Multiclass Text Classification with

Logistic Regression Implemented with PyTorch and CE Loss

First, we will do some initialization.

Primero se importan todas las librerías necesarias para la tarea, incluyendo numpy, pandas, tqdm, torch y random. También se verifica si hay un GPU disponible para realizar operaciones, en mi caso no lo hay puesto que trabajo en una Mac, la cual no tiene una tarjeta gráfica dedicada, por lo que las operaciones se realizan directamente en el cpu. Además, se define una semilla aleatoria para obtener resultados consistentes.

In [1]:

```
import random
import torch
import numpy as np
import pandas as pd
from tqdm.notebook import tqdm

# enable tqdm in pandas
tqdm.pandas()

# set to True to use the gpu (if there is one available)
use_gpu = True

# select device
device = torch.device('cuda' if use_gpu and torch.cuda.is_available() else '
print(f'device: {device.type}')

# random seed
seed = 1234

# set random seed
if seed is not None:
    print(f'random seed: {seed}')
    random.seed(seed)
    np.random.seed(seed)
    torch.manual_seed(seed)
```

```
device: cpu
random seed: 1234
```

We will be using the AG's News Topic Classification Dataset. It is stored in two CSV files: `train.csv` and `test.csv`, as well as a `classes.txt` that stores the labels of the classes to predict.

First, we will load the training dataset using pandas and take a quick look at how the data.

Ahora se carga el archivo con la información y se le asigna un nombre a cada columna, son 3 columnas, una representa la clase, otra el título y otra la descripción.

In [2]:

```
train_df = pd.read_csv('train.csv')
train_df.columns = ['class index', 'title', 'description']
train_df
```

Out[2]:

	class index	title	description
0	3	Wall St. Bears Claw Back Into the Black (Reuters)	Reuters - Short-sellers, Wall Street's dwindli...
1	3	Carlyle Looks Toward Commercial Aerospace (Reu...	Reuters - Private investment firm Carlyle Grou...
2	3	Oil and Economy Cloud Stocks' Outlook (Reuters)	Reuters - Soaring crude prices plus worries\ab...
3	3	Iraq Halts Oil Exports from Main Southern Pipe...	Reuters - Authorities have halted oil export\f...
4	3	Oil prices soar to all-time record, posing new...	AFP - Tearaway world oil prices, toppling reco...
...	...	...	...
119995	1	Pakistan's Musharraf Says Won't Quit as Army C...	KARACHI (Reuters) - Pakistani President Perve...
119996	2	Renteria signing a top-shelf deal	Red Sox general manager Theo Epstein acknowled...
119997	2	Saban not going to Dolphins yet	The Miami Dolphins will put their courtship of...
119998	2	Today's NFL games	PITTSBURGH at NY GIANTS Time: 1:30 p.m. Line: ...
119999	2	Nets get Carter from Raptors	INDIANAPOLIS -- All-Star Vince Carter was trad...

120000 rows × 3 columns

The dataset consists of 120,000 examples, each consisting of a class index, a title, and a description. The class labels are distributed in a separated file. We will add the labels to the dataset so that we can interpret the data more easily. Note that the label indexes are one-based, so we need to subtract one to retrieve them from the list.

Luego se crea una nueva columna donde se utilizan los valores de 'class index' para asignarle un string de acuerdo con su valor.

In [ ]:

```python
# Estos son los labels que se utilizan
labels = ['World', 'Sports', 'Business', 'Sci/Tech']
classes = train_df['class index'].map(lambda i: labels[i-1]) # Función que m
train_df.insert(1, 'class', classes) # se inserta la columna
train_df
```

Out[ ]:

	class index	class	title	description
0	3	Business	Wall St. Bears Claw Back Into the Black (Reuters)	Reuters - Short-sellers, Wall Street's dwindli...
1	3	Business	Carlyle Looks Toward Commercial Aerospace (Reu...	Reuters - Private investment firm Carlyle Grou...
2	3	Business	Oil and Economy Cloud Stocks' Outlook (Reuters)	Reuters - Soaring crude prices plus worries\ab...
3	3	Business	Iraq Halts Oil Exports from Main Southern Pipe...	Reuters - Authorities have halted oil export\f...
4	3	Business	Oil prices soar to all-time record, posing new...	AFP - Tearaway world oil prices, toppling reco...
...	...	...	...	...
119995	1	World	Pakistan's Musharraf Says Won't Quit as Army C...	KARACHI (Reuters) - Pakistani President Perve...
119996	2	Sports	Renteria signing a top-shelf deal	Red Sox general manager Theo Epstein acknowled...
119997	2	Sports	Saban not going to Dolphins yet	The Miami Dolphins will put their courtship of...
119998	2	Sports	Today's NFL games	PITTSBURGH at NY GIANTS Time: 1:30 p.m. Line: ...
119999	2	Sports	Nets get Carter from Raptors	INDIANAPOLIS -- All-Star Vince Carter was trad...

120000 rows × 4 columns

Let's inspect how balanced our examples are by using a bar plot.

En la siguiente grafica se muestra que las clases del dataset estan balanceadas para que no haya un bias en los resultados.

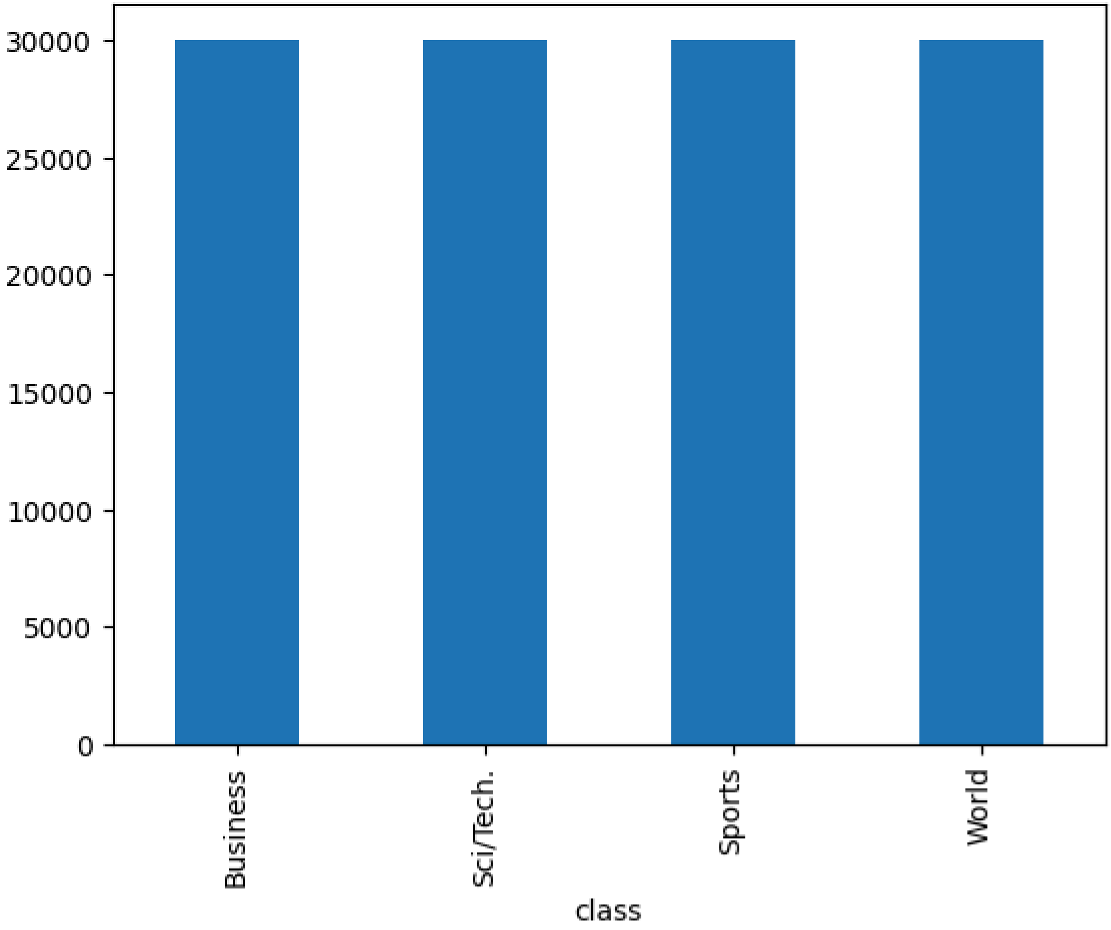
In [4]:

```python
pd.value_counts(train_df['class']).plot.bar()
```

```
/var/folders/dz/d8yvxk91663f2sr57qnjf7240000gn/T/ipykernel_20171/1245903889.
py:1: FutureWarning: pandas.value_counts is deprecated and will be removed i
n a future version. Use pd.Series(obj).value_counts() instead.
  pd.value_counts(train_df['class']).plot.bar()
```

Out[4]:

```
<Axes: xlabel='class'>
```



The classes are evenly distributed. That's great!

However, the text contains some spurious backslashes in some parts of the text. They are meant to represent newlines in the original text. An example can be seen below, between the words "dwindling" and "band".

Aqui simplemente se imprime el primer dato registrado para la columna de descripción

In [5]:

```
print(train_df.loc[0, 'description'])
```

```
Reuters - Short-sellers, Wall Street's dwindling\band of ultra-cynics, are s
eeing green again.
```

We will replace the backslashes with spaces on the whole column using pandas replace method.

Aqui se hace un pequeño procesamiento de los datos donde se convierte el texto a minúsculas y juntan los contenidos de las columnas 'title' y 'description' en una nueva columna llamada 'text', además, se reemplazan los backslashes por espacios.

In [6]:

```
title = train_df['title'].str.lower()
descr = train_df['description'].str.lower()
text = title + " " + descr
```

```
train_df['text'] = text.str.replace('\\', ' ', regex=False)
train_df
```

Out[6]:

	class index	class	title	description	text
0	3	Business	Wall St. Bears Claw Back Into the Black (Reuters)	Reuters - Short-sellers, Wall Street's dwindli...	wall st. bears claw back into the black (reute...
1	3	Business	Carlyle Looks Toward Commercial Aerospace (Reu...	Reuters - Private investment firm Carlyle Grou...	carlyle looks toward commercial aerospace (reu...
2	3	Business	Oil and Economy Cloud Stocks' Outlook (Reuters)	Reuters - Soaring crude prices plus worries\ab...	oil and economy cloud stocks' outlook (reuters...
3	3	Business	Iraq Halts Oil Exports from Main Southern Pipe...	Reuters - Authorities have halted oil export\f...	iraq halts oil exports from main southern pipe...
4	3	Business	Oil prices soar to all-time record, posing new...	AFP - Tearaway world oil prices, toppling reco...	oil prices soar to all-time record, posing new...
...	...	...	...	...	...
119995	1	World	Pakistan's Musharraf Says Won't Quit as Army C...	KARACHI (Reuters) - Pakistani President Perve...	pakistan's musharraf says won't quit as army c...
119996	2	Sports	Renteria signing a top-shelf deal	Red Sox general manager Theo Epstein acknowled...	renteria signing a top-shelf deal red sox gene...
119997	2	Sports	Saban not going to Dolphins yet	The Miami Dolphins will put their courtship of...	saban not going to dolphins yet the miami dolp...
119998	2	Sports	Today's NFL games	PITTSBURGH at NY GIANTS Time: 1:30 p.m. Line: ...	today's nfl games pittsburgh at ny giants time...
119999	2	Sports	Nets get Carter from Raptors	INDIANAPOLIS -- All-Star Vince Carter was trad...	nets get carter from raptors indianapolis -- a...

120000 rows × 5 columns

Now we will proceed to tokenize the title and description columns using NLTK's word_tokenize(). We will add a new column to our dataframe with the list of tokens.

Aquí se utiliza la columna 'text' y se crea una nueva columna 'tokens' utilizando la librerie nltk, los tokens simplemente son las palabras.

In [7]:

```
from nltk.tokenize import word_tokenize

train_df['tokens'] = train_df['text'].progress_map(word_tokenize)
train_df
```

```
 0%|          | 0/120000 [00:00<?, ?it/s]
```

Out[7]:

	class index	class	title	description	text	tokens
0	3	Business	Wall St. Bears Claw Back Into the Black (Reuters)	Reuters - Short-sellers, Wall Street's dwindli...	wall st. bears claw back into the black (reute...	[wall, st., bears, claw, back, into, the, blac...
1	3	Business	Carlyle Looks Toward Commercial Aerospace (Reu...	Reuters - Private investment firm Carlyle Grou...	carlyle looks toward commercial aerospace (reu...	[carlyle, looks, toward, commercial, aerospace...
2	3	Business	Oil and Economy Cloud Stocks' Outlook (Reuters)	Reuters - Soaring crude prices plus worries\ab...	oil and economy cloud stocks' outlook (reuters...	[oil, and, economy, cloud, stocks, ', outlook,...
3	3	Business	Iraq Halts Oil Exports from Main Southern Pipe...	Reuters - Authorities have halted oil export\f...	iraq halts oil exports from main southern pipe...	[iraq, halts, oil, exports, from, main, southe...
4	3	Business	Oil prices soar to all-time record, posing new...	AFP - Tearaway world oil prices, toppling reco...	oil prices soar to all-time record, posing new...	[oil, prices, soar, to, all-time, record, ,, p...
...	...	...	...	...	...	...
119995	1	World	Pakistan's Musharraf Says Won't Quit as Army C...	KARACHI (Reuters) - Pakistani President Perve...	pakistan's musharraf says won't quit as army c...	[pakistan, 's, musharraf, says, wo, n't, quit,...
119996	2	Sports	Renteria signing a top-shelf deal	Red Sox general manager Theo Epstein acknowled...	renteria signing a top-shelf deal red sox gene...	[renteria, signing, a, top-shelf, deal, red, s...
119997	2	Sports	Saban not going to Dolphins yet	The Miami Dolphins will put their courtship of...	saban not going to dolphins yet the miami dolp...	[saban, not, going, to, dolphins, yet, the, mi...
119998	2	Sports	Today's NFL games	PITTSBURGH at NY GIANTS Time: 1:30 p.m. Line: ...	today's nfl games pittsburgh at ny giants time...	[today, 's, nfl, games, pittsburgh, at, ny, gi...
119999	2	Sports	Nets get Carter from	INDIANAPOLIS -- All-Star	nets get carter from	[nets, get, carter, from,

class index	class	title	description	text	tokens
		Raptors	Vince Carter was trad...	raptors indianapolis -- a...	raptors, indianapoli...

120000 rows × 6 columns

Now we will create a vocabulary from the training data. We will only keep the terms that repeat beyond some threshold established below.

Se arma un vocabulario, a este solo se agregan palabras que aparecen más de 10 veces a lo largo de todos los documentos, además se construyen la lista 'id_to_token' y el diccionario 'token_to_id'. La lista permite obtener el token si se tiene el id y el diccionario permite obtener el id si se tiene el token.

In [8]:

```
threshold = 10
tokens = train_df['tokens'].explode().value_counts()
tokens = tokens[tokens > threshold]
id_to_token = ['[UNK]'] + tokens.index.tolist()
token_to_id = {w:i for i,w in enumerate(id_to_token)}
vocabulary_size = len(id_to_token)
print(f'vocabulary size: {vocabulary_size:,}')
```

```
vocabulary size: 19,668
```

Aquí se define una función que permite contar las apariciones de cada token en un texto haciendo uso de el diccionario 'token_to_id' del paso anterior. El conteo de las apariciones de cada token luego se agrega en una columna llamada 'features'

In [9]:

```
from collections import defaultdict

def make_feature_vector(tokens, unk_id=0):
    vector = defaultdict(int)
    for t in tokens:
        i = token_to_id.get(t, unk_id)
        vector[i] += 1
    return vector

train_df['features'] = train_df['tokens'].progress_map(make_feature_vector)
train_df
```

```
  0%|          | 0/120000 [00:00<?, ?it/s]
```

Out[9]:

	class index	class	title	description	text	tokens	features
0	3	Business	Wall St. Bears Claw Back Into the Black (Reuters)	Reuters - Short-sellers, Wall Street's dwindli...	wall st. bears claw back into the black (reute...	[wall, st., bears, claw, back, into, the, blac...	{427: 2, 566: 1, 1609: 1, 15347: 1, 120: 1, 73...
1	3	Business	Carlyle Looks Toward Commercial Aerospace (Reu...	Reuters - Private investment firm Carlyle Grou...	carlyle looks toward commercial aerospace (reu...	[carlyle, looks, toward, commercial, aerospace...	{16371: 2, 1077: 1, 854: 1, 1287: 1, 4243: 1, ...
2	3	Business	Oil and Economy Cloud Stocks' Outlook (Reuters)	Reuters - Soaring crude prices plus worries\ab...	oil and economy cloud stocks' outlook (reuters...	[oil, and, economy, cloud, stocks, ', outlook,...	{66: 1, 9: 2, 351: 2, 4575: 1, 158: 1, 116: 1,...
3	3	Business	Iraq Halts Oil Exports from Main Southern Pipe...	Reuters - Authorities have halted oil export\f...	iraq halts oil exports from main southern pipe...	[iraq, halts, oil, exports, from, main, southe...	{77: 2, 7404: 1, 66: 3, 1785: 1, 32: 2, 900: 2...
4	3	Business	Oil prices soar to all-time record, posing new...	AFP - Tearaway world oil prices, toppling reco...	oil prices soar to all-time record, posing new...	[oil, prices, soar, to, all-time, record, ,, p...	{66: 2, 99: 2, 4376: 1, 4: 2, 3590: 1, 149: 1,...
...	...	...	...	...	...	...	...
119995	1	World	Pakistan's Musharraf Says Won't Quit as Army C...	KARACHI (Reuters) - Pakistani President Perve...	pakistan's musharraf says won't quit as army c...	[pakistan, 's, musharraf, says, wo, n't, quit,...	{383: 1, 23: 1, 1625: 2, 91: 1, 1804: 1, 285: ...
119996	2	Sports	Renteria signing a top-shelf deal	Red Sox general manager Theo Epstein acknowled...	renteria signing a top-shelf deal red sox gene...	[renteria, signing, a, top-shelf, deal, red, s...	{8468: 2, 2634: 1, 5: 4, 0: 3, 127: 1, 203: 3,...
119997	2	Sports	Saban not going to Dolphins yet	The Miami Dolphins will put their courtship of...	saban not going to dolphins yet the miami dolp...	[saban, not, going, to, dolphins, yet, the, mi...	{7747: 2, 68: 1, 660: 1, 4: 2, 1440: 2, 704: 1...

	class index	class	title	description	text	tokens	features
119998	2	Sports	Today's NFL games	PITTSBURGH at NY GIANTS Time: 1:30 p.m. Line: ...	today's nfl games pittsburgh at ny giants time...	[today, 's, nfl, games, pittsburgh, at, ny, gi...	{106: 1, 23: 1, 728: 1, 225: 1, 1588: 1, 22: 1...
119999	2	Sports	Nets get Carter from Raptors	INDIANAPOLIS -- All-Star Vince Carter was trad...	nets get carter from raptors indianapolis -- a...	[nets, get, carter, from, raptors, indianapoli...	{2163: 2, 226: 1, 2403: 2, 32: 1, 2994: 2, 219...

120000 rows × 7 columns

Ahora se define una función para convertir el vector de 'features' en un vector del tamaño del vocabulario, se juntan estos vectores en 'X_train' con sus respectivas clases en 'y_train'. Luego estas dos variables se convierten en tensores utilizando la libreria torch, ya que este es el formato que se debe utilizar para entrenar los modelos.

In [10]:

```python
def make_dense(feats):
    x = np.zeros(vocabulary_size)
    for k,v in feats.items():
        x[k] = v
    return x

X_train = np.stack(train_df['features'].progress_map(make_dense))
y_train = train_df['class index'].to_numpy() - 1

X_train = torch.tensor(X_train, dtype=torch.float32)
y_train = torch.tensor(y_train)
```

```
 0%|          | 0/120000 [00:00<?, ?it/s]
```

En esta sección se definen hiperparámetros importantes como el learning rate, el número de épocas, el número de datos, número de features y cantidad de labels.

También se defin el tipo de modelo (lineal) y se mueve a el device apropiado (cpu en este caso), además se define la función de costo (cross entropy por ser un problema multiclase) y el optimizador (gradiente descendente estocástico).

Posteriormente se entrena el modelo, considerando el valor de la loss function para actualizar los pesos.

In [11]:

```python
from torch import nn
from torch import optim

# hyperparameters
lr = 1.0
```

```
n_epochs = 5
n_examples = X_train.shape[0]
n_feats = X_train.shape[1]
n_classes = len(labels)

# initialize the model, loss function, optimizer, and data-loader
model = nn.Linear(n_feats, n_classes).to(device)
loss_func = nn.CrossEntropyLoss()
optimizer = optim.SGD(model.parameters(), lr=lr)

# train the model
indices = np.arange(n_examples)
for epoch in range(n_epochs):
    np.random.shuffle(indices)
    for i in tqdm(indices, desc=f'epoch {epoch+1}'):
        # clear gradients
        model.zero_grad()
        # send datum to right device
        x = X_train[i].unsqueeze(0).to(device)
        y_true = y_train[i].unsqueeze(0).to(device)
        # predict label scores
        y_pred = model(x)
        # compute loss
        loss = loss_func(y_pred, y_true)
        # backpropagate
        loss.backward()
        # optimize model parameters
        optimizer.step()
```

```
epoch 1:   0%|          | 0/120000 [00:00<?, ?it/s]
epoch 2:   0%|          | 0/120000 [00:00<?, ?it/s]
epoch 3:   0%|          | 0/120000 [00:00<?, ?it/s]
epoch 4:   0%|          | 0/120000 [00:00<?, ?it/s]
epoch 5:   0%|          | 0/120000 [00:00<?, ?it/s]
```

Next, we evaluate on the test dataset

En esta sección sigue la evaluación del modelo considerando un test dataset, a este nuevo conjunto de datos se le debe aplicar el mismo procesamiento que a los datos de entrenamiento.

In [13]:

```
# repeat all preprocessing done above, this time on the test set
test_df = pd.read_csv('test.csv')
test_df.columns = ['class index', 'title', 'description']
test_df['text'] = test_df['title'].str.lower() + " " + test_df['description'
test_df['text'] = test_df['text'].str.replace('\\', ' ', regex=False)
test_df['tokens'] = test_df['text'].progress_map(word_tokenize)
test_df['features'] = test_df['tokens'].progress_map(make_feature_vector)

X_test = np.stack(test_df['features'].progress_map(make_dense))
y_test = test_df['class index'].to_numpy() - 1
X_test = torch.tensor(X_test, dtype=torch.float32)
y_test = torch.tensor(y_test)
```

```
 0%|          | 0/7600 [00:00<?, ?it/s]
 0%|          | 0/7600 [00:00<?, ?it/s]
 0%|          | 0/7600 [00:00<?, ?it/s]
```

Luego se utiliza el modelo entrenado para realizar predicciones y se comparan estas predicciones con el valor real. De esta manera se obtienen metricas como precision, recall y el f1 score.

In [14]:

```
from sklearn.metrics import classification_report

# set model to evaluation mode
model.eval()

# don't store gradients
with torch.no_grad():
    X_test = X_test.to(device)
    y_pred = torch.argmax(model(X_test), dim=1)
    y_pred = y_pred.cpu().numpy()
    print(classification_report(y_test, y_pred, target_names=labels))
```

```
              precision    recall  f1-score   support

       World       0.95      0.82      0.88      1900
      Sports       0.93      0.98      0.95      1900
    Business       0.85      0.86      0.85      1900
   Sci/Tech.       0.82      0.89      0.85      1900

    accuracy                           0.89      7600
   macro avg       0.89      0.89      0.88      7600
weighted avg       0.89      0.89      0.88      7600
```