In [2]:
```
import pandas as pd
from matplotlib import pyplot as plt
%matplotlib inline
import seaborn as sns
import re
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfTransformer
from sklearn.svm import SVC
from sklearn.model_selection import train_test_split , GridSearchCV
from sklearn.metrics import accuracy_score , ConfusionMatrixDisplay , classification_r
```

In [3]:
```
df = pd.read_csv('train.csv')
```

In [4]:
```
df.tail()
```

Out[4]:

	id	title	author	text	label
20795	20795	Rapper T.I.: Trump a 'Poster Child For White S...	Jerome Hudson	Rapper T. I. unloaded on black celebrities who...	0
20796	20796	N.F.L. Playoffs: Schedule, Matchups and Odds -...	Benjamin Hoffman	When the Green Bay Packers lost to the Washing...	0
20797	20797	Macy's Is Said to Receive Takeover Approach by...	Michael J. de la Merced and Rachel Abrams	The Macy's of today grew from the union of sev...	0
20798	20798	NATO, Russia To Hold Parallel Exercises In Bal...	Alex Ansary	NATO, Russia To Hold Parallel Exercises In Bal...	1
20799	20799	What Keeps the F-35 Alive	David Swanson	David Swanson is an author, activist, journa...	1

In [5]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20800 entries, 0 to 20799
Data columns (total 5 columns):
 #   Column  Non-Null Count  Dtype 
---  ------  --------------  ----- 
 0   id      20800 non-null  int64 
 1   title   20242 non-null  object
 2   author  18843 non-null  object
 3   text    20761 non-null  object
 4   label   20800 non-null  int64 
dtypes: int64(2), object(3)
memory usage: 812.6+ KB
```

In [6]:
```
df.describe()
```

Out[6]:

	id	label
count	20800.000000	20800.000000
mean	10399.500000	0.500625
std	6004.587135	0.500012
min	0.000000	0.000000
25%	5199.750000	0.000000
50%	10399.500000	1.000000
75%	15599.250000	1.000000
max	20799.000000	1.000000

In [7]:
```
df.isnull().sum()
```

Out[7]:
```
id           0
title      558
author    1957
text        39
label        0
dtype: int64
```

In [8]:
```
df.duplicated().sum()
```

Out[8]:
```
0
```

In [9]:
```
unique_values = df['label'].unique()
unique_count = len(unique_values)
print(unique_count)
```

```
2
```

In [10]:
```
value_counts = df['label'].value_counts()
for value, count in value_counts.items():
    print(f"{value}: {count}")
```

```
1: 10413
0: 10387
```

In [11]:
```
df['label'].isnull().sum()
```

Out[11]:
```
0
```

In [12]:
```
df['label']
```

Out[12]:
```
0        1
1        0
2        1
3        1
4        1
        ..
20795    0
20796    0
20797    0
20798    1
20799    1
Name: label, Length: 20800, dtype: int64
```

In [13]:
```
df.fillna(" ", inplace= True)
df['content'] = df['title'] + " " + df['author'] + " " + df['text']
df.head()
```

Out[13]:

	id	title	author	text	label	content
0	0	House Dem Aide: We Didn't Even See Comey's Let...	Darrell Lucus	House Dem Aide: We Didn't Even See Comey's Let...	1	House Dem Aide: We Didn't Even See Comey's Let...
1	1	FLYNN: Hillary Clinton, Big Woman on Campus - ...	Daniel J. Flynn	Ever get the feeling your life circles the rou...	0	FLYNN: Hillary Clinton, Big Woman on Campus - ...
2	2	Why the Truth Might Get You Fired	Consortiumnews.com	Why the Truth Might Get You Fired October 29, ...	1	Why the Truth Might Get You Fired Consortiumne...
3	3	15 Civilians Killed In Single US Airstrike Hav...	Jessica Purkiss	Videos 15 Civilians Killed In Single US Airstr...	1	15 Civilians Killed In Single US Airstrike Hav...
4	4	Iranian woman jailed for fictional unpublished...	Howard Portnoy	Print \nAn Iranian woman has been sentenced to...	1	Iranian woman jailed for fictional unpublished...

In [14]:
```
port_stem = PorterStemmer()
def stemming(content):
    #replace any non-alphabetic characters in the content variable with a space charac
    stemmed_content= re.sub('[^a-zA-Z]',' ',content)
    #Convert all words into lower case letters
    stemmed_content = stemmed_content.lower()
    # Split the words into list
    stemmed_content = stemmed_content.split()
    #generate a list of stemmed words from stemmed_content, excluding any stop words f
    stemmed_content = [port_stem.stem(word) for word in stemmed_content if not word ir
    #Join the elements from the list 'stemmed_content' into a single string separated
    stemmed_content = " ".join(stemmed_content)
    return stemmed_content
```

In [16]:
```
df['content']= df['content'].apply(stemming)
df['content']
```

Out[16]:
```
0        hous dem aid even see comey letter jason chaff...
1        flynn hillari clinton big woman campu breitbar...
2        truth might get fire consortiumnew com truth m...
3        civilian kill singl us airstrik identifi jessi...
4        iranian woman jail fiction unpublish stori wom...
                              ...
20795    rapper trump poster child white supremaci jero...
20796    n f l playoff schedul matchup odd new york tim...
20797    maci said receiv takeov approach hudson bay ne...
20798    nato russia hold parallel exercis balkan alex ...
20799    keep f aliv david swanson david swanson author...
Name: content, Length: 20800, dtype: object
```

In [17]:
```
transformer = TfidfTransformer(smooth_idf=False)
count_vectorizer = CountVectorizer(ngram_range=(1, 2))
counts = count_vectorizer.fit_transform(df['content'].values)
tfidf = transformer.fit_transform(counts)
```

In [18]:
```
targets = df['label'].values
```

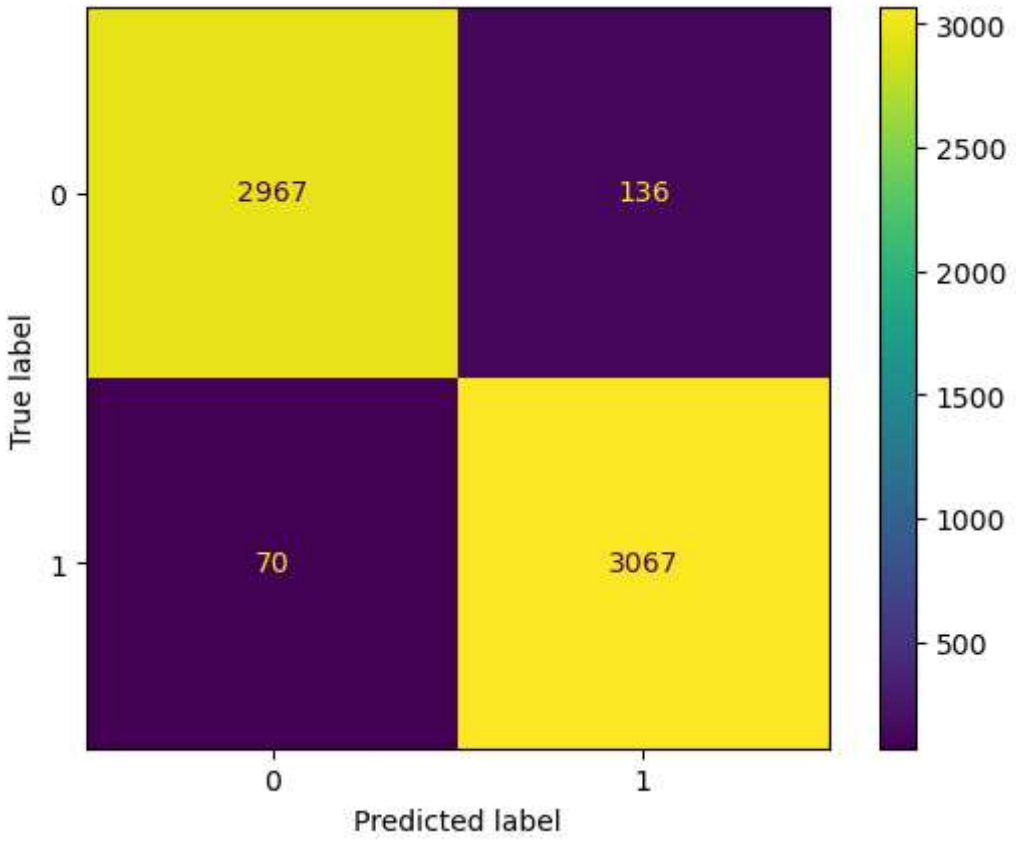
In [19]:
```
X_train, X_test, y_train, y_test = train_test_split(tfidf, targets, test_size=0.3, rar
```

In [20]:
```
def train(model , model_name):
    model.fit(X_train,y_train)
    #print(f"Training accuracy of {model_name} is {model.score(X_train,y_train)}")
    print(f"testing accuracy of {model_name} is {model.score(X_test,y_test)}")
def conf_matrix(model):
    ConfusionMatrixDisplay.from_estimator(
        model,
        X_test,
        y_test
    )
def class_report(model):
    print(classification_report(
        y_test,
        model.predict(X_test)
    ))
```

In [21]:
```
svc_model= SVC()
train(svc_model, 'SVM')
```

```
testing accuracy of SVM is 0.9669871794871795
```

In [22]:
```
conf_matrix(svc_model)
```



In [23]:
```
class_report(svc_model)
```

```
              precision    recall  f1-score   support

           0       0.98      0.96      0.97      3103
           1       0.96      0.98      0.97      3137

    accuracy                           0.97      6240
   macro avg       0.97      0.97      0.97      6240
weighted avg       0.97      0.97      0.97      6240
```

In [30]:
```
from nltk.corpus import stopwords
import string
def process_text(s):

    # Check string to see if they are a punctuation
    nopunc = [char for char in s if char not in string.punctuation]

    # Join the characters again to form the string.
    nopunc = ''.join(nopunc)

    # Convert string to lowercase and remove stopwords
    clean_string = [word for word in nopunc.split() if word.lower() not in stopwords.w
    return clean_string
# Tokenize the text :Convert the normal text strings in to a list of tokens (words tha
#rerun, takes LOOOONG
df['Clean Text'] = df['content'].apply(process_text)
```

In [31]:
```
df.sample(5)
```

Out[31]:

	id	title	author	text	label	content	Clean Text
10288	10288	Graham Hancock Explains Why So Many People Are...	Brianna Acuesta	Hancock speaks a truth that most politicians d...	1	graham hancock explain mani peopl pro trump an...	[graham, hancock, explain, mani, peopl, pro, t...
8134	8134	Reasons to Risk Nuclear Annihilation	Consortiumnews.com	Reasons to Risk Nuclear Annihilation latest ...	1	reason risk nuclear annihil consortiumnew com ...	[reason, risk, nuclear, annihil, consortiumnew...
14784	14784	Strangers on an 18-Hour Train - The New York T...	Rafiq Ebrahim	Many years ago, before my family immigrated to...	0	stranger hour train new york time rafiq ebrahi...	[stranger, hour, train, new, york, time, rafiq...
14753	14753	Brazil Gets an Ounce of Revenge on Germany - T...	Jeré Longman, Doug Mills and Chang W. Lee	RIO DE JANEIRO — An Olympic gold medal in s...	0	brazil get ounc reveng germani new york time j...	[brazil, get, ounc, reveng, germani, new, york...
8504	8504	In a Call to The Times, Trump Blames Democrats...	Maggie Haberman	WASHINGTON — Just moments after the Republi...	0	call time trump blame democrat failur health b...	[call, time, trump, blame, democrat, failur, h...

In [32]:
```
bow_transformer = CountVectorizer(analyzer=process_text).fit(df['Clean Text'])
```

In [33]:
```
news_bow = bow_transformer.transform(df['Clean Text'])
```

In [34]:
```
tfidf_transformer = TfidfTransformer().fit(news_bow)
news_tfidf = tfidf_transformer.transform(news_bow)
print(news_tfidf.shape)
```

```
(20800, 20599)
```

In [35]:
```
from sklearn.naive_bayes import MultinomialNB
fakenews_detect_model = MultinomialNB().fit(news_tfidf, df['label'])
```

In [36]:
```
predictions = fakenews_detect_model.predict(news_tfidf)
print(predictions)
```

```
[1 0 1 ... 0 1 1]
```

In [37]:
```
print (classification_report(df['label'], predictions))
```

	precision	recall	f1-score	support
0	1.00	1.00	1.00	10387
1	1.00	1.00	1.00	10413
accuracy			1.00	20800
macro avg	1.00	1.00	1.00	20800
weighted avg	1.00	1.00	1.00	20800

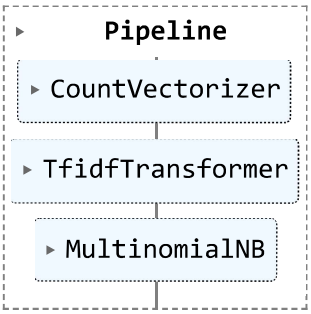
In [38]:
```
news_train, news_test, text_train, text_test = train_test_split(df['content'], df['lab

print(len(news_train), len(news_test), len(news_train) + len(news_test))
```

```
14560 6240 20800
```

In [39]:
```
from sklearn.pipeline import Pipeline
from sklearn.naive_bayes import MultinomialNB
pipeline = Pipeline([
    ('bow', CountVectorizer(analyzer=process_text)),  # strings to token integer count
    ('tfidf', TfidfTransformer()),  # integer counts to weighted TF-IDF scores
    ('classifier', MultinomialNB()),  # train on TF-IDF vectors w/ Naive Bayes classif
])
pipeline.fit(news_train,text_train)
```

Out[39]:



In [40]:
```
predictions = pipeline.predict(news_test)
print(classification_report(predictions,text_test))
```

	precision	recall	f1-score	support
0	0.98	0.82	0.89	3771
1	0.78	0.98	0.87	2469
accuracy			0.88	6240
macro avg	0.88	0.90	0.88	6240
weighted avg	0.90	0.88	0.88	6240

In [120…
```
df.shape
```

Out[120]:
```
(20800, 7)
```

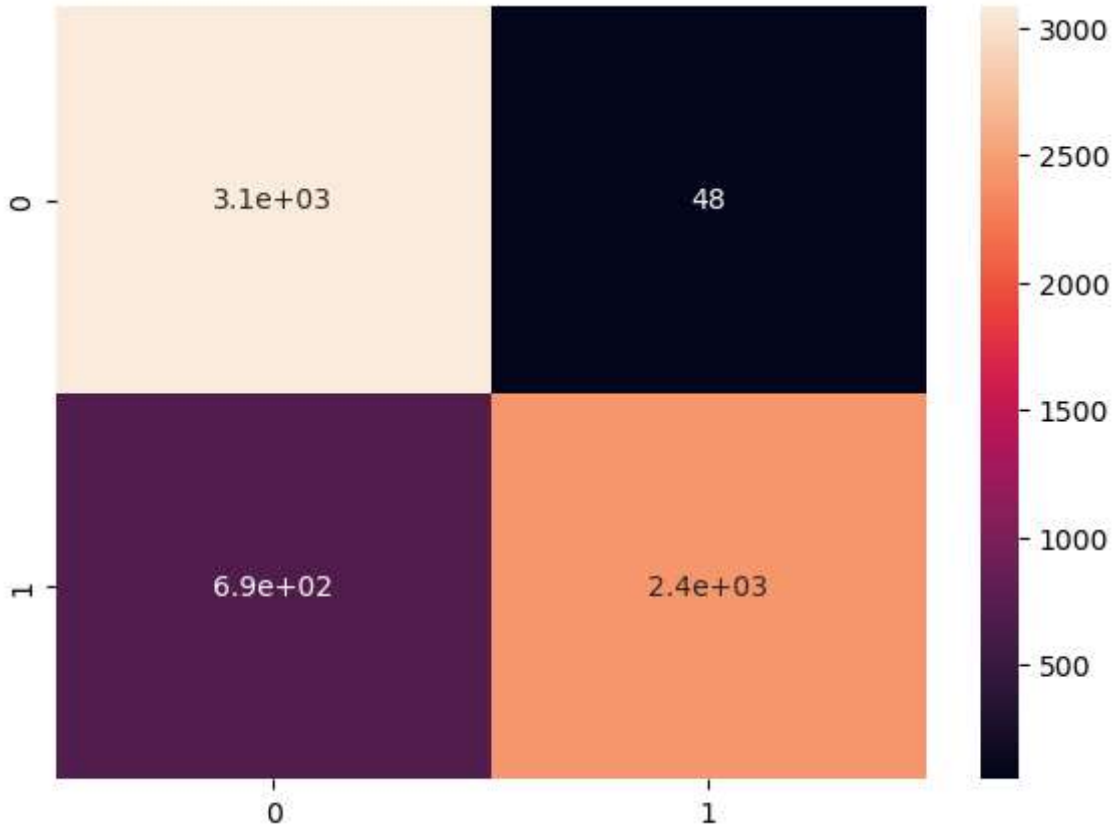
In [41]:
```
accuracy = accuracy_score(list(text_test), predictions)

print("Model Accuracy : ", accuracy)
```

```
Model Accuracy :  0.882051282051282
```

In [42]:
```
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(list(text_test), predictions)
plt.figure(figsize = (7, 5))
sns.heatmap(cm, annot = True)
```

Out[42]: <Axes: >

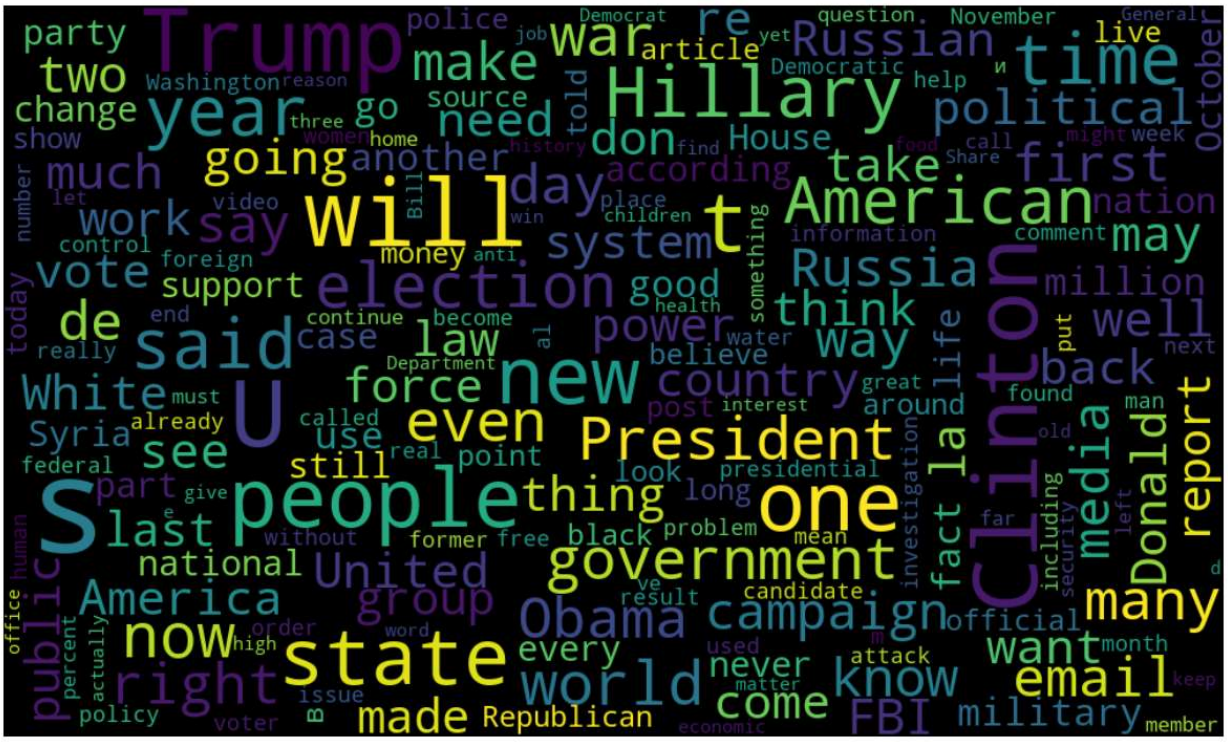


In [7]:
```
!pip install wordcloud
```

```
Collecting wordcloud
  Downloading wordcloud-1.9.2-cp310-cp310-win_amd64.whl (152 kB)
     ---------------------------------- 152.1/152.1 kB 698.8 kB/s eta 0:00:00
Requirement already satisfied: matplotlib in c:\users\sneha\anaconda3\lib\site-packag
es (from wordcloud) (3.7.0)
Requirement already satisfied: pillow in c:\users\sneha\anaconda3\lib\site-packages
(from wordcloud) (9.4.0)
Requirement already satisfied: numpy>=1.6.1 in c:\users\sneha\anaconda3\lib\site-pack
ages (from wordcloud) (1.23.5)
Requirement already satisfied: packaging>=20.0 in c:\users\sneha\anaconda3\lib\site-p
ackages (from matplotlib->wordcloud) (22.0)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\sneha\anaconda3\lib\site
-packages (from matplotlib->wordcloud) (1.4.4)
Requirement already satisfied: cycler>=0.10 in c:\users\sneha\anaconda3\lib\site-pack
ages (from matplotlib->wordcloud) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\sneha\anaconda3\lib\site
-packages (from matplotlib->wordcloud) (4.25.0)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\sneha\anaconda3\lib\site-
packages (from matplotlib->wordcloud) (1.0.5)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\sneha\anaconda3\lib\s
ite-packages (from matplotlib->wordcloud) (2.8.2)
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\sneha\anaconda3\lib\site-
packages (from matplotlib->wordcloud) (3.0.9)
Requirement already satisfied: six>=1.5 in c:\users\sneha\anaconda3\lib\site-packages
(from python-dateutil>=2.7->matplotlib->wordcloud) (1.16.0)
Installing collected packages: wordcloud
Successfully installed wordcloud-1.9.2
```

In [8]:
```
from wordcloud import WordCloud
```

In [11]:
```
consolidated = ' '.join(
    word for word in df['text'][df['label'] == 1].astype(str))
wordCloud = WordCloud(width=1000,
                      height=600,
                      random_state=21,
                      max_font_size=110,
                      collocations=False)
plt.figure(figsize=(15, 10))
plt.imshow(wordCloud.generate(consolidated), interpolation='bilinear')
plt.axis('off')
plt.show()
```



In [15]:
```
consolidated = ' '.join(
    word for word in df['text'][df['label'] == 0].astype(str))
wordCloud = WordCloud(width=1000,
                      height=600,
                      random_state=21,
                      max_font_size=110,
                      collocations=False)
plt.figure(figsize=(15, 10))
plt.imshow(wordCloud.generate(consolidated), interpolation='bilinear')
plt.axis('off')
plt.show()
```