

# NLP Classification of Articles

## Import Necessary Packages

```
In [1]: 1 import pandas as pd
2 import numpy as np
3
4 #for text pre-processing
5 import re, string
6 import nltk
7 from nltk.tokenize import word_tokenize
8 from nltk.corpus import stopwords
9 from nltk.tokenize import word_tokenize
10 from nltk.stem import SnowballStemmer
11 from nltk.corpus import wordnet
12 from nltk.stem import WordNetLemmatizer
13
14 nltk.download('punkt')
15 nltk.download('averaged_perceptron_tagger')
16 nltk.download('wordnet')
17
18 #for model-building
19 from sklearn.model_selection import train_test_split
20 from sklearn.linear_model import LogisticRegression
21 from sklearn.naive_bayes import MultinomialNB
22 from sklearn.metrics import classification_report, f1_score, accuracy_score
23 from sklearn.metrics import roc_curve, auc, roc_auc_score
24
25 # bag of words
26 from sklearn.feature_extraction.text import TfidfVectorizer
27 from sklearn.feature_extraction.text import CountVectorizer
28
29 #for word embedding
30 import gensim
31 from gensim.models import Word2Vec
```

```
[nltk_data] Downloading package punkt to
[nltk_data]   C:\Users\tp511\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]   C:\Users\tp511\AppData\Roaming\nltk_data...
[nltk_data]   Package averaged_perceptron_tagger is already up-to-
[nltk_data]   date!
[nltk_data] Downloading package wordnet to
[nltk_data]   C:\Users\tp511\AppData\Roaming\nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
```

## Load dataset

```
In [2]: 1 data_train = pd.read_csv('data_train.txt', header=None)
2 data_train.columns = ['text']
3 data_test = pd.read_csv('data_valid.txt', header=None)
4 data_test.columns = ['text']
5 labels_train = pd.read_csv('labels_train_original.txt', header=None)
6 labels_train.columns = ["label"]
7 labels_test = pd.read_csv('labels_valid_original.txt', header=None)
8 labels_test.columns = ["label"]
```

```
In [3]: 1 data_train
```

Out[3]:

	text
0	the sign in front of the steepled church read ...
1	lindsey larsen a soprano and samuel ramey the ...
2	to the editor sylvia ann hewlett 's book creat...
3	illinois tool works inc glenview ill a maker o...
4	to the editor robert schaeffer op ed feb 19 ex...
...	...
1995	to the editor re invective 's comeback by will...
1996	a critic 's notebook article on aug 2 about th...
1997	mike shanahan the winning coach in super bowls...
1998	dear diary at a recent performance of the full...
1999	for a half the jets had survived hanging close...

2000 rows × 1 columns

```
In [4]: 1 df_train = pd.DataFrame()
2 df_train['text'] = data_train
3 df_train['orig_label'] = labels_train
4 df_train
```

Out[4]:

	text	orig_label
0	the sign in front of the steepled church read ...	News
1	lindsey larsen a soprano and samuel ramey the ...	Classifieds
2	to the editor sylvia ann hewlett 's book creat...	Opinion
3	illinois tool works inc glenview ill a maker o...	News
4	to the editor robert schaeffer op ed feb 19 ex...	Opinion
...	...	...
1995	to the editor re invective 's comeback by will...	Opinion
1996	a critic 's notebook article on aug 2 about th...	Features
1997	mike shanahan the winning coach in super bowls...	News
1998	dear diary at a recent performance of the full...	Classifieds
1999	for a half the jets had survived hanging close...	News

2000 rows × 2 columns

```
In [5]: 1 df_test = pd.DataFrame()
2 df_test['text'] = data_test
3 df_test['orig_label'] = labels_test
4 df_test
```

Out[5]:

	text	orig_label
0	to the editor re restructuring for security by...	Opinion
1	to the editor in small town gay america op ed ...	Opinion
2	don king the boxing promoter has stated that m...	Opinion
3	to the editor bill keller god and george w bus...	Opinion
4	andres rios stood in front of il monello and r...	News
...	...	...
1995	there are plenty of lines in shakespeare 's wo...	Features
1996	to the editor sadly we appear as hypocrites to...	Opinion
1997	the baseball fates played a cruel joke on the ...	News
1998	under pressure from congress to explain lapses...	Classifieds
1999	the cops are not backing off not one bit never...	News

2000 rows × 2 columns

```
In [6]: ▶ 1 # Converting labels to numeric
2 def label_conv(dataset):
3     dataset['target'] = None
4     for i in range(len(dataset)):
5         if dataset['orig_label'][i] == "News":
6             dataset['target'][i] = 0
7         elif dataset['orig_label'][i] == "Opinion":
8             dataset['target'][i] = 1
9         elif dataset['orig_label'][i] == "Classifieds":
10            dataset['target'][i] = 2
11        elif dataset['orig_label'][i] == "Features":
12            dataset['target'][i] = 3
13
14 label_conv(df_train)
15 label_conv(df_test)
```

```
In [7]: ▶ 1 # Number of observations for each class in training dataset
2 print(df_train['target'].value_counts())

2    523
0    504
3    488
1    485
Name: target, dtype: int64
```

## Pre-process Text

In [8]: ▶

```
1 #convert to lowercase, strip and remove punctuations
2 def preprocess(text):
3     text = text.lower()
4     text=text.strip()
5     text=re.compile('<.*?>').sub('', text)
6     text = re.compile('[%s]' % re.escape(string.punctuation)).sub(' ', text)
7     text = re.sub('\s+', ' ', text)
8     text = re.sub(r'\[[0-9]*\]', ' ', text)
9     text=re.sub(r'^\w\s', '', str(text).lower().strip())
10    text = re.sub(r'\d', ' ', text)
11    text = re.sub(r'\s+', ' ', text)
12    return text
13
14
15 # STOPWORD REMOVAL
16 def stopword(string):
17     a= [i for i in string.split() if i not in stopwords.words('english')]
18     return ' '.join(a)
19
20 #LEMMATIZATION
21 # Initialize the Lemmatizer
22 wl = WordNetLemmatizer()
23
24 # This is a helper function to map NTLK position tags
25 def get_wordnet_pos(tag):
26     if tag.startswith('J'):
27         return wordnet.ADJ
28     elif tag.startswith('V'):
29         return wordnet.VERB
30     elif tag.startswith('N'):
31         return wordnet.NOUN
32     elif tag.startswith('R'):
33         return wordnet.ADV
34     else:
35         return wordnet.NOUN
36
37 # Tokenize the sentence
38 def lemmatizer(string):
39     word_pos_tags = nltk.pos_tag(word_tokenize(string)) # Get position tags
40     a=[wl.lemmatize(tag[0], get_wordnet_pos(tag[1])) for idx, tag in enumerate(word_pos_tags)]
41     return " ".join(a)
```

```
In [9]: 1 # Final pre-processing
2 def finalpreprocess(string):
3     return lemmatizer(stopword(preprocess(string)))
4
5 df_train['clean_text'] = df_train['text'].apply(lambda x: finalpreprocess(x))
6 df_train.head()
```

Out[9]:

	text	orig_label	target	clean_text
0	the sign in front of the steepled church read ...	News	0	sign front steepled church read sunday sermon ...
1	lindsey larsen a soprano and samuel ramey the ...	Classifieds	2	lindsey larsen soprano samuel ramey bass marry...
2	to the editor sylvia ann hewlett 's book creat...	Opinion	1	editor sylvia ann hewlett book create life ing...
3	illinois tool works inc glenview ill a maker o...	News	0	illinois tool work inc glenview ill maker ever...
4	to the editor robert schaeffer op ed feb 19 ex...	Opinion	1	editor robert schaeffer op ed feb explain dear...

## Vectorize the text

```
In [11]: 1 #SPLITTING THE TRAINING DATASET INTO TRAIN AND TEST
2 X_train, X_test, y_train, y_test = train_test_split(df_train["clean_text"],
3 y_test = y_test.astype('int')
4 y_train = y_train.astype('int')
5
6 #Word2Vec
7 # Word2Vec runs on tokenized sentences
8 X_train_tok= [nltk.word_tokenize(i) for i in X_train]
9 X_test_tok= [nltk.word_tokenize(i) for i in X_test]
```

```

In [12]: ▶ 1 #Tf-Idf
2 tfidf_vectorizer = TfidfVectorizer(use_idf=True)
3 X_train_vectors_tfidf = tfidf_vectorizer.fit_transform(X_train)
4 X_test_vectors_tfidf = tfidf_vectorizer.transform(X_test)
5
6 #building Word2Vec model
7 class MeanEmbeddingVectorizer(object):
8     def __init__(self, word2vec):
9         self.word2vec = word2vec
10        # if a text is empty we should return a vector of zeros
11        # with the same dimensionality as all the other vectors
12        self.dim = len(next(iter(word2vec.values()))))
13    def fit(self, X, y):
14        return self
15    def transform(self, X):
16        return np.array([
17            np.mean([self.word2vec[w] for w in words if w in self.word2vec
18                    or [np.zeros(self.dim)], axis=0)
19            for words in X])
20
21 df_train['clean_text_tok']=[nltk.word_tokenize(i) for i in df_train['clean_text']]
22 model = Word2Vec(df_train['clean_text_tok'],min_count=1)
23 w2v = dict(zip(model.wv.index_to_key, model.wv.vectors))
24 modelw = MeanEmbeddingVectorizer(w2v)
25
26 # converting text to numerical data using Word2Vec
27 X_train_vectors_w2v = modelw.transform(X_train_tok)
28 X_test_vectors_w2v = modelw.transform(X_test_tok)

```

## Machine Learning Step

```
In [14]: ► 1 #FITTING THE CLASSIFICATION MODEL using Logistic Regression(tf-idf)
2 lr_tfidf = LogisticRegression(solver = 'liblinear', C=10, penalty = 'l2')
3 lr_tfidf.fit(X_train_vectors_tfidf, y_train)
4
5 #Predict y value for test dataset
6 y_predict = lr_tfidf.predict(X_test_vectors_tfidf)
7 y_prob = lr_tfidf.predict_proba(X_test_vectors_tfidf)[:,-1]
8
9 print(classification_report(y_test,y_predict))
10 print('Confusion Matrix:',confusion_matrix(y_test, y_predict))
11
12 # fpr, tpr, thresholds = roc_curve(y_test, y_prob)
13 # roc_auc = auc(fpr, tpr)
14 # print('AUC:', roc_auc)
```

	precision	recall	f1-score	support
0	0.69	0.66	0.67	96
1	0.79	0.91	0.85	101
2	0.70	0.67	0.68	114
3	0.55	0.53	0.54	89
accuracy			0.69	400
macro avg	0.69	0.69	0.69	400
weighted avg	0.69	0.69	0.69	400

```
Confusion Matrix: [[63  9 13 11]
 [ 4 92  0  5]
 [14  2 76 22]
 [10 13 19 47]]
```



```
In [15]: ► 1 #FITTING THE CLASSIFICATION MODEL using Logistic Regression (W2v)
2 lr_w2v=LogisticRegression(solver = 'liblinear', C=10, penalty = 'l2')
3 lr_w2v.fit(X_train_vectors_w2v, y_train) #model
4
5 #Predict y value for test dataset
6 y_predict = lr_w2v.predict(X_test_vectors_w2v)
7 y_prob = lr_w2v.predict_proba(X_test_vectors_w2v)[: ,1]
8
9 print(classification_report(y_test,y_predict))
10 print('Confusion Matrix:',confusion_matrix(y_test, y_predict))
11
12 # fpr, tpr, thresholds = roc_curve(y_test, y_prob)
13 # roc_auc = auc(fpr, tpr)
14 # print('AUC:', roc_auc)
```

	precision	recall	f1-score	support
0	0.58	0.54	0.56	96
1	0.64	0.80	0.71	101
2	0.55	0.49	0.52	114
3	0.48	0.44	0.46	89
accuracy			0.57	400
macro avg	0.56	0.57	0.56	400
weighted avg	0.56	0.57	0.56	400

```
Confusion Matrix: [[52 11 19 14]
 [ 9 81  4  7]
 [19 18 56 21]
 [10 17 23 39]]
```

```
In [16]: 1 #FITTING THE CLASSIFICATION MODEL using Naive Bayes(tf-idf)
2 nb_tfidf = MultinomialNB()
3 nb_tfidf.fit(X_train_vectors_tfidf, y_train)
4
5 #Predict y value for test dataset
6 y_predict = nb_tfidf.predict(X_test_vectors_tfidf)
7 y_prob = nb_tfidf.predict_proba(X_test_vectors_tfidf)[: ,1]
8
9 print(classification_report(y_test,y_predict))
10 print('Confusion Matrix:',confusion_matrix(y_test, y_predict))
11
12 # fpr, tpr, thresholds = roc_curve(y_test, y_prob)
13 # roc_auc = auc(fpr, tpr)
14 # print('AUC:', roc_auc)
```

	precision	recall	f1-score	support
0	0.62	0.64	0.63	96
1	0.77	0.83	0.80	101
2	0.70	0.61	0.65	114
3	0.56	0.60	0.58	89
accuracy			0.67	400
macro avg	0.66	0.67	0.66	400
weighted avg	0.67	0.67	0.67	400

```
Confusion Matrix: [[61 10 16  9]
 [ 8 84  1  8]
 [17  4 69 24]
 [12 11 13 53]]
```

```
In [17]: 1 #Pre-processing the new dataset
2 df_test['clean_text'] = df_test['text'].apply(lambda x: finalpreprocess(x))
3 X_test=df_test['clean_text']
4
5 #converting words to numerical data using tf-idf
6 X_vector=tfidf_vectorizer.transform(X_test)
7
8 #use the best model to predict 'target' value for the new dataset
9 y_predict = lr_tfidf.predict(X_vector)
10 y_prob = lr_tfidf.predict_proba(X_vector)[: ,1]
11 df_test['predict_prob']= y_prob
12 df_test['target']= y_predict
13 final=df_test[['clean_text','target']].reset_index(drop=True)
14 print(final.head())
```

	clean_text	target
0	editor restructure security warren rudman gary...	1
1	editor small town gay america op ed nov adam g...	1
2	king box promoter state recent presidential ex...	1
3	editor bill keller god george w bush column ma...	1
4	andres rios stand front il monello recite dish...	2

