Tokenización

Instalación de Librerías

In [14]:
```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn import metrics
```

In [15]:
```
df = pd.read_csv('df_total.csv',encoding='UTF-8')
df.head()
```

Out[15]:

	url	news	Type
0	https://www.larepublica.co/redirect/post/3201905	Durante el foro La banca articulador empresari...	Otra
1	https://www.larepublica.co/redirect/post/3210288	El regulador de valores de China dijo el domin...	Regulaciones
2	https://www.larepublica.co/redirect/post/3240676	En una industria históricamente masculina como...	Alianzas
3	https://www.larepublica.co/redirect/post/3342889	Con el dato de marzo el IPC interanual encaden...	Macroeconomia
4	https://www.larepublica.co/redirect/post/3427208	Ayer en Cartagena se dio inicio a la versión n...	Otra

Separación de datos.

In [16]:
```
X = df['news']
y = df['Type']
print(df['Type'].value_counts())
```

```
Type
Macroeconomia     340
Alianzas          247
Innovacion        195
Regulaciones      142
Sostenibilidad    137
Otra              130
Reputacion         26
Name: count, dtype: int64
```

In [17]:
```
X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.2)
vetorizer = CountVectorizer()
```

Vectorizamos

In [18]:
```
X_train_transformed = vetorizer.fit_transform(X_train)
X_test_transformed = vetorizer.transform(X_test)
```

In [19]:
```
model = MultinomialNB()
model.fit(X_train_transformed,y_train)
y_pred = model.predict(X_test_transformed)
print(metrics.accuracy_score(y_test, y_pred))
```

```
0.7868852459016393
```

Stemming

Librerías

In [20]:
```
import nltk
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
from nltk.stem import SnowballStemmer
```

In [21]:
```
nltk.download('punkt')
nltk.download('stopwords')
```

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

Out[21]:
```
True
```

In [22]:
```
stemmer = SnowballStemmer('spanish')
```

In [23]:
```
def tokenize_and_stem(text):
    tokens = word_tokenize(text.lower())
    stems = [stemmer.stem(token) for token in tokens if token.isalpha()]
    return ' '.join(stems)
```

In [24]:
```
df['news_stemmer'] = df['news'].apply(tokenize_and_stem)
df['news_stemmer'][3]
```

Out[24]:
```
'con el dat de marz el ipc interanual encaden su decimoquint tas posit consec
ut la inflacion public por el ine se ha manten igual respect al avanc del de
marz y se situ punt por encim del dat de febrer que ascend al esos punt de di
ferent la mayor part la coloc el grup de la viviend punt por la sub de la ele
ctr y el del transport punt por el alza de los carbur tambien impuls el ipc d
e marz el aument de los preci de la restaur y los servici de aloj y al encare
c generaliz de los aliment especial del pesc y el marisc de la carn de las le
gumbr y hortaliz y de la lech el ques y los ten en cuent la rebaj del impuest
especial sobr la electr y las variacion sobr otros impuest el ipc interanual
alcanz en marz nuev decim mas que la tas general del asi lo reflej el ipc a i
mpuest constant que el ine tambien public en el marc de esta inflacion subyac
ent sin aliment no elabor ni product energet aument en marz cuatr decim hast
su valor mas alto desd septiembr de de este mod la subyacent se situ mas de s
eis punt por debaj de la tas del ipc el ultim año la calefaccion el alumbr y
la distribu de agu se han encarec los aceit y gras han elev sus preci un y el
transport personal es un mas car por el mayor cost de los carbur tambien regi
str alzas de dos digit los huev y la lech un mas car que hac un año y la carn
de ovin y el pesc fresc y congel con repunt del en ambos estan los preci por
comunidadescastillal manch se situ a la cabez con una tas de inflacion del se
gu de castill y leon aragon la rioj galici extremadur cantabri y comun valenc
ian el otro las comun dond se registr las menor sub fueron canari madr balear
asturi pais vasc y cataluñ ipc disp su tas mensual al tas mensual el ipc regi
str en marz un increment del respect a febrer su mayor alza mensual en cualqu
i mes desd cuand se camb la metodolog de esta estadist par recog mejor la evo
lu del merc echand la vist mas atras tom seri anterior el repunt mensual de m
arz es el mas elev desd agost de el terc mes de el indic de preci de consum a
rmoniz ipca situ su tas interanual en mas de dos punt por encim de la de febr
er por su part el indic adelant del ipca avanz un en tas mensual'
```

Separamos los datos en variables de entrada y etiquetas

In [25]:
```
X = df['news_stemmer']
y = df['Type']
print(df['Type'].value_counts())
```

```
Type
Macroeconomia     340
Alianzas          247
Innovacion        195
Regulaciones      142
Sostenibilidad    137
Otra              130
Reputacion         26
Name: count, dtype: int64
```

In [26]:
```
X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.2)
vetorizer = CountVectorizer()
X_train_transformed = vetorizer.fit_transform(X_train)
X_test_transformed = vetorizer.transform(X_test)
model = MultinomialNB()
model.fit(X_train_transformed,y_train)
y_pred = model.predict(X_test_transformed)
print(metrics.accuracy_score(y_test, y_pred))
```

```
0.8032786885245902
```

Lemmatization

Librerías

In [33]:
```
import spacy
nlp = spacy.load('es_core_news_sm')
```

In [34]:
```
def lemmatize_text(text):
    doc = nlp(text.lower())
    lemmas = [token.lemma_ for token in doc if token.is_alpha]
    return ' '.join(lemmas)
```

In [35]:
```
df['news_lemas'] = df['news'].apply(lemmatize_text)
```

In [36]:
```
X = df['news_lemas']
y = df['Type']
print(df['Type'].value_counts())
```

```
Type
Macroeconomia     340
Alianzas          247
Innovacion        195
Regulaciones      142
Sostenibilidad    137
Otra              130
Reputacion         26
Name: count, dtype: int64
```

In [37]:
```
X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.2)
vetorizer = CountVectorizer()
X_train_transformed = vetorizer.fit_transform(X_train)
X_test_transformed = vetorizer.transform(X_test)
model = MultinomialNB()
model.fit(X_train_transformed,y_train)
y_pred = model.predict(X_test_transformed)
print(metrics.accuracy_score(y_test, y_pred))
```

```
0.8401639344262295
```

In [ ]:
```
```