Part	<u> </u>	<pre>from sklearn.linear_model import LogisticRegression from sklearn.metrics import accuracy_score</pre>
Part	ı	nltk.download('stopwords') [nltk_data] Downloading package stopwords to
Company	2]:	[nltk_data] C:\Users\kinbd.com\AppData\Roaming\nltk_data [nltk_data] Package stopwords is already up-to-date! True
The content of the	[['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you', "you're", "you've", "you'll", "you'd", 'your', 'yours', 'yourself', 'yourselves', 'he' 'him', 'his', 'himself', 'she', "she's", 'her', 'hers', 'herself', 'it', "it's", 'its', 'itself', 'they', 'them', 'their', 'theirs', 'themselves', 'what', ich', 'who', 'whom', 'this', 'that', "that'll", 'these', 'those', 'am', 'is', 'are', 'was', 'were', 'be', 'been', 'being', 'have', 'has', 'had', 'having',
Company Comp	t r	o', 'does', 'did', 'doing', 'a', 'an', 'the', 'and', 'but', 'if', 'or', 'because', 'as', 'until', 'while', 'of', 'at', 'by', 'for', 'with', 'about', 'again t', 'between', 'into', 'through', 'during', 'before', 'after', 'above', 'below', 'to', 'from', 'up', 'down', 'in', 'out', 'on', 'off', 'over', 'under', 'ag n', 'further', 'then', 'once', 'here', 'there', 'when', 'where', 'why', 'how', 'all', 'any', 'both', 'each', 'few', 'more', 'most', 'other', 'some', 'such' 'no', 'nor', 'not', 'only', 'own', 'same', 'so', 'than', 'too', 'very', 's', 't', 'can', 'will', 'just', 'don', "don't", 'should', "should've", 'now', 'd', L', 'm', 'o', 're', 've', 'y', 'ain', 'aren', "aren't", 'couldn', "couldn't", 'didn', "didn't", 'doesn', "doesn't", 'hadn', "hadn't", 'hasn', "hasn't", 'ha
The property is a property of the property o	r	n', "haven't", 'isn', "isn't", 'ma', 'mightn', "mightn't", 'mustn', "mustn't", 'needn', "needn't", 'shan', "shan't", 'shouldn', "shouldn't", 'wasn', "wasn' 'weren', "weren't", 'won', "won't", 'wouldn', "wouldn't"]
March Marc		
The content of the	:	
	1	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 1 FLYNN: Hillary Clinton, Big Woman on Campus Daniel J. Flynn Ever get the feeling your life circles the rou 0
The state of the s	3	3 15 Civilians Killed In Single US Airstrike Hav Jessica Purkiss Videos 15 Civilians Killed In Single US Airstr 1
The content of the	. j	id 0
The content of the	a t	author 1957 cext 39 Label 0
### Company of the Co		
]:	<pre>X=news_dataset.drop(columns='label',axis=1)</pre>
The content of the		print(Y)
The content of the	1	0 House Dem Aide: We Didn't Even See Comey's Let 1 FLYNN: Hillary Clinton, Big Woman on Campus 2 2 Why the Truth Might Get You Fired 3 15 Civilians Killed In Single US Airstrike Hav
And Section 10 Page 10	2	 20795 20795 Rapper T.I.: Trump a 'Poster Child For White S 20796 20796 N.F.L. Playoffs: Schedule, Matchups and Odds 20797 20797 Macy's Is Said to Receive Takeover Approach by
Control contro	2	20799 20799 What Keeps the F-35 Alive author \ Darrell Lucus
The content of the second content of the s	3	Consortiumnews.com Jessica Purkiss Howard Portnoy
	2	Jerome Hudson 20796 Benjamin Hoffman 20797 Michael J. de la Merced and Rachel Abrams 20798 Alex Ansary
	0 1 2	text \ House Dem Aide: We Didn't Even See Comey's Let Ever get the feeling your life circles the rou Why the Truth Might Get You Fired October 29,
Sett Tam Setting and the grade of the state	2	Videos 15 Civilians Killed In Single US Airstr Print \nAn Iranian woman has been sentenced to Rapper T. I. unloaded on black celebrities who When the Green Bay Packers lost to the Washing
Section of the control of the contro	2	20797 The Macy's of today grew from the union of sev 20798 NATO, Russia To Hold Parallel Exercises In Bal 20799 David Swanson is an author, activist, journa content
Section of the control of the contro	2	Darrell Lucus House Dem Aide: We Didn't Even S Daniel J. Flynn FLYNN: Hillary Clinton, Big Wo Consortiumnews.com Why the Truth Might Get You Jessica Purkiss 15 Civilians Killed In Single
Company Comp		 20795 Jerome Hudson Rapper T.I.: Trump a 'Poster Chi 20796 Benjamin Hoffman N.F.L. Playoffs: Schedule, Ma 20797 Michael J. de la Merced and Rachel Abrams Macy 20798 Alex Ansary NATO, Russia To Hold Parallel Exer
The content of the	2	David Swanson What Keeps the F-35 Alive [20800 rows x 5 columns]
2000 2 Security Professional Community Prof	3	2 1 3 1 4 1
Ser Sceniforcites (Parameter) and the entry of the control of the		20796 0 20797 0 20798 1 20799 1
Internal Content Conte	:	
	:	<pre>stemmed_content = re.sub('[^a-zA-Z]',' ',content) stemmed_content = stemmed_content.lower() stemmed_content = stemmed_content.split() stemmed_content = [port_stem.stem(word) for word in stemmed_content if not word in stopwords.words('english')]</pre>
Principles States Teach Image data and news constants Teach Image	: [<pre>stemmed_content = ' '.join(stemmed_content) return stemmed_content</pre>
	1 2 3	daniel j flynn flynn hillari clinton big woman consortiumnew com truth might get fire jessica purkiss civilian kill singl us airstri howard portnoy iranian woman jail fiction unpu
Teach	2	20795 jerom hudson rapper trump poster child white s 20796 benjamin hoffman n f l playoff schedul matchup 20797 michael j de la merc rachel abram maci said re 20798 alex ansari nato russia hold parallel exercis
Print(r)	· ·	Name: content, Length: 20800, dtype: object X=news_dataset['content'].values
Vestorizer = TfisiPvectorizer() vestorizer = TfisiPvectorizer() print(x) (0, 1000)	:	print(Y)
venturing = fidifectorized venturing	:	Y. shape
Print(X) (0. 15886)	: [<pre>vectorizer = TfidfVectorizer()</pre>
(6, 15688) 0.78485083587728946 (8, 13473) 6.286809879337967 (9, 7692) 0.2873182807943589 (9, 7692) 0.28731828087943989 (9, 7692) 0.28731828087943989 (9, 7692) 0.2873182808818492 (9, 3696) 0.38931931828082559 (9, 7696) 0.286868128553713 (9, 7696) 0.286868128553713 (1, 76780) 0.380971748085510157 (1, 76780) 0.380971748085510157 (1, 76780) 0.380971748085510157 (1, 76780) 0.380971748085510157 (1, 76780) 0.380971748085510157 (1, 76780) 0.380971748085510157 (1, 76780) 0.380971748085610157 (1, 76780) 0.380971748085510157 (1, 76780) 0.380971748085610157 (1, 76780) 0.380971748087748087610157 (1, 76780) 0.380971748087748087610157 (1, 76780) 0.380971748087748087610157 (1, 76780) 0.3809971748087610157 (1, 76780) 0.3809971748087610157 (1, 76780) 0.380999997748087610157 (1, 76780) 0.38099997748087610157 (1, 76780) 0.38099997748087610157 (1, 76780) 0.38099997748087610157 (1, 76780) 0.38099997748087610157 (1, 76780) 0.380999997748087610157 (1, 76780) 0.380999997748087610157 (1, 76780) 0.380999997748087610157 (1, 76780) 0.380999997748087610157 (1, 76780) 0.380999997748087610157 (1, 76780) 0.380999997748087610157 (1, 76780) 0.380999997748087610157 (1, 76780) 0.3809999997748087610157 (1, 76780) 0.3809999997748087610157 (1, 76780) 0.380999999774808087610157 (1, 76780) 0.3809999997757 (1, 76780) 0.380999999775		
(8, 7692)		(0, 15686)
(0, 2099)		(0, 7692) 0.24785219520671603 (0, 7005) 0.21874169089359144 (0, 4973) 0.233316966909351 (0, 3792) 0.2705332480845492
(1, 5803)		(0, 2959) 0.2468450128533713 (0, 2483) 0.3676519686797209 (0, 267) 0.27010124977708766 (1, 16799) 0.30071745655510157
(1, 1497)		(1, 5503) 0.7143299355715573 (1, 3568) 0.26373768806048464 (1, 2813) 0.19094574062359204 (1, 2223) 0.3827320386859759
(2, 3183)		(1, 1497) 0.2939891562094648 (2, 15611) 0.41544962664721613 (2, 9620) 0.49351492943649944 (2, 5968) 0.3474613386728292
(26797, 19388)		(2, 5389) 0.3866530551182615 (2, 3103) 0.46097489583229645 : : (20797, 13122) 0.2482526352197606 (20797, 12344) 0.27263457663336677
(20797, 8364) 0.22322585870464118 (20797, 7042) 0.1799048897328688 (20797, 3643) 0.21155500613623743 (20797, 1287) 0.33538056804139855 (20797, 699) 0.305685846079762347 (20797, 43) 0.29710221860700626 (20798, 13046) 0.22362267488270608 (20798, 11052) 0.4460515589182236 (20798, 16177) 0.319240370187028 (20798, 6889) 0.32496285694299426 (20798, 5082) 0.4603701450239529 (20798, 5082) 0.4603701450239529 (20798, 588) 0.3112141524658974 (20798, 588) 0.3112141524658974 (20798, 588) 0.3112141524658974 (20799, 3693) 0.48938939273780013 (20799, 3623) 0.37927626273066584 (20799, 3623) 0.37927626273066584 (20799, 377) 0.5677577267055112 ***X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.2, stratify=Y, random_state=2) ***model = LogisticRegression()*** model.fit(X_train, Y_train)		(20797, 12138) 0.24778257724396507 (20797, 10306) 0.08038079000566466 (20797, 9588) 0.174553480255222 (20797, 9518) 0.2954204003420313 (20797, 8988) 0.36160868928090795
(20797, 43)		(20797, 8364) 0.22322585870464118 (20797, 7042) 0.21799048897828688 (20797, 3643) 0.21155500613623743 (20797, 1287) 0.33538056804139865 (20797, 699) 0.30685846079762347
(20798, 5032) 0.4083701450239529 (20798, 1125) 0.44660515589182236 (20798, 588) 0.3112141524638974 (20798, 350) 0.28446937819072576 (20799, 14852) 0.5677577267055112 (20799, 8036) 0.45983893273780013 (20799, 3623) 0.37927626273066584 (20799, 377) 0.5677577267055112 X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.2, stratify=Y, random_state=2) model = LogisticRegression() model.fit(X_train, Y_train)		(20797, 43) 0.29710241860700626 (20798, 13046) 0.22363267488270608 (20798, 11052) 0.4460515589182236 (20798, 10177) 0.3192496370187028
<pre>(20799, 8036) 0.45983893273780013 (20799, 3623) 0.37927626273066584 (20799, 377) 0.5677577267055112 X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.2, stratify=Y, random_state=2) model = LogisticRegression() model.fit(X_train, Y_train)</pre>		(20798, 5032) 0.4083701450239529 (20798, 1125) 0.4460515589182236 (20798, 588) 0.3112141524638974 (20798, 350) 0.28446937819072576
<pre>model = LogisticRegression() model.fit(X_train, Y_train)</pre>		(20799, 8036) 0.45983893273780013 (20799, 3623) 0.37927626273066584 (20799, 377) 0.5677577267055112
model: Tit(_train, T_train)		
LogisticRegression()		
<pre>X_train_prediction = model.predict(X_train) training_data_accuracy = accuracy_score(X_train_prediction, Y_train)</pre>	:	<pre>X_train_prediction = model.predict(X_train)</pre>
print('Accuracy score of the training data : ', training_data_accuracy) Accuracy score of the training data : 0.9865985576923076		
<pre>X_test_prediction = model.predict(X_test) test_data_accuracy = accuracy_score(X_test_prediction, Y_test)</pre>	:	<pre>X_test_prediction = model.predict(X_test)</pre>
print('Accuracy score of the test data : ', test_data_accuracy) Accuracy score of the test data : 0.9790865384615385	A	
<pre>X_new = X_test[0] prediction = model.predict(X_new) print(prediction)</pre>		<pre>prediction = model.predict(X_new)</pre>
<pre>if (prediction[0]==0): print('The news is Real') else: print('The news is Fake')</pre>		<pre>print('The news is Real') else:</pre>
[1] The news is Fake		print(The news is ruke)

In [1]:

import numpy as np
import pandas as pd
import re
from nltk.corpus import stopwords

from nltk.stem.porter import PorterStemmer

 $\textbf{from} \ \ \textbf{sklearn.feature_extraction.text} \ \ \textbf{import} \ \ \textbf{TfidfVectorizer}$