In [22]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import nltk
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from nltk.stem import PorterStemmer, WordNetLemmatizer
import string

from sklearn.feature_extraction.text import CountVectorizer
import scipy.sparse as sp
import numpy as np
```

In [23]:
```
#importing the csv files
real_news_df = pd.read_csv('True.csv')
fake_news_df = pd.read_csv('Fake.csv')
```

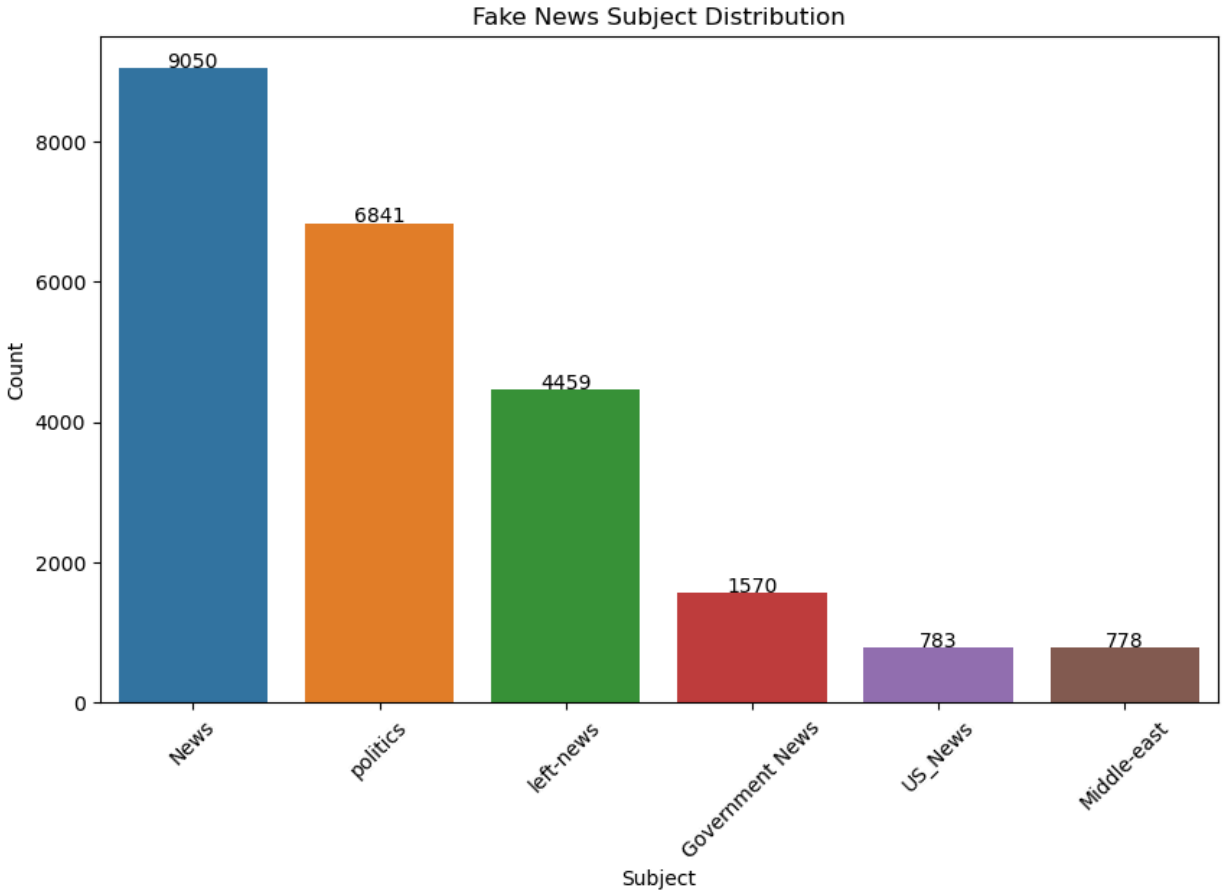
Data Exploration

In [24]:
```
print(real_news_df.isnull().sum())
print(fake_news_df.isnull().sum())
```

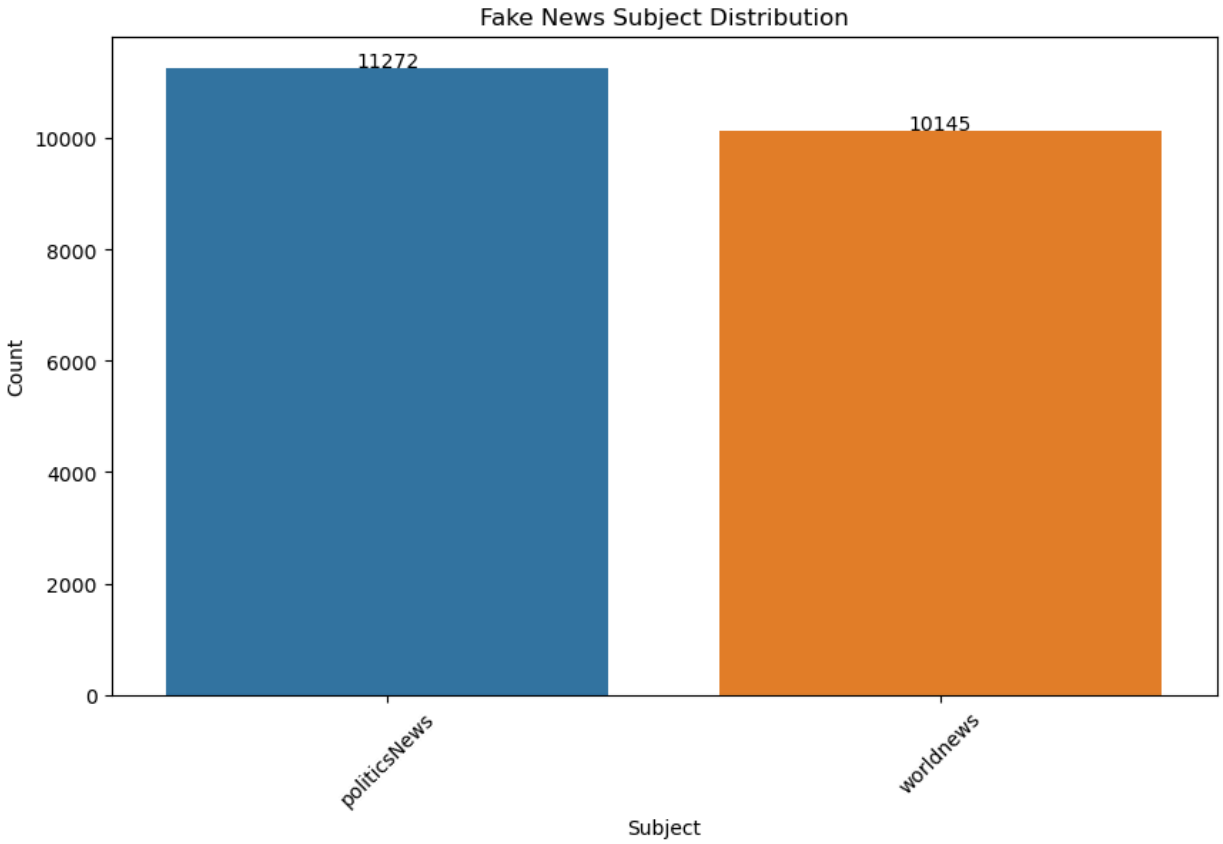
```
title      0
text       0
subject    0
date       0
dtype: int64
title      0
text       0
subject    0
date       0
dtype: int64
```

In [25]:
```
subject_counts = fake_news_df['subject'].value_counts()

plt.figure(figsize=(10, 6))
sns.barplot(x=subject_counts.index, y=subject_counts.values)
plt.title('Fake News Subject Distribution')
plt.xlabel('Subject')
plt.ylabel('Count')
plt.xticks(rotation=45)
for index, value in enumerate(subject_counts):
    plt.text(index, value + 10, str(value), ha='center')
plt.show()
```

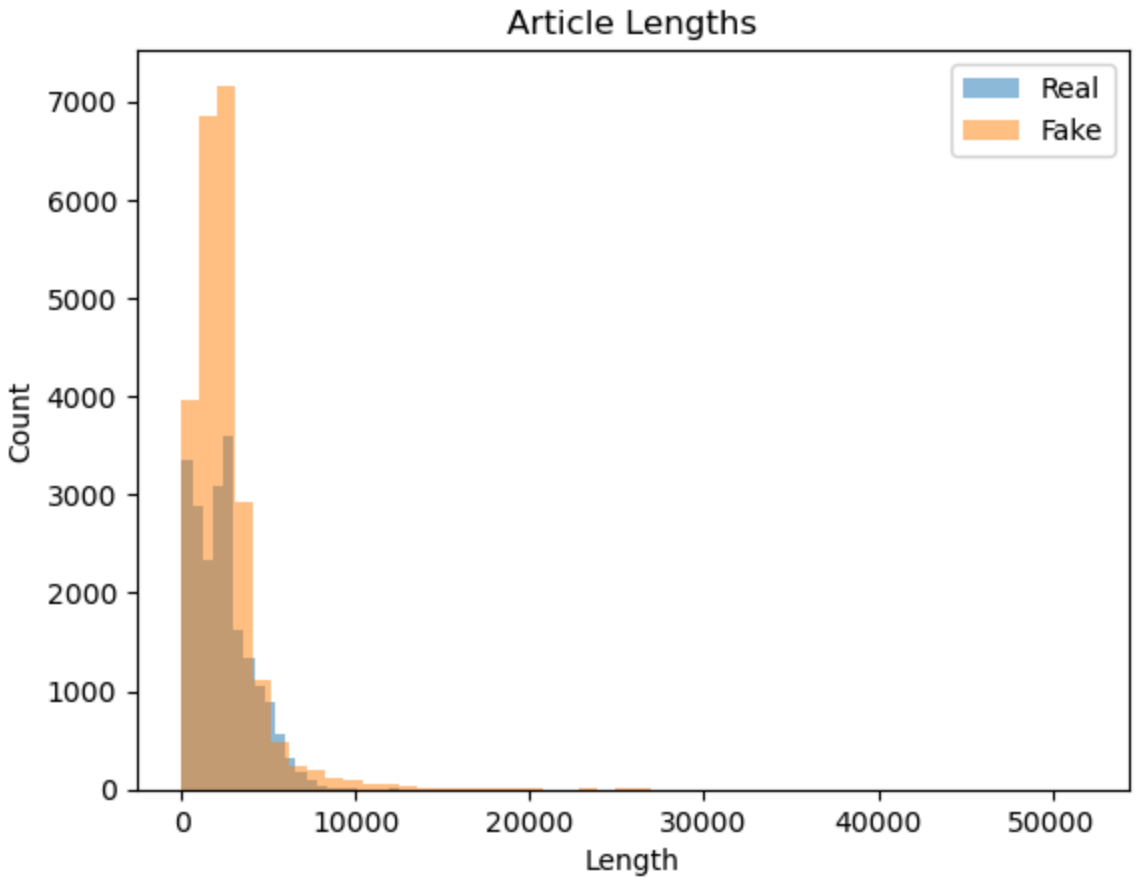


In [26]:
```
subject_counts = real_news_df['subject'].value_counts()
plt.figure(figsize=(10, 6))
sns.barplot(x=subject_counts.index, y=subject_counts.values)
plt.title('Fake News Subject Distribution')
plt.xlabel('Subject')
plt.ylabel('Count')
plt.xticks(rotation=45)
for index, value in enumerate(subject_counts):
    plt.text(index, value + 10, str(value), ha='center')
plt.show()
```



In [27]:
```
real_lengths = real_news_df['text'].apply(len)
fake_lengths = fake_news_df['text'].apply(len)

plt.hist(real_lengths, bins=50, alpha=0.5, label='Real')
plt.hist(fake_lengths, bins=50, alpha=0.5, label='Fake')
plt.title('Article Lengths')
plt.xlabel('Length')
plt.ylabel('Count')
plt.legend()
plt.show()
```



Pre-Processing Steps:

1.Lowercasing the text 2.Removing punctuation and digits 3.Removing stop words 4.Stemming or lemmatizing the text

In [28]:
```
nltk.download('wordnet')

stop_words = set(stopwords.words('english'))
stemmer = PorterStemmer()
lemmatizer = WordNetLemmatizer()

def preprocess_text(text):
    # Lowercase the text
    text = text.lower()

    # Remove punctuation and digits
    text = text.translate(str.maketrans('', '', string.punctuation + string.digits))

    # Tokenize the text
    words = word_tokenize(text)

    # Remove stop words
    words = [word for word in words if word not in stop_words]

    # Stem or lemmatize the words
    words = [stemmer.stem(word) for word in words]

    # Join the words back into a string
```

```
    text = ' '.join(words)

    return text
```

```
[nltk_data] Downloading package wordnet to
[nltk_data]     C:\Users\shema\AppData\Roaming\nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
```

In [29]:
```
real_news_df['text'] = real_news_df['text'].apply(preprocess_text)
fake_news_df['text'] = fake_news_df['text'].apply(preprocess_text)
```

In [30]:
```
vectorizer = CountVectorizer()
X_real = vectorizer.fit_transform(real_news_df['text'])
X_fake = vectorizer.transform(fake_news_df['text'])

X = sp.vstack([X_real, X_fake])
y = np.concatenate([np.ones(X_real.shape[0]), np.zeros(X_fake.shape[0])])
```

In [32]:
```
from sklearn.model_selection import train_test_split

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

In [33]:
```
from sklearn.linear_model import LogisticRegression

clf = LogisticRegression(random_state=42)
clf.fit(X_train, y_train)
```

```
C:\Users\shema\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:814: Con
vergenceWarning: lbfgs failed to converge (status=1):
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(
```

Out[33]:
```
LogisticRegression(random_state=42)
```

In [37]:
```
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score

y_pred = clf.predict(X_test)

accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred)
recall = recall_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred)

print('Accuracy:', accuracy)
print('Precision:', precision)
print('Recall:', recall)
print('F1 Score:', f1)
```

```
Accuracy: 0.994988864142539
Precision: 0.9935498733010827
Recall: 0.9960739030023095
F1 Score: 0.9948102871641102
```

In [40]:
```
from sklearn.naive_bayes import MultinomialNB
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC


models = [
    MultinomialNB(),
    LogisticRegression(),
    DecisionTreeClassifier(),
    RandomForestClassifier(),
    SVC()
]

for model in models:
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)
    accuracy = accuracy_score(y_test, y_pred)
    print(f'{model.__class__.__name__}: {accuracy*100:.2f}')
    print("-"*30)
```

```
MultinomialNB: 94.22
------------------------------
C:\Users\shema\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:814: Con
vergenceWarning: lbfgs failed to converge (status=1):
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(
LogisticRegression: 99.50
------------------------------
DecisionTreeClassifier: 99.52
------------------------------
RandomForestClassifier: 99.24
------------------------------
SVC: 99.19
------------------------------
```

In [41]:
```
from sklearn.model_selection import GridSearchCV

# Define a list of hyperparameters to search over
hyperparameters = {
    'penalty': ['l1', 'l2'],
    'C': [0.1, 1, 10, 100],
    'solver': ['liblinear', 'saga']
}

# Perform grid search to find the best hyperparameters
grid_search = GridSearchCV(LogisticRegression(), hyperparameters, cv=5)
grid_search.fit(X_train, y_train)

# Print the best hyperparameters and test accuracy
```

```
print('Best hyperparameters:', grid_search.best_params_)
print('Test accuracy:', grid_search.score(X_test, y_test))
```

```
C:\Users\shema\anaconda3\lib\site-packages\sklearn\linear_model\_sag.py:352: Converge
nceWarning: The max_iter was reached which means the coef_ did not converge
  warnings.warn(
C:\Users\shema\anaconda3\lib\site-packages\sklearn\linear_model\_sag.py:352: Converge
nceWarning: The max_iter was reached which means the coef_ did not converge
  warnings.warn(
C:\Users\shema\anaconda3\lib\site-packages\sklearn\linear_model\_sag.py:352: Converge
nceWarning: The max_iter was reached which means the coef_ did not converge
  warnings.warn(
C:\Users\shema\anaconda3\lib\site-packages\sklearn\linear_model\_sag.py:352: Converge
nceWarning: The max_iter was reached which means the coef_ did not converge
  warnings.warn(
C:\Users\shema\anaconda3\lib\site-packages\sklearn\linear_model\_sag.py:352: Converge
nceWarning: The max_iter was reached which means the coef_ did not converge
  warnings.warn(
C:\Users\shema\anaconda3\lib\site-packages\sklearn\linear_model\_sag.py:352: Converge
nceWarning: The max_iter was reached which means the coef_ did not converge
  warnings.warn(
C:\Users\shema\anaconda3\lib\site-packages\sklearn\linear_model\_sag.py:352: Converge
nceWarning: The max_iter was reached which means the coef_ did not converge
  warnings.warn(
C:\Users\shema\anaconda3\lib\site-packages\sklearn\linear_model\_sag.py:352: Converge
nceWarning: The max_iter was reached which means the coef_ did not converge
  warnings.warn(
C:\Users\shema\anaconda3\lib\site-packages\sklearn\linear_model\_sag.py:352: Converge
nceWarning: The max_iter was reached which means the coef_ did not converge
  warnings.warn(
C:\Users\shema\anaconda3\lib\site-packages\sklearn\linear_model\_sag.py:352: Converge
nceWarning: The max_iter was reached which means the coef_ did not converge
  warnings.warn(
C:\Users\shema\anaconda3\lib\site-packages\sklearn\linear_model\_sag.py:352: Converge
nceWarning: The max_iter was reached which means the coef_ did not converge
  warnings.warn(
C:\Users\shema\anaconda3\lib\site-packages\sklearn\linear_model\_sag.py:352: Converge
nceWarning: The max_iter was reached which means the coef_ did not converge
  warnings.warn(
C:\Users\shema\anaconda3\lib\site-packages\sklearn\linear_model\_sag.py:352: Converge
nceWarning: The max_iter was reached which means the coef_ did not converge
  warnings.warn(
C:\Users\shema\anaconda3\lib\site-packages\sklearn\linear_model\_sag.py:352: Converge
nceWarning: The max_iter was reached which means the coef_ did not converge
  warnings.warn(
C:\Users\shema\anaconda3\lib\site-packages\sklearn\linear_model\_sag.py:352: Converge
nceWarning: The max_iter was reached which means the coef_ did not converge
  warnings.warn(
C:\Users\shema\anaconda3\lib\site-packages\sklearn\linear_model\_sag.py:352: Converge
nceWarning: The max_iter was reached which means the coef_ did not converge
  warnings.warn(
C:\Users\shema\anaconda3\lib\site-packages\sklearn\linear_model\_sag.py:352: Converge
nceWarning: The max_iter was reached which means the coef_ did not converge
  warnings.warn(
C:\Users\shema\anaconda3\lib\site-packages\sklearn\linear_model\_sag.py:352: Converge
nceWarning: The max_iter was reached which means the coef_ did not converge
  warnings.warn(
C:\Users\shema\anaconda3\lib\site-packages\sklearn\linear_model\_sag.py:352: Converge
nceWarning: The max_iter was reached which means the coef_ did not converge
  warnings.warn(
C:\Users\shema\anaconda3\lib\site-packages\sklearn\linear_model\_sag.py:352: Converge
nceWarning: The max_iter was reached which means the coef_ did not converge
  warnings.warn(
```

```
C:\Users\shema\anaconda3\lib\site-packages\sklearn\linear_model\_sag.py:352: Converge
nceWarning: The max_iter was reached which means the coef_ did not converge
  warnings.warn(
C:\Users\shema\anaconda3\lib\site-packages\sklearn\linear_model\_sag.py:352: Converge
nceWarning: The max_iter was reached which means the coef_ did not converge
  warnings.warn(
C:\Users\shema\anaconda3\lib\site-packages\sklearn\linear_model\_sag.py:352: Converge
nceWarning: The max_iter was reached which means the coef_ did not converge
  warnings.warn(
C:\Users\shema\anaconda3\lib\site-packages\sklearn\linear_model\_sag.py:352: Converge
nceWarning: The max_iter was reached which means the coef_ did not converge
  warnings.warn(
C:\Users\shema\anaconda3\lib\site-packages\sklearn\linear_model\_sag.py:352: Converge
nceWarning: The max_iter was reached which means the coef_ did not converge
  warnings.warn(
C:\Users\shema\anaconda3\lib\site-packages\sklearn\linear_model\_sag.py:352: Converge
nceWarning: The max_iter was reached which means the coef_ did not converge
  warnings.warn(
C:\Users\shema\anaconda3\lib\site-packages\sklearn\linear_model\_sag.py:352: Converge
nceWarning: The max_iter was reached which means the coef_ did not converge
  warnings.warn(
C:\Users\shema\anaconda3\lib\site-packages\sklearn\linear_model\_sag.py:352: Converge
nceWarning: The max_iter was reached which means the coef_ did not converge
  warnings.warn(
C:\Users\shema\anaconda3\lib\site-packages\sklearn\linear_model\_sag.py:352: Converge
nceWarning: The max_iter was reached which means the coef_ did not converge
  warnings.warn(
C:\Users\shema\anaconda3\lib\site-packages\sklearn\linear_model\_sag.py:352: Converge
nceWarning: The max_iter was reached which means the coef_ did not converge
  warnings.warn(
C:\Users\shema\anaconda3\lib\site-packages\sklearn\linear_model\_sag.py:352: Converge
nceWarning: The max_iter was reached which means the coef_ did not converge
  warnings.warn(
C:\Users\shema\anaconda3\lib\site-packages\sklearn\linear_model\_sag.py:352: Converge
nceWarning: The max_iter was reached which means the coef_ did not converge
  warnings.warn(
C:\Users\shema\anaconda3\lib\site-packages\sklearn\linear_model\_sag.py:352: Converge
nceWarning: The max_iter was reached which means the coef_ did not converge
  warnings.warn(
C:\Users\shema\anaconda3\lib\site-packages\sklearn\linear_model\_sag.py:352: Converge
nceWarning: The max_iter was reached which means the coef_ did not converge
  warnings.warn(
C:\Users\shema\anaconda3\lib\site-packages\sklearn\linear_model\_sag.py:352: Converge
nceWarning: The max_iter was reached which means the coef_ did not converge
  warnings.warn(
C:\Users\shema\anaconda3\lib\site-packages\sklearn\linear_model\_sag.py:352: Converge
nceWarning: The max_iter was reached which means the coef_ did not converge
  warnings.warn(
C:\Users\shema\anaconda3\lib\site-packages\sklearn\linear_model\_sag.py:352: Converge
nceWarning: The max_iter was reached which means the coef_ did not converge
  warnings.warn(
C:\Users\shema\anaconda3\lib\site-packages\sklearn\linear_model\_sag.py:352: Converge
nceWarning: The max_iter was reached which means the coef_ did not converge
  warnings.warn(
C:\Users\shema\anaconda3\lib\site-packages\sklearn\linear_model\_sag.py:352: Converge
nceWarning: The max_iter was reached which means the coef_ did not converge
  warnings.warn(
C:\Users\shema\anaconda3\lib\site-packages\sklearn\linear_model\_sag.py:352: Converge
nceWarning: The max_iter was reached which means the coef_ did not converge
  warnings.warn(
```

```
Best hyperparameters: {'C': 10, 'penalty': 'l2', 'solver': 'liblinear'}
Test accuracy: 0.99543429844098
```

In [ ]:
```
tfidf_v = TfidfVectorizer()
tfidf_X_train = tfidf_v.fit_transform(X_train)
tfidf_X_test = tfidf_v.transform(X_test)
```