In [52]:
```
import pandas as pd


from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer

from sklearn.naive_bayes import MultinomialNB
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import accuracy_score
```

In [54]:
```
df = pd.read_csv("../admin1/Downloads/test.csv")
df
```

Out[54]:

	Class Index	Title	Description
0	3	Fears for T N pension after talks	Unions representing workers at Turner Newall...
1	4	The Race is On: Second Private Team Sets Launc...	SPACE.com - TORONTO, Canada -- A second\team o...
2	4	Ky. Company Wins Grant to Study Peptides (AP)	AP - A company founded by a chemistry research...
3	4	Prediction Unit Helps Forecast Wildfires (AP)	AP - It's barely dawn when Mike Fitzpatrick st...
4	4	Calif. Aims to Limit Farm-Related Smog (AP)	AP - Southern California's smog-fighting agenc...
...	...	...	...
7595	1	Around the world	Ukrainian presidential candidate Viktor Yushch...
7596	2	Void is filled with Clement	With the supply of attractive pitching options...
7597	2	Martinez leaves bitter	Like Roger Clemens did almost exactly eight ye...
7598	3	5 of arthritis patients in Singapore take Bext...	SINGAPORE : Doctors in the United States have ...
7599	3	EBay gets into rentals	EBay plans to buy the apartment and home renta...

7600 rows × 3 columns

In [56]:
```
df.isnull().sum()
```

Out[56]:
```
Class Index    0
Title          0
Description    0
dtype: int64
```

In [58]:
```
df['Class Index'].value_counts()
```

Out[58]:
```
Class Index
3    1900
4    1900
2    1900
1    1900
Name: count, dtype: int64
```

In [60]:
```
# VECTORIZATION
# 1. Combine Title and Description into a single feature
# Adding a space between them ensures words aren't accidentally joined

df['Full_Text'] = df['Title'] + " " + df['Description']

# 2. Convert combined text to numerical features using TF-IDF
tfidf = TfidfVectorizer(stop_words='english', max_features=5000)

# 3. fit_transform on the new combined column
x = tfidf.fit_transform(df['Full_Text'])
y = df['Class Index']
```

In [62]:
```
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2)
```

In [64]:
```
model = MultinomialNB()
model.fit(x_train, y_train)
y_pred = model.predict(x_test)
acc = accuracy_score(y_test, y_pred)
acc
```

Out[64]: 0.8835526315789474

In [66]:
```
model = RandomForestClassifier()
model.fit(x_train, y_train)
y_pred = model.predict(x_test)
acc = accuracy_score(y_test, y_pred)
acc
```

Out[66]: 0.8217105263157894

In [67]:
```
model = LogisticRegression()
model.fit(x_train, y_train)
y_pred = model.predict(x_test)
acc = accuracy_score(y_test, y_pred)
acc
```

```
/home/admin1/anaconda3/lib/python3.9/site-packages/sklearn/linear_model/_logistic.py:4
60: ConvergenceWarning: 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[67]: 0.8861842105263158

In [ ]:

In [ ]: