

Nikolas Dimitrio Badani Gasdaglis 20092  
Juan Angel Carrera Soto 20593  
Data Science  
Sección 10

### Laboratorio 4 : Minería de Textos

#### Descripción de los datos

En el set de datos se puede observar que hay un total de 7613 observaciones, en base a las 5 variables presentes en la tabla. Dichas variables consisten en las siguientes :

- Id : para ayudar a identificar cada uno de los tweets
- Keyword : Una palabra clave que ayuda a hacer una relación directa entre los tweets del set de datos
- Location : La ubicación desde donde el tweet fue enviado
- Text : El texto que contiene el tweet que fue enviado
- Target : Ayuda para identificar si el tweet es sobre un desastre verdadero (1) o no (0)

#### Análisis Exploratorio

```
Rows: 7,613
Columns: 5
$ id      <dbl> 1, 4, 5, 6, 7, 8, 10, 13, 14, 15, 16, 17, 18, 19, 20, 23, 24, ...
$ keyword <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA...
$ location <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA...
$ text    <chr> "Our Deeds are the Reason of this #earthquake May ALLAH Forgiv...
$ target  <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0,...
```

```
> summary(train)
      id      keyword      location      text
Min.   : 1    Length:7613    Length:7613    Length:7613
1st Qu.: 2734   Class :character Class :character Class :character
Median : 5408   Mode  :character   Mode  :character   Mode  :character
Mean    : 5442
3rd Qu.: 8146
Max.    :10873

      target
Min.   :0.0000
1st Qu.:0.0000
Median :0.0000
Mean    :0.4297
3rd Qu.:1.0000
Max.    :1.0000
```

Se puede observar claramente que el análisis indica que en las columnas "Keyword" y "Location" pueden haber casillas que no están rellenas con información. Por lo cual, esas casillas estarían ausentes de datos.

```
> table(train$keyword)
```

ablaze	accident	aftershock
36	35	34
airplane%20accident	ambulance	annihilated
35	38	34
annihilation	apocalypse	armageddon
29	32	42
army	arson	arsonist
34	32	34
attack	attacked	avalanche
36	35	30
battle	bioterror	bioterrorism
26	37	30
blaze	blazing	bleeding
38	34	35
blew%20up	blight	blizzard
33	32	37
blood	bloody	blown%20up
35	35	33
body%20bag	body%20bagging	body%20bags

Según el análisis realizado a la columna “Keyword”, se pudo llegar a la conclusión de que varias palabras relacionadas con el término ‘Desastre’, entre ellas la palabra “Accidente”, se utilizan varias veces a lo largo de la lista de los tweets que conforman el set de datos. También se puede notar que algunas palabras son más utilizadas que otras en los tweets.

```
> table(train$target)
```

```

  0    1
4342 3271
> |
```

```
> CrossTable(train$target)
```

```

Cell Contents
|-----|
|              N |
|      N / Table Total |
|-----|
```

Total Observations in Table: 7613

	0		1	
	-----		-----	
	4342		3271	
	0.570		0.430	
	-----		-----	



De acuerdo con las tablas y el gráfico que se realizó a la columna "Target" del set de datos, se pudo observar que hay un mayor número de tweets identificados con el número 0. Esto indica que de todos los tweets que conforman el set de datos, la mayoría (4342) no tratan acerca de un desastre verdadero. Siendo más específicos, sólo el 43% de los tweets del set de datos hablan sobre un desastre verdadero. Mientras que el 57% no lo son.

```

> describe(train)
train

  5 Variables      7613 Observations
-----
id
  n missing distinct    Info    Mean    Gmd    .05    .10    .25
  7613      0      7613      1    5442    3623    548.4   1102.2   2734.0
  .50      .75      .90      .95
  5408.0   8146.0   9818.8  10356.2

lowest :      1      4      5      6      7, highest: 10869 10870 10871 10872 10873
-----
keyword
  n missing distinct
  7552      61      221

lowest : ablaze      accident      aftershock      airplane%20acciden
t ambulance
highest: wounded      wounds      wreck      wreckage
wrecked
-----
location
  n missing distinct
  5079      2534      3279

lowest : -?s?s?j??s-      -6.152261,106.775995      ?
      ? ??????? ? ( ?? âj ? ? ? âj) ? icon by @Hashiren_3 ?
highest: zboyer@washingtontimes.com      Zeerust, South Africa      Zero Branco
      Ziam af      Zimbabwe
-----
text
  n missing distinct
  7613      0      7503
-----
target
  n missing distinct    Info    Sum    Mean    Gmd
  7613      0      2    0.735    3271    0.4297    0.4902
-----

```

El análisis realizado confirma lo que se había mencionado anteriormente. Claramente se puede observar que en las columnas “Keyword” y “Location” hay datos faltantes. Lo cual indica que estas dos columnas tienen espacios vacíos. De las 7613 variables establecidas en el set de datos, la columna de palabras clave solamente tiene perdido el 1% de dichas variables. Mientras que en la columna de location, el análisis indica que el 33% de sus variables se encuentran perdidas. Lo cual, indica que no hay información acerca de la locación desde donde fueron enviados 2534 tweets.

## Razonamiento Preprocesamiento

### 1. Convertir el texto a minúsculas:

**Razón:** Esto se hace para garantizar que la misma palabra con diferentes casos no se trate como palabras distintas. Por ejemplo, "Hola" y "hola" se tratarían como la misma palabra.

### 2. Quitar URLs:

**Razón:** Las URLs generalmente no añaden información significativa para el análisis de texto y pueden ser una fuente de ruido, especialmente en tareas como la clasificación de texto.

### 3. Quitar caracteres especiales como #, @, y apóstrofes:

**Razón:** Estos caracteres suelen ser ruidosos y no aportan mucho a la semántica del texto. Sin embargo, en algunos casos, como el análisis de sentimientos en tweets, podrían ser útiles.

### 4. Quitar emoticones:

**Razón:** Los emoticones pueden ser útiles para algunas tareas específicas como el análisis de sentimientos, pero en general pueden considerarse como ruido en el texto.

### 5. Quitar signos de puntuación:

**Razón:** Los signos de puntuación raramente aportan valor en tareas de análisis de texto y generalmente se consideran ruido.

### 6. Quitar palabras vacías (stopwords):

**Razón:** Palabras como "y", "o", "el", "la", etc., son muy comunes pero no aportan información significativa para muchas tareas de análisis de texto.

### 7. Quitar números:

**Razón:** Los números pueden ser útiles para ciertas tareas, pero en muchos casos son irrelevantes. Por ejemplo, en el análisis de sentimientos, los números raramente aportan algún sentimiento.

## Procesamiento en python:

```
# Importing necessary libraries for text preprocessing
import re
import string
from nltk.corpus import stopwords

# Function to preprocess text
def preprocess_text(text):
    # Convert text to lowercase
    text = text.lower()

    # Remove URLs
    text = re.sub(r'http\S+|www\S+|https\S+', '', text, flags=re.MULTILINE)

    # Remove special characters like #, @, and apostrophes
    text = re.sub(r'\@|_|#\w+|\'[\w\d\s]+|[\w\d\s]+\'', '', text)
```

```

# Remove emoticons
text = text.encode('ascii', 'ignore').decode('ascii')

# Remove punctuation
text = text.translate(str.maketrans('', '', string.punctuation))

# Remove stopwords
stop_words = set(stopwords.words('english'))
text = ' '.join([word for word in text.split() if word not in
stop_words])

# Remove numbers (we can customize this further based on
domain-specific needs)
text = re.sub(r'\d+', '', text)

return text

# Applying the preprocessing function to the 'text' column
df['preprocessed_text'] = df['text'].apply(preprocess_text)

# Displaying the first few rows to see the changes
df[['text', 'preprocessed_text']].head()

```

## Modelos de LSTM:

```
import torch.nn as nn
```

```

class TweetClassifier(nn.Module):
    def __init__(self, vocab_size, embedding_dim, hidden_dim):
        super(TweetClassifier, self).__init__()

        # Embedding Layer
        self.embedding = nn.Embedding(vocab_size, embedding_dim)

        # LSTM Layer
        self.lstm = nn.LSTM(embedding_dim, hidden_dim,
batch_first=True)

        # Fully Connected Layer
        self.fc = nn.Linear(hidden_dim, 1)

        # Sigmoid Activation

```

```

self.sigmoid = nn.Sigmoid()

def forward(self, x):
    x = self.embedding(x)
    lstm_out, _ = self.lstm(x)
    lstm_out = lstm_out[:, -1, :]
    out = self.fc(lstm_out)
    out = self.sigmoid(out)
    return out

```

Resultaods de entrenarlo 100 Epocas:

```

Epoch 0: train loss 0.6835, test loss 0.6823
Epoch 10: train loss 0.3170, test loss 0.5469
Epoch 20: train loss 0.0530, test loss 1.1886
Epoch 30: train loss 0.0523, test loss 1.1545
Epoch 40: train loss 0.0357, test loss 1.6654
Epoch 50: train loss 0.0341, test loss 1.9416
Epoch 60: train loss 0.0337, test loss 2.0960
Epoch 70: train loss 0.0511, test loss 1.2306
Epoch 80: train loss 0.0343, test loss 2.0912
Epoch 90: train loss 0.0334, test loss 2.2868
Epoch 99: train loss 0.0332, test loss 2.4382

```

```

... Accuracy: 0.7209455022980958
Precision: 0.665680473372781
Recall: 0.6933744221879815

<function print(*args, sep=' ', end='\n', file=No

```