fake_or_real_news.csv')

# Print unique values in the 'label' column before encoding
print("Unique labels before encoding:", df['label'].unique())

# Initialize LabelEncoder
label_encoder = LabelEncoder()

# Encode the 'label' column
df['label_encoded'] = label_encoder.fit_transform(df['label'])

# Print unique values in the encoded 'label' column
print("Unique encoded labels:", df['label_encoded'].unique())

# Plot the count of fake and real news
sns.countplot(df['label_encoded'])
plt.title('The number of fake/real news')
plt.xlabel('Label')
plt.ylabel('Count')
plt.xticks(ticks=[0, 1], labels=label_encoder.classes_)  # Adding label names
plt.show()
```

```
Unique labels before encoding: ['FAKE' 'REAL']
Unique encoded labels: [0 1]
```



In [25]:
```
#DataFlair - Split the dataset
x_train,x_test,y_train,y_test=train_test_split(df['text'], labels, test_size=0.2, random_state=7)

#DataFlair - Initialize a TfidfVectorizer
tfidf_vectorizer=TfidfVectorizer(stop_words='english', max_df=0.7)

#DataFlair - Fit and transform train set, transform test set
tfidf_train=tfidf_vectorizer.fit_transform(x_train)
tfidf_test=tfidf_vectorizer.transform(x_test)

#DataFlair - Initialize a PassiveAggressiveClassifier
pac=PassiveAggressiveClassifier(max_iter=50)
pac.fit(tfidf_train,y_train)

#DataFlair - Predict on the test set and calculate accuracy
y_pred=pac.predict(tfidf_test)
score=accuracy_score(y_test,y_pred)
print(f'Accuracy: {round(score*100,2)}%')
```

```
Accuracy: 92.58%
```

In [23]:
```
target=df.label.value_counts()
target
```

Out[23]:
```
label
REAL    3171
FAKE    3164
Name: count, dtype: int64
```

In [26]:
```
#DataFlair - Build confusion matrix
confusion_matrix(y_test,y_pred, labels=['FAKE','REAL'])
```

Out[26]:
```
array([[586,  52],
       [ 42, 587]], dtype=int64)
```