```
# Data Loading
from google.colab import drive
import pandas as pd
drive.mount('/content/drive')
file_path = '/content/drive/My Drive/english_news_dataset.csv'
df = pd.read_csv(file_path)
```

```
Mounted at /content/drive
```

```
df.shape
```

```
(199706, 4)
```

```
import numpy as np
from bs4 import BeautifulSoup
import regex
```

```
import re
from nltk import word_tokenize, sent_tokenize
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import MultinomialNB
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.pipeline import make_pipeline
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
from sklearn.model_selection import cross_val_score
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.ensemble import RandomForestClassifier
```

```
df.head()
df
```

	Headline	Content	News Categories	Date
0	Congress leader Baljinder Singh shot dead at h...	Congress leader Baljinder Singh was shot dead ...	['national']	19-09-2023

1	17-year-old girl preparing for NEET dies by su...	Another NEET aspirant died by suicide in Rajas...	['national']	19-09-2023
2	Hampers to welcome MPs in new Parliament tomor...	In order to mark the first-ever working day of...	['national']	19-09-2023
3	Only 10% women lawmakers in RS, while only 14%...	Congress President Mallikarjun Kharge, while s...	['national']	19-09-2023
4	Ganesh temple decorated with notes, coins wort...	The Sri Sathya Ganapathi Temple in Bengaluru a...	['national']	19-09-2023
...	...	...	...	...
199701	Cause for age related diabetes can be pancreat...	The pancreas is an incredibly important organ,...	['science', 'Health___Fitness']	2024-01-20
199702	Study unveils Why sugary drinks may be bad for...	A recent study published in Oral Diseases has...	['science', 'Health___Fitness']	2024-01-20
	Why do Hospital acquired infections happen?	Hospital acquired infections (HAIs) refer to		2024-01

GROUPING OF CATEGORIES

```
from sklearn.model_selection import StratifiedKFold, cross_val_score
threshold = 5

# Identify classes with fewer instances
class_counts = df['News Categories'].value_counts()
rare_classes = class_counts[class_counts < threshold].index

# Group rare classes into a broader category 'Other'
df['category_grouped'] = df['News Categories'].apply(lambda x: 'Other' if x in rare_classes else x)
```

```
df["News Categories"]
```

```
0                          ['national']
1                          ['national']
2                          ['national']
3                          ['national']
4                          ['national']
                       ...
199701    ['science', 'Health___Fitness']
199702    ['science', 'Health___Fitness']
199703               ['Health___Fitness']
199704    ['science', 'Health___Fitness']
199705               ['Health___Fitness']
Name: News Categories, Length: 199706, dtype: object
```

```
df['category_grouped']
```

```
0                          ['national']
1                          ['national']
2                          ['national']
3                          ['national']
4                          ['national']
                       ...
199701    ['science', 'Health___Fitness']
199702    ['science', 'Health___Fitness']
199703               ['Health___Fitness']
199704    ['science', 'Health___Fitness']
199705               ['Health___Fitness']
Name: category_grouped, Length: 199706, dtype: object
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 199706 entries, 0 to 199705
Data columns (total 5 columns):
 #   Column            Non-Null Count   Dtype 
---  ------            --------------   ----- 
 0   Headline          199706 non-null  object
 1   Content           199706 non-null  object
 2   News Categories   199706 non-null  object
 3   Date              199706 non-null  object
 4   category_grouped  199706 non-null  object
dtypes: object(5)
memory usage: 7.6+ MB
```

```
df.isnull().sum()
```

```
Headline            0
Content             0
News Categories     0
Date                0
category_grouped    0
dtype: int64
```

```
df.duplicated().sum()
```

```
35452
```

```
df["Date"].head()
```

```
0    19-09-2023
1    19-09-2023
```

```
2    19-09-2023
3    19-09-2023
4    19-09-2023
Name: Date, dtype: object
```

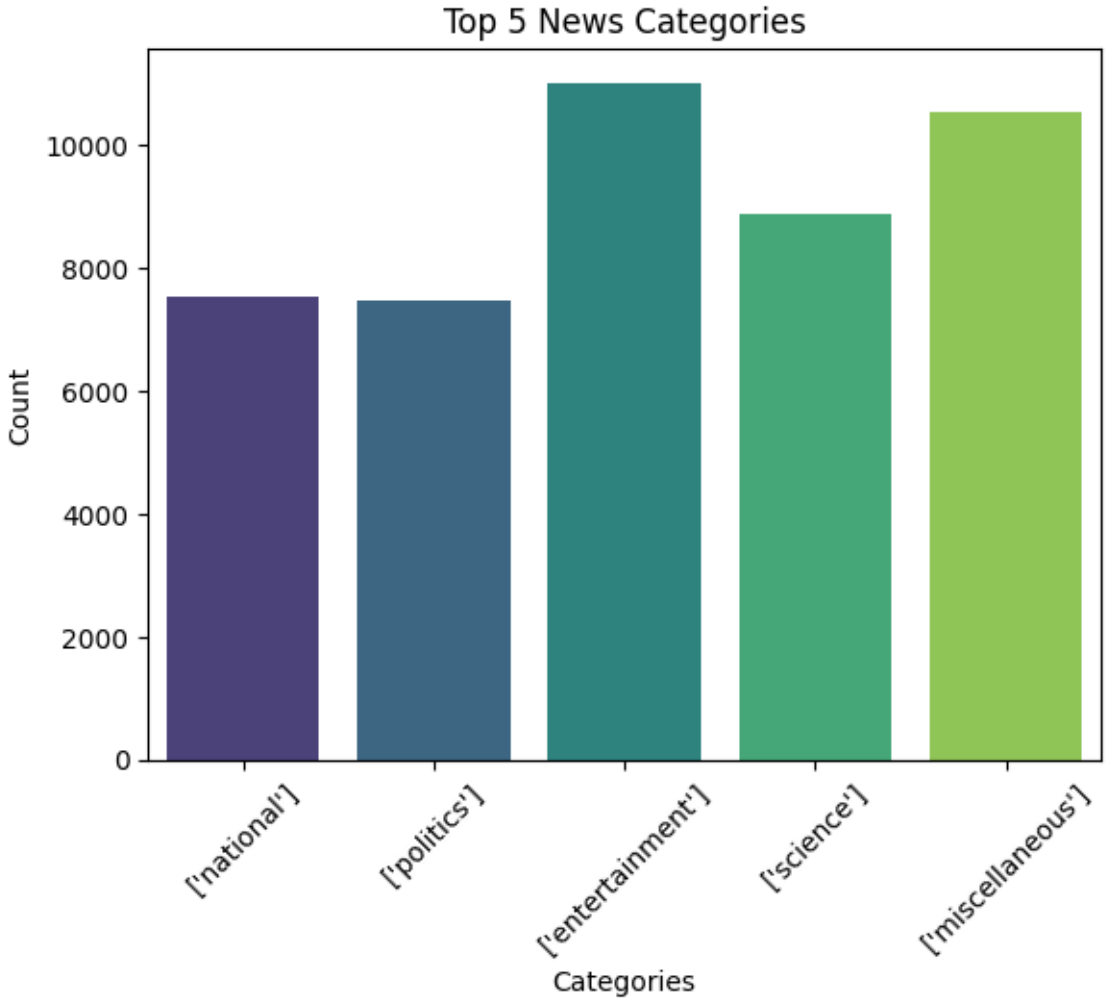
```
df["News Categories"].unique().sum()
```

```
'['national']['entertainment', 'national']['politics', 'national']['world', 'national']['national', 'technolog
y']['business', 'national']['sports', 'national']['world', 'national', 'Health___Fitness']['national', 'Health_
__Fitness']['business', 'technology']['business']['business', 'startup']['automobile', 'business', 'technolog
y']['business', 'fashion']['world', 'business']['world', 'business', 'technology']['automobile', 'business']['b
usiness', 'entertainment', 'national']['world', 'business', 'national']['business', 'science', 'technology']['c
ryptocurrency', 'business', 'technology']['automobile', 'business', 'national']['politics']['politics', 'sport
s', 'Asia_Cup_2023']['politics', 'entertainment']['sports']['sports', 'entertainment']['sports', 'Asia_Cup_202
3']['Asia_Cup_2023', 'sports']['sports', 'Asia_Cup_2023', 'entertainment']['technology']['world', 'technology']
```

```
import seaborn as sns
import matplotlib.pyplot as plt
top_n = 5
top_categories = df['News Categories'].value_counts().nlargest(top_n).index

df_top = df[df['News Categories'].isin(top_categories)]

sns.countplot(x='News Categories', data=df_top, palette='viridis')
plt.title(f'Top {top_n} News Categories')
plt.xlabel('Categories')
plt.xticks(rotation=45)
```

```
plt.ylabel('Count')
plt.show();
```

```
<ipython-input-13-03ea06321cc3>:8: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable

  sns.countplot(x='News Categories', data=df_top, palette='viridis')
```



```
df['News Categories']
```

```
0                          ['national']
1                          ['national']
2                          ['national']
3                          ['national']
4                          ['national']
                       ...
199701    ['science', 'Health___Fitness']
199702    ['science', 'Health___Fitness']
199703               ['Health___Fitness']
199704    ['science', 'Health___Fitness']
199705               ['Health___Fitness']
Name: News Categories, Length: 199706, dtype: object
```

```
import string
string.punctuation
punc=string.punctuation
def remove_punc(text):
    return text.translate(str.maketrans('', '',punc))

df["News Categories"]=df["News Categories"].apply(remove_punc)
df.head()
```

	Headline	Content	News Categories	Date	category_grouped
0	Congress leader Baljinder Singh shot dead at h...	Congress leader Baljinder Singh was shot dead ...	national	19-09-2023	['national']
1	17-year-old girl preparing for NEET dies by su...	Another NEET aspirant died by suicide in Rajas...	national	19-09-2023	['national']
2	Hampers to welcome MPs in new Parliament tomor...	In order to mark the first-ever working day of...	national	19-09-2023	['national']

```
df['Date'] = pd.to_datetime(df['Date'], infer_datetime_format=True)
```

```
<ipython-input-16-45a55d54a14d>:1: UserWarning: Parsing dates in DD/MM/YYYY format when dayfirst=False (the defa
  df['Date'] = pd.to_datetime(df['Date'], infer_datetime_format=True)
```

```
df['Date'] = pd.to_datetime(df['Date'], format='%Y-%m-%d')
```

```
df['year'] = df['Date'].dt.year
df['month'] = df['Date'].dt.month
df['day'] = df['Date'].dt.day
```

```
df.head()
```

	Headline	Content	News Categories	Date	category_grouped	year	month	day
0	Congress leader Baljinder Singh shot dead at h...	Congress leader Baljinder Singh was shot dead ...	national	2023-09-19	['national']	2023	9	19
1	17-year-old girl preparing for NEET dies by su...	Another NEET aspirant died by suicide in Rajas...	national	2023-09-19	['national']	2023	9	19
2	Hampers to welcome MPs in new Parliament tomor...	In order to mark the first-ever working day of...	national	2023-09-19	['national']	2023	9	19

```
df=df.drop('Date',axis=1)
df.head()
```

	Headline	Content	News Categories	category_grouped	year	month	day
0	Congress leader Baljinder Singh shot dead at h...	Congress leader Baljinder Singh was shot dead ...	national	['national']	2023	9	19
1	17-year-old girl preparing for NEET dies by su...	Another NEET aspirant died by suicide in Rajas...	national	['national']	2023	9	19
2	Hampers to welcome MPs in new Parliament tomor...	In order to mark the first-ever working day of...	national	['national']	2023	9	19

PREPARATION OF TEXT

```
# LOWER CASE
df["Content"]=df["Content"].str.lower()
df.head()
```

	Headline	Content	News Categories	category_grouped	year	month	day

0	Congress leader Baljinder Singh shot dead at h...	congress leader baljinder singh was shot dead ...	national	['national']	2023	9	19
1	17-year-old girl preparing for NEET dies by su...	another neet aspirant died by suicide in rajas...	national	['national']	2023	9	19
2	Hampers to welcome MPs in new Parliament tomor...	in order to mark the first-ever working day of...	national	['national']	2023	9	19

```
# REMOVAL OF HTML TAGS
def has_html_tags(text):
    soup = BeautifulSoup(text, 'html.parser')
    return bool(soup.find())

df['has_html_tags'] = df['Content'].apply(has_html_tags)
```

```
df.head()
```

	Headline	Content	News Categories	category_grouped	year	month	day	has_html_tags
0	Congress leader Baljinder Singh shot dead at h...	congress leader baljinder singh was shot dead ...	national	['national']	2023	9	19	False
1	17-year-old girl preparing for NEET dies by su...	another neet aspirant died by suicide in rajas...	national	['national']	2023	9	19	False

2	Hampers to welcome MPs in new Parliament tomor...	in order to mark the first-ever working day of...	national	['national']	2023	9	19	False

```
count_true = df['has_html_tags'].sum()
count_true
```

```
0
```

```
df = df.drop('has_html_tags', axis=1)
df
```

	Headline	Content	News Categories	category_grouped	year	month	day
0	Congress leader Baljinder Singh shot dead at h...	congress leader baljinder singh was shot dead ...	national	['national']	2023	9	19
1	17-year-old girl preparing for NEET dies by su...	another neet aspirant died by suicide in rajas...	national	['national']	2023	9	19
2	Hampers to welcome MPs in new Parliament tomor...	in order to mark the first-ever working day of...	national	['national']	2023	9	19

3	Only 10% women lawmakers in RS, while only 14%...	congress president mallikarjun kharge, while s...	national	['national']	2023	9	19
4	Ganesh temple decorated with notes, coins wort...	the sri sathya ganapathi temple in bengaluru a...	national	['national']	2023	9	19
...	...	...	...	...	...	...	...
199701	Cause for age related diabetes can be pancreat...	the pancreas is an incredibly important organ,...	science HealthFitness	['science', 'Health___Fitness']	2024	1	20
199702	Study unveils Why sugary drinks may be bad for...	a recent study published in oral diseases has...	science HealthFitness	['science', 'Health___Fitness']	2024	1	20

```
# REMOVAL OF EMOJIS
import regex
def has_emoji(text):
    emoji_pattern = regex.compile(r'\p{Emoji}', flags=regex.UNICODE)
    return bool(emoji_pattern.search(text))


has_emojis =  df['Content'].apply(has_emoji)

has_emojis
```

```
0         False
1          True
2         False
3          True
4          True
          ...
199701    False
```

```
199702    False
199703    False
199704     True
199705    False
Name: Content, Length: 199706, dtype: bool
```

```
has_emojis.sum()
```

```
127178
```

```
def remove_emojis(text):
    emoji_pattern = regex.compile(r'\p{Emoji}', flags=regex.UNICODE)
    return emoji_pattern.sub('', text)

df['Content'] = df['Content'].apply(remove_emojis)
```

```
has_emojis =  df['Content'].apply(has_emoji)
has_emojis
```

```
0         False
1         False
2         False
3         False
4         False
          ...
199701    False
199702    False
199703    False
199704    False
199705    False
Name: Content, Length: 199706, dtype: bool
```

```
has_emojis.sum()
```

```
0
```

REMOVAL OF URLS, PUNCTUATION, STOPWORDS, ABBREVIATIONS

```
import re
def remove_url(text):
    pattern=re.compile(r'https?://\S+|www\.S+')
    return pattern.sub(r'',text)
df["Content"]=df["Content"].apply(remove_url)
```

```
import string
string.punctuation
```

```
'!"#$%&\'()*+,-./:;<=>?@[\\]^_`{|}~'
```

```
punc=string.punctuation
def remove_punc(text):
    return text.translate(str.maketrans('', '',punc))
df["Content"]=df["Content"].apply(remove_punc)
```

```
import nltk
from nltk.corpus import stopwords
nltk.download('stopwords')
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
True
```

```
stop_words=set(stopwords.words('english'))
```

```
def remove_stopwords(text):
    words = text.split()
    filtered_words = [word for word in words if word not in stop_words]
    return " ".join(filtered_words)

df["Content"]=df["Content"].apply(lambda x: remove_stopwords(x))
```

```
import nltk
nltk.download('punkt')
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
True
```

```
abbreviation_dict = {
    'LOL': 'laugh out loud',
    'BRB': 'be right back',
    'OMG': 'oh my god',
    'AFAIK': 'as far as I know',
    'AFK': 'away from keyboard',
    'ASAP': 'as soon as possible',
    'ATK': 'at the keyboard',
    'ATM': 'at the moment',
    'A3': 'anytime, anywhere, anyplace',
    'BAK': 'back at keyboard',
    'BBL': 'be back later',
    'BBS': 'be back soon',
```

```
    'BFN': 'bye for now',
    'B4N': 'bye for now',
    'BRB': 'be right back',

    'BRT': 'be right there',
    'BTW': 'by the way',
    'B4': 'before',
    'B4N': 'bye for now',
    'CU': 'see you',
    'CUL8R': 'see you later',
    'CYA': 'see you',
    'FAQ': 'frequently asked questions',
    'FC': 'fingers crossed',
    'FWIW': 'for what it\'s worth',
    'FYI': 'For Your Information',
    'GAL': 'get a life',
    'GG': 'good game',
    'GN': 'good night',
    'GMTA': 'great minds think alike',
    'GR8': 'great!',
    'G9': 'genius',
    'IC': 'i see',
    'ICQ': 'i seek you',
    'ILU': 'i love you',
    'IMHO': 'in my honest/humble opinion',
    'IMO': 'in my opinion',
    'IOW': 'in other words',
    'IRL': 'in real life',
    'KISS': 'keep it simple, stupid',
    'LDR': 'long distance relationship',
    'LMAO': 'laugh my a.. off',
    'LOL': 'laughing out loud',
    'LTNS': 'long time no see',
    'L8R': 'later',
    'MTE': 'my thoughts exactly',
    'M8': 'mate',
```

```
    'NRN': 'no reply necessary',
    'OIC': 'oh i see',
    'PITA': 'pain in the a..',
    'PRT': 'party',
    'PRW': 'parents are watching',
    'QPSA?': 'que pasa?',
    'ROFL': 'rolling on the floor laughing',
    'ROFLOL': 'rolling on the floor laughing out loud',
    'ROTFLMAO': 'rolling on the floor laughing my a.. off',
    'SK8': 'skate',
    'STATS': 'your sex and age',
    'ASL': 'age, sex, location',
    'THX': 'thank you',
    'TTFN': 'ta-ta for now!',
    'TTYL': 'talk to you later',
    'U': 'you',
    'U2': 'you too',
    'U4E': 'yours for ever',
    'WB': 'welcome back',
    'WTF': 'what the f...',
    'WTG': 'way to go!',
    'WUF': 'where are you from?',
    'W8': 'wait...',
    '7K': 'sick laughter',
    'TFW': 'that feeling when',
    'MFW': 'my face when',
    'MRW': 'my reaction when',
    'IFYP': 'i feel your pain',
    'LOL': 'laughing out loud',
    'TNTL': 'trying not to laugh',
    'JK': 'just kidding',
    'IDC': 'i don’t care',
    'ILY': 'i love you',
    'IMU': 'i miss you',
    'ADIH': 'another day in hell',
```

```
    'IDC': 'i don’t care',
    'ZZZ': 'sleeping, bored, tired',
    'WYWH': 'wish you were here',
    'TIME': 'tears in my eyes',
    'BAE': 'before anyone else',
    'FIMH': 'forever in my heart',
    'BSAAW': 'big smile and a wink',
    'BWL': 'bursting with laughter',
    'LMAO': 'laughing my a** off',
    'BFF': 'best friends forever',
    'CSL': 'can’t stop laughing',
}
```

```
def replace_abbreviations(text, abbreviation_dict):
    for abbreviation, full_form in abbreviation_dict.items():
        text = text.replace(abbreviation, full_form)
    return text
df['Content'] = df['Content'].apply(lambda x: replace_abbreviations(x,abbreviat
df.head()
```

	Headline	Content	News Categories	category_grouped	year	month	day
0	Congress leader Baljinder Singh shot dead at h...	congress leader baljinder singh shot dead hous...	national	['national']	2023	9	19
1	17-year-old girl preparing for NEET dies by su...	another neet aspirant died suicide rajasthans ...	national	['national']	2023	9	19
2	Hampers to welcome MPs in new Parliament tomor...	order mark firstever working day new parliamen...	national	['national']	2023	9	19

TOKENIZATION AND DATA AGGREGATION

```
import nltk
nltk.download('punkt')
from nltk import word_tokenize , sent_tokenize

def tokenize_text(text):
    # Tokenize each sentence into words
    words_list = [word_tokenize(sentence) for sentence in sent_tokenize(text)]

    words = ' '.join(' '.join(words) for words in words_list)
```

```
    return words

df["Content"] = df["Content"].apply(tokenize_text)

df.head()
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
```

	Headline	Content	News Categories	category_grouped	year	month	day
0	Congress leader Baljinder Singh shot dead at h...	congress leader baljinder singh shot dead hous...	national	['national']	2023	9	19
1	17-year-old girl preparing for NEET dies by su...	another neet aspirant died suicide rajasthans ...	national	['national']	2023	9	19
2	Hampers to welcome MPs in new Parliament tomor...	order mark firstever working day new parliamen...	national	['national']	2023	9	19

```
# SPLITTING
!pip install imbalanced-learn

from imblearn.over_sampling import SMOTE
from sklearn.model_selection import train_test_split
from sklearn.model_selection import StratifiedKFold, cross_val_score
from sklearn.naive_bayes import MultinomialNB
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import LabelEncoder
from sklearn.utils.class_weight import compute_class_weight

X = df['Content']
y = df['category_grouped']

# Encoding labels
le = LabelEncoder()
```

```
y_encoded = le.fit_transform(y)

class_weights_train = compute_class_weight('balanced', classes=np.unique(y_encoded), y=y_encoded)
X
```

```
Requirement already satisfied: imbalanced-learn in /usr/local/lib/python3.10/dist-packages (0.10.1)
Requirement already satisfied: numpy>=1.17.3 in /usr/local/lib/python3.10/dist-packages (from imbalanced-learn)
Requirement already satisfied: scipy>=1.3.2 in /usr/local/lib/python3.10/dist-packages (from imbalanced-learn) (
Requirement already satisfied: scikit-learn>=1.0.2 in /usr/local/lib/python3.10/dist-packages (from imbalanced-
Requirement already satisfied: joblib>=1.1.1 in /usr/local/lib/python3.10/dist-packages (from imbalanced-learn)
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.10/dist-packages (from imbalanced-
0         congress leader baljinder singh shot dead hous...
1         another neet aspirant died suicide rajasthans ...
2         order mark firstever working day new parliamen...
3         congress president mallikarjun kharge speaking...
4         sri sathya ganapathi temple bengaluru adorned ...
                              ...
199701    pancreas incredibly important organ particular...
199702    recent study published oral diseases reported ...
199703    hospitalacquired infections hais refer infecti...
199704    scientists university oxford uk launched first...
199705    high blood pressure happens force blood pushin...
Name: Content, Length: 199706, dtype: object
```

```
y.shape
```

```
(199706,)
```

```
y_encoded
```

```
array([284, 284, 284, ...,  72, 346,  72])
```

```
from sklearn.model_selection import train_test_split

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y_encoded, test_size=0.2, random_state=42)
```

```
# MODELING : BoW
# Multinomial Naive Bayes with Bag of Words
from sklearn.naive_bayes import MultinomialNB
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.pipeline import make_pipeline
from sklearn.metrics import accuracy_score,classification_report


model = make_pipeline(CountVectorizer(), MultinomialNB())

model.fit(X_train, y_train)

y_pred = model.predict(X_test)

accuracy = accuracy_score(y_test, y_pred)
print(f"MultinomialNB with Bag of Words accuracy: {accuracy:.3f}")
# Print classification report
print("Classification Report:\n", classification_report(y_test, y_pred))
```

```
              429       1.00      1.00      1.00         1
              430       1.00      1.00      1.00         5
              431       1.00      0.95      0.98        62
```

431	1.00	0.95	0.98	62
432	1.00	1.00	1.00	5
433	1.00	0.97	0.99	109
434	1.00	1.00	1.00	6
435	1.00	0.50	0.67	22
436	1.00	1.00	1.00	39
437	1.00	0.42	0.59	31
438	1.00	1.00	1.00	5
439	1.00	1.00	1.00	26
440	0.98	0.61	0.75	72
441	1.00	1.00	1.00	100
442	0.00	0.00	0.00	1
443	1.00	0.14	0.25	7
444	0.00	0.00	0.00	4
445	1.00	0.50	0.67	2
446	1.00	1.00	1.00	9
447	1.00	1.00	1.00	2
448	1.00	1.00	1.00	2
449	1.00	1.00	1.00	7
450	1.00	1.00	1.00	5
451	1.00	1.00	1.00	1
452	1.00	1.00	1.00	6
453	1.00	0.33	0.50	3
454	1.00	0.88	0.93	56
455	1.00	0.84	0.91	19
456	0.88	0.36	0.51	42
457	0.00	0.00	0.00	1
458	1.00	0.60	0.75	5
459	1.00	1.00	1.00	51
460	0.79	0.89	0.84	680
461	0.00	0.00	0.00	4
462	1.00	0.95	0.98	22
463	1.00	1.00	1.00	6
464	0.00	0.00	0.00	4
465	1.00	1.00	1.00	6
466	1.00	0.40	0.57	5
467	1.00	0.94	0.96	263
468	0.00	0.00	0.00	1
469	0.00	0.00	0.00	4
470	0.00	0.00	0.00	2
471	1.00	1.00	1.00	1

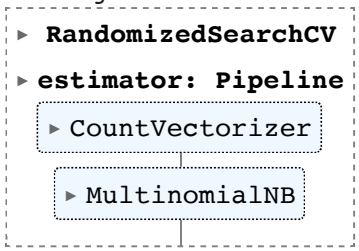
```
              471       1.00      1.00      1.00         1
              472       0.00      0.00      0.00         8
              473       1.00      0.33      0.50        39
              474       1.00      1.00      1.00         1
              475       1.00      0.89      0.94         9
              476       1.00      1.00      1.00         3
              477       0.88      0.68      0.76        31
              478       1.00      0.64      0.78       117
              479       0.98      1.00      0.99       268
              480       0.79      0.82      0.80      1157

         accuracy                           0.89     39942
        macro avg       0.77      0.67      0.70     39942
     weighted avg       0.89      0.89      0.88     39942

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Precis
  _warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Precis
```

```
# CROSS-VALIDATION
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import StratifiedKFold

cv_scores = cross_val_score(model, X, y_encoded, cv=StratifiedKFold(n_splits=3, shuffle=True), scoring='accuracy')

print(f"Cross-Validation Scores:{cv_scores}")

print(f"Mean Accuracy: {np.mean(cv_scores):.2f}")
```

```
Cross-Validation Scores:[0.88128108 0.88278328 0.87895085]
Mean Accuracy: 0.88
```

```
# FINE TUNING
from sklearn.model_selection import RandomizedSearchCV
from scipy.stats import uniform, randint

param_dist = {
    'countvectorizer__max_features': [5000, 10000, None],
    'countvectorizer__ngram_range': [(1, 1), (1, 2)],
    'multinomialnb__alpha': uniform(0.1, 2.0)  # Example range for alpha
}
cv = StratifiedKFold(n_splits=5, shuffle=True, random_state=42)

random_search = RandomizedSearchCV(model, param_distributions=param_dist, n_iter=5, scoring='accuracy', cv=cv, verb
random_search.fit(X, y_encoded)
```

```
Fitting 5 folds for each of 5 candidates, totalling 25 fits
```




```
best_params = random_search.best_params_
print("Best Parameters:", best_params)
```

```
Best Parameters: {'countvectorizer__max_features': 10000, 'countvectorizer__ngram_range': (1, 1), 'multinomialnb
```

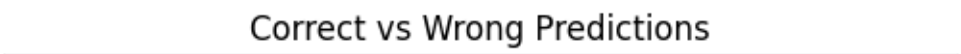
```
best_model = random_search.best_estimator_

best_model.fit(X_train, y_train)

y_pred_best = best_model.predict(X_test)

accuracy = accuracy_score(y_test, y_pred_best)

print(f"Best Model Accuracy: {accuracy:.3f}")
```

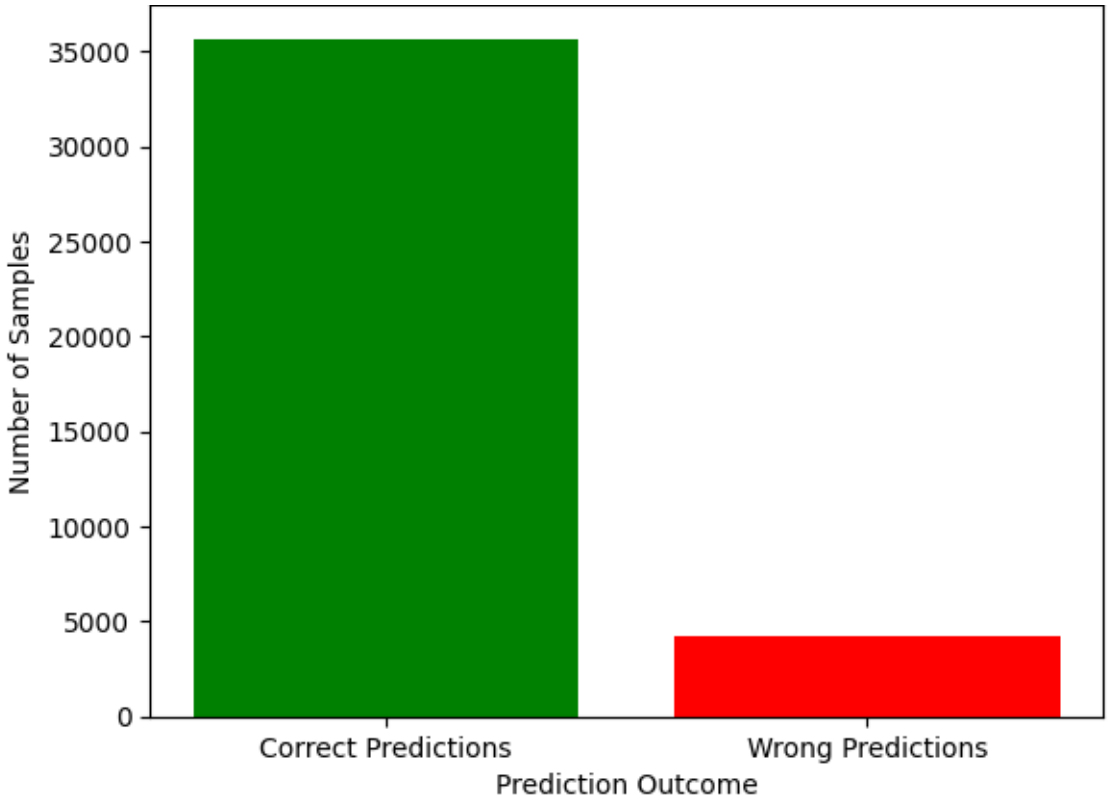
```
Best Model Accuracy: 0.893
```

ERROR ANALYSIS AND FINAL DATAFRAME

```
# Inverse transform the predicted labels to get the original class labels
predicted_labels_original = le.inverse_transform(y_pred_best)
correct_predictions = sum(y_test == y_pred_best)
wrong_predictions = len(y_test) - correct_predictions
print(f'Correct Predictions: {correct_predictions}, Wrong Predictions: {wrong_p
labels = ['Correct Predictions', 'Wrong Predictions']
values = [correct_predictions, wrong_predictions]

plt.bar(labels, values, color=['green', 'red'])
plt.title('Correct vs Wrong Predictions')
plt.xlabel('Prediction Outcome')
plt.ylabel('Number of Samples')
plt.show()
```

```
Correct Predictions: 35671, Wrong Predictions: 4271
```





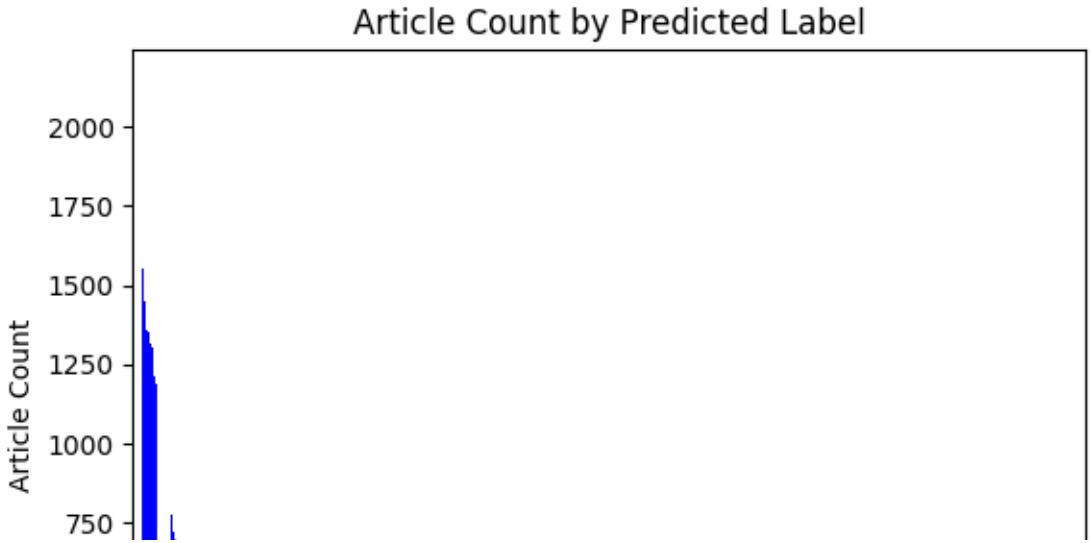

```
#final dataframe with text and predicted labels
final_df = pd.DataFrame({'Content': X_test, 'Predicted_Labels': predicted_labels_original, 'Actual_Labels': le.inve

final_df.head()
```

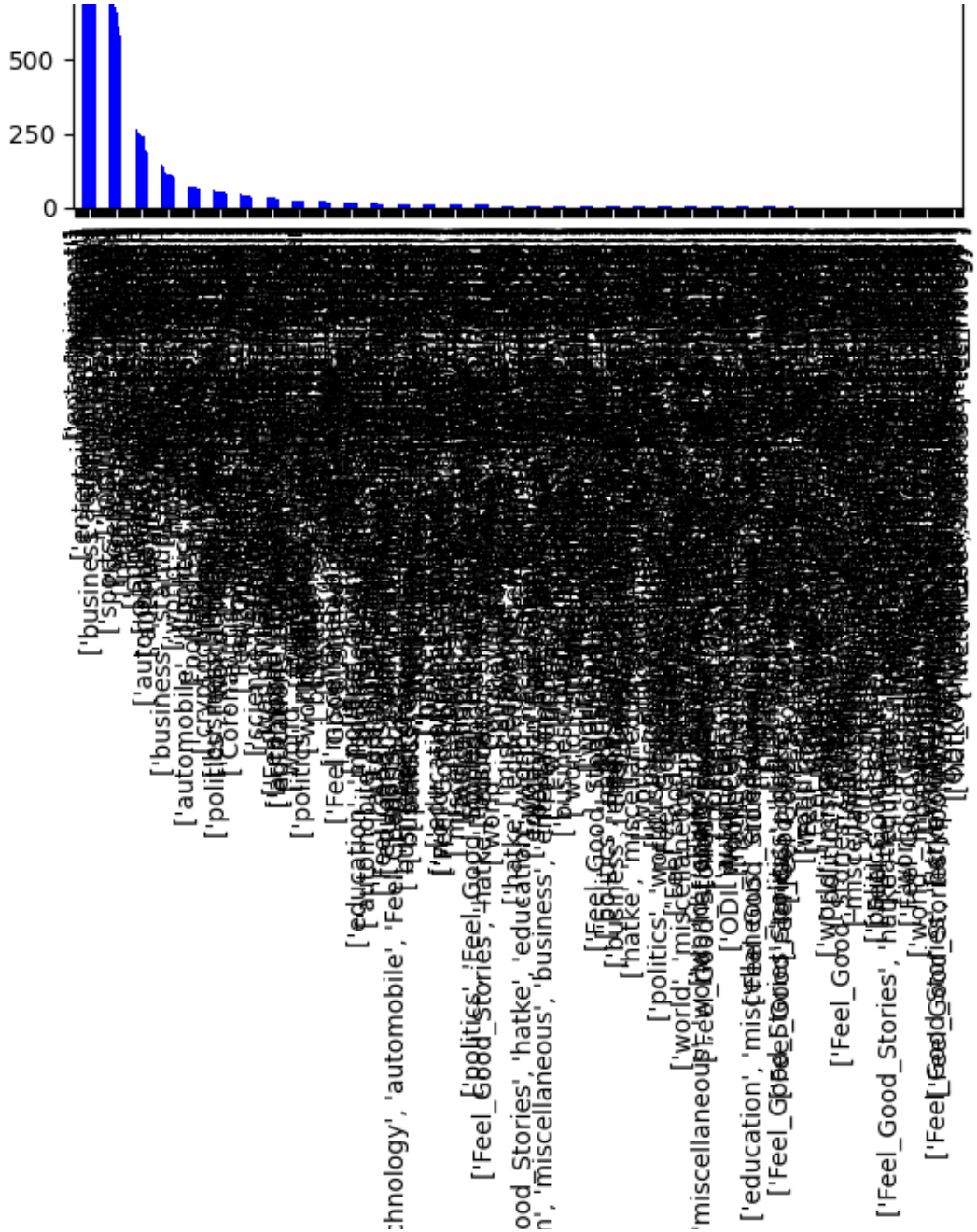
	Content	Predicted Labels	Actual Labels

23219	congress sandeep dikshit monday asked aimim ch...	['politics']	['politics']
48597	square ceo alyssa henry stepping nine years pa...	['business', 'startup']	['business', 'startup']
162019	every womans menstruation cycle reflects certa...	['Health___Fitness']	['Health___Fitness']
35753	jonty rhodes superman runout inzamamulhaq worl...	['sports', 'ODI_World_Cup_2023']	['sports', 'ODI_World_Cup_2023']
108844	cartrade technologies reported yoy surge profi...	['business', 'startup', 'technology']	['business', 'startup', 'technology']

Next steps: Generate code with `final_df` View recommended plots

```
# Article Count by Predicted Label

import matplotlib.pyplot as plt
final_df['Predicted_Labels'].value_counts().plot(kind='bar', color='blue')
plt.xlabel('Predicted Label')
plt.ylabel('Article Count')
_ = plt.title('Article Count by Predicted Label')
```




500
250
0

Key Insights and *Takeways*:

1. Using BOW rather that TF-IDF bec BOW have highest acc than TF-IDF
2. Use those hyperparamters [max_feature=None , ngram_range=(1,2) , alpha=0.1116] to get the accuracy 0.984

3. After using the best model the Correct predictions = 39300 (thats very good) & the wrong predictions are : 642
4. Using other models may be give high accuracy and Correct predections