In [ ]:
```
import numpy as np
from sklearn.svm import LinearSVC
from sklearn import datasets
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix, accuracy_score
%matplotlib inline
from sklearn.feature_extraction.text import TfidfVectorizer
```

In [ ]:
```
import warnings #to remove the warnings
warnings.filterwarnings('ignore')
```

In [ ]:
```
df= pd.read_csv('F:/GAIP/Research project/ScAN/fake_or_real_news.csv') #read the datas
#Explore the dataset

df.head()
df.describe() #To know more about the dataset

df
```

Out[ ]:

	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
...	...	...	...	...
6330	4490	State Department says it can't find emails fro...	The State Department told the Republican Natio...	REAL
6331	8062	The 'P' in PBS Should Stand for 'Plutocratic' ...	The 'P' in PBS Should Stand for 'Plutocratic' ...	FAKE
6332	8622	Anti-Trump Protesters Are Tools of the Oligarc...	Anti-Trump Protesters Are Tools of the Oligar...	FAKE
6333	4021	In Ethiopia, Obama seeks progress on peace, se...	ADDIS ABABA, Ethiopia —President Obama convene...	REAL
6334	4330	Jeb Bush Is Suddenly Attacking Trump. Here's W...	Jeb Bush Is Suddenly Attacking Trump. Here's W...	REAL

6335 rows × 4 columns

In [ ]:
```
labels = df.label
labels.head()
```

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

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

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

In [ ]:
```
x_train,x_test,y_train,y_test=train_test_split(df['text'], labels, test_size=0.2, ranc
```

In [ ]:
```
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)
```

In [ ]:
```
modelLogistic = LogisticRegression()
modelLogistic.fit(tfidf_train,y_train)
```

Out[ ]:
```
LogisticRegression()
```

In [ ]:
```
y_pred=modelLogistic.predict(tfidf_test)
score=accuracy_score(y_test,y_pred)
print(f'Accuracy: {round(score*100,2)}%')
```

```
Accuracy: 91.71%
```

In [ ]:
```
confusion_matrix(y_test,y_pred, labels=['FAKE','REAL'])
```

Out[ ]:
```
array([[600,  38],
       [ 67, 562]], dtype=int64)
```

In [ ]:
```
modelSVC = LinearSVC()
modelSVC.fit(tfidf_train,y_train)
```

Out[ ]:
```
LinearSVC()
```

In [ ]:
```
y_pred=modelSVC.predict(tfidf_test)
score=accuracy_score(y_test,y_pred)
print(f'Accuracy: {round(score*100,2)}%')
```

```
Accuracy: 93.21%
```