

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

### Laboratorio 5 : Análisis de Sentimientos

#### 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+|\'[\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

```

## Cálculos de Frecuencia

```
> 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
33	33	41
bomb	bombed	bombing
34	38	29
bridge%20collapse	buildings%20burning	buildings%20on%20fire
35	35	33
burned	burning	burning%20buildings
33	34	37
bush%20fires	casualties	casualty
25	35	34
catastrophe	catastrophic	chemical%20emergency
36	30	33
cliff%20fall	collapse	collapsed
36	34	35
collide	collided	collision
34	40	39
crash	crashed	crush
33	34	37



crushed	curfew	cyclone
31	37	32
damage	danger	dead
41	36	30
death	deaths	debris
36	38	37
deluge	deluged	demolish
42	27	34
demolished	demolition	derail
28	35	35
derailed	derailment	desolate
38	39	29
desolation	destroy	destroyed
36	37	32
destruction	detonate	detonation
34	36	32
devastated	devastation	disaster
31	36	35
displaced	drought	drown
36	35	32
drowned	drowning	dust%20storm
38	34	36
earthquake	electrocute	electrocuted
39	32	34
emergency	emergency%20plan	emergency%20services
37	35	33
engulfed	epicentre	evacuate
36	12	40
evacuated	evacuation	explode
36	36	38
exploded	explosion	eyewitness
33	39	32
famine	fatal	fatalities
39	38	45
fatality	fear	fire
37	40	38
fire%20truck	first%20responders	flames
33	29	39

flattened	flood	flooding
34	35	38
floods	forest%20fire	forest%20fires
36	19	32
hail	hailstorm	harm
35	32	41
hazard	hazardous	heat%20wave
34	35	34
hellfire	hijack	hijacker
39	33	35
hijacking	hostage	hostages
32	31	37
hurricane	injured	injuries
38	35	33
injury	inundated	inundation
38	35	10
landslide	lava	lightning
33	34	33
loud%20bang	mass%20murder	mass%20murderer
34	33	32
massacre	mayhem	meltdown
36	30	33
military	mudslide	natural%20disaster
34	37	34
nuclear%20disaster	nuclear%20reactor	obliterate
34	36	31
obliterated	obliteration	oil%20spill
31	29	38
outbreak	pandemonium	panic
40	37	37
panicking	police	quarantine
33	37	34
quarantined	radiation%20emergency	rainstorm
37	9	34
razed	refugees	rescue
35	36	22
rescued	rescuers	riot
35	35	34

rioting	35	ruin	37
sandstorm	37	screamed	36
screams	35	seismic	39
sinking	41	siren	29
smoke	34	snowstorm	35
stretcher	33	structural%20failure	35
suicide%20bomber	31	suicide%20bombing	39
survive	32	survived	30
terrorism	34	terrorist	11
thunder	38	thunderstorm	35
tragedy	36	trapped	31
traumatised	35	trouble	34
twister	40	typhoon	38
violent%20storm	33	volcano	24
weapon	39	weapons	39
wild%20fires	31	wildfire	40
wounded	37	wounds	37
wreckage	39	wrecked	39

Como se puede observar, el análisis realizado permitió encontrar que las palabras clave en los tweets tiene una relación directa con el peligro al estar relacionados con desastres naturales, situaciones de emergencia, equipos de rescate, elementos que se forman o se encuentran presentes en un accidente o desastre natural. También se pudo determinar que la frecuencia en la que todas estas palabras son utilizadas en la lista de tweets del data set es bastante grande. Esto se debe a que todas las palabras poseen una frecuencia que ronda entre los valores del 9 hasta el 42.

## **N-Grama**

```
> train_bigrama <-  
+   train %>%  
+   unnest_tokens(input = "keyword", output = "bigrama", token = "ngrams", n = 2)  
> View(train_bigrama)  
> train_bigrama %>%  
+   count(bigrama, sort = T)  
# A tibble: 38 x 2  
  bigrama                n  
  <chr>                <int>  
1 NA                    6448  
2 body 20bags           41  
3 oil 20spill            38  
4 burning 20buildings    37  
5 cliff 20fall           36  
6 dust 20storm           36  
7 nuclear 20reactor      36  
8 airplane 20accident    35  
9 bridge 20collapse      35  
10 buildings 20burning    35  
# i 28 more rows  
# i Use `print(n = ...)` to see more rows  
> |
```

Según el análisis realizado con el n-grama, se puede observar la frecuencia y la cantidad en que las palabras clave de los tweets están siendo utilizadas. Esto no solo permite tener una idea más clara sobre qué tanto se han utilizado estas palabras clave, sino también permite identificar sobre qué tipo de emergencia o desastre está hablando el tweet.

## **Palabras Positivas y Negativas**

```
> train_sentiment %>%  
+   mutate(polarity_level = ifelse(sentiment > 0, "Positive", "Negative")) %>%  
+   count(keyword, polarity_level)  
  keyword polarity_level  n  
1:   ablaze      Negative 46  
2:   ablaze      Positive 13  
3:  accident      Negative 48  
4:  accident      Positive  8  
5: aftershock      Negative 37  
---  
438: wreckage      Positive  7  
439: wrecked        Negative 48  
440: wrecked        Positive  3  
441:      <NA>      Negative 70  
442:      <NA>      Positive 18  
> |
```

## Determinación del Tweet en Base a las Palabras

```
train %>%
  get_sentences() %>%
  sentiment() -> train_sentiment
```

	target	element_id	sentence_id	word_count	sentiment
ason of this #earthquake May AL...	1	1	1	13	0.16641006
nge Sask.	1	2	1	6	-0.10206207
	1	2	2	1	0.00000000
'shelter in place' are being noti...	1	3	1	12	0.00000000
or shelter in place orders are exp...	1	3	2	10	0.03162278
e #wildfires evacuation orders in...	1	4	1	7	-0.18898224
to from Ruby #Alaska as smoke ...	1	5	1	16	-0.06250000
> California Hwy.	1	6	1	4	0.00000000
ections due to Lake County fire -...	1	6	2	11	-0.07537784
ry rain causes flash flooding of s...	1	7	1	14	-0.48107024
and I can see a fire in the woods...	1	8	1	15	0.19364917
/ evacuation happening now in t...	1	9	1	12	-0.33197640
nado is coming to our area...	1	10	1	10	-0.23717082

Según el análisis realizado, al utilizar el comando que se muestra para el set de datos que contiene los tweets se puede obtener el sentimiento (sentiment) que contiene ese mismo tweet. De esta forma, en base al resultado obtenido del sentimiento, se puede determinar si el tweet es positivo, negativo o neutral.

## Top 10 Tweets Más Negativos

target	element_id	sentence_id	word_count	sentiment
0	7473	2	12	-2.5980762
0	2020	1	7	-1.7008401
1	2377	1	15	-1.4794796
1	2392	1	15	-1.4794796
1	1541	1	21	-1.3965945
1	4548	1	15	-1.3168143
1	4557	1	15	-1.3168143
1	4562	1	15	-1.3168143
1	7301	1	15	-1.3039044
1	4235	2	3	-1.2990381

id	keyword	location	text
10689	wreck	NA	wreck wreck wreck wreck wreck wreck wreck v
2898	damage	Your Conversation	This real shit will damage a bitch
3415	derail	Nairobi-KENYA	24 killed in two simultaneous rail crash
3437	derail	Nairobi-KENYA	24 killed in two simultaneous rail crash
2225	chemical%20emergency	Las Vegas, Nevada	Bomb Crash Loot Riot Emergency Pipe B
6464	injured	Mumbai	Udhampur terror attack: Militants attack
6477	injured	NA	Udhampur terror attack: Militants attack
6484	injured	India	Udhampur terror attack: Militants attack
10446	wild%20fires	NA	God forbid this is true #California has e
6015	hazardous	NA	hazardous waste...noxious...

De acuerdo con el análisis, con la creación de la variable sentiment, no solo se puede determinar qué tan negativo es el tweet. Sino que también se puede identificar mediante el ID que fue asignado a cada uno de los tweets del set de datos. De este modo, brinda una forma más efectiva de poder saber e identificar cuales son los 10 tweets más negativos que se encuentran presentes en el dataset.

Como se puede contemplar en la columna "Target", la mayoría de los tweets que se encuentran entre los 10 más negativos del dataset poseen el número 1. Por lo cual, se puede concluir que estos tweets cumplen para poder ser clasificados como negativos al estar hablando de un desastre real.

### **Top 10 Tweets Más Positivos**

target	element_id	sentence_id	word_count	sentiment
1	5794	3	6	1.4288690
0	6911	3	5	1.3080998
0	6805	1	24	1.1483004
0	2284	3	2	1.1313708
0	535	3	4	1.1250000
0	1941	1	4	1.1250000
0	6296	4	4	1.1250000
0	6688	4	4	1.1250000
1	7286	2	9	1.0800000
0	7265	2	2	1.0606602

id	keyword	location	text
8266	rioting	NA	No justice=no peace Justice=peace.
9909	trouble	Kawartha Lakes, Ontario, Canad	but yes I would agree.
9749	tragedy	Ml©xico	I'm gunning down romance It never did
3277	demolish	Otsego, MI	#fitness #inspiration
777	avalanche	NA	Please Favorite & Share
2788	curfew	California	But no lies though.
8994	stretcher	NA	Please Favorite & Share
9581	thunder	NA	Please Favorite & Share
10426	whirlwind	Here, unless there.	The bomb saved millions of lives yes th
10399	whirlwind	140920-21 & 150718-19 BEIJING	Pretty smile.

Como se mencionó anteriormente, la creación de la variable sentiment, no solo ayuda a determinar la negatividad o positividad del tweet. Sino que también se puede identificar mediante el ID que fue asignado a cada uno de los tweets del set de datos. De este modo, brinda una forma más efectiva de poder saber e identificar cuales son los 10 tweets más positivos que se encuentran presentes en el dataset.

Según la columna “Target”, la mayoría de los tweets que se encuentran entre los 10 más positivos del dataset poseen el número 0. Esto significa que estos mismos tweets no están hablando de un desastre real. Por lo cual, se puede deducir que estos tweets si cumplen para poder ser clasificados como tweets positivos.

### Cree una variable que contenga la “negatividad” de cada tweet

```

from nltk.sentiment import SentimentIntensityAnalyzer

# Initialize the Sentiment Intensity Analyzer
sia = SentimentIntensityAnalyzer()

# Calculate the negativity scores for the tweets in the dataset
df['negativity_score'] = df['preprocessed_text'].apply(lambda x: sia.polarity_scores(str(x))['neg'])

# Show the first few rows of the dataframe with the added negativity score
df.head()

```

	id	keyword	location	text	target	preprocessed_text	negativity_score
0	1	NaN	NaN	Our Deeds are the Reason of this #earthquake M...	1	deeds reason may allah forgive us	0.000
1	4	NaN	NaN	Forest fire near La Ronge Sask. Canada	1	forest fire near la ronge sask canada	0.286
2	5	NaN	NaN	All residents asked to 'shelter in place' are ...	1	notified officers evacuation shelter place ord...	0.000
3	6	NaN	NaN	13,000 people receive #wildfires evacuation or...	1	people receive evacuation orders california	0.000
4	7	NaN	NaN	Just got sent this photo from Ruby #Alaska as ...	1	got sent photo ruby smoke pours school	0.000

Se utilizó la biblioteca NLTK para inicializar un SentimentIntensityAnalyzer. Esta herramienta se utiliza para calcular la puntuación de negatividad para cada tweet preprocesado en el DataFrame df. Se añade una nueva columna llamada negativity\_score al DataFrame que contiene estas puntuaciones de negatividad. Finalmente, muestra las primeras filas del DataFrame actualizado para comprobar los cambios.

### ¿La inclusión de esta variable mejoró los resultados del modelo de clasificación?

Modelo Antiguo:

```
11] ✓ 0.0s
.. Accuracy: 0.7288246881155613
   Precision: 0.6849529780564263
   Recall: 0.6733436055469953
```

Nuevo modelo con negativity:

```
]
Accuracy: 0.7130663164806303
Precision: 0.6615853658536586
Recall: 0.6687211093990755
F1 Score: 0.6651340996168582
```

Ambos modelos son clasificadores de texto contruidos usando PyTorch, y están diseñados específicamente para analizar tweets. El primero, `TweetClassifier`, es un modelo más simple que usa una capa de embedding, seguida de una capa LSTM y una capa completamente conectada. Este modelo alcanzó una precisión del 72.88%, una precisión del 68.49% y un recall del 67.33%.

El segundo modelo, `TweetClassifierWithNegativity`, añade una capa adicional de complejidad al incluir la puntuación de negatividad como una característica adicional. A pesar de esta característica adicional, el rendimiento del modelo es ligeramente inferior al del primer modelo en términos de precisión y precisión, aunque comparable en términos de recall. Este modelo obtiene una precisión del 71.31%, una precisión del 66.16% y un recall del 66.87%. Además, este segundo modelo ofrece un valor de F1 Score de 0.6651, un indicador más completo del balance entre la precisión y el recall.

Dado que el modelo más simple supera ligeramente al modelo más complejo en la mayoría de las métricas, esto podría indicar que la adición de la puntuación de negatividad no mejora significativamente el rendimiento del modelo. A menudo, agregar más características no garantiza una mejor precisión y puede incluso dar lugar a un sobreajuste si no se maneja cuidadosamente. Es posible que el rendimiento pueda mejorarse más con una ingeniería de características más detallada, ajuste de hiperparámetros, o incluso mediante la adición de más datos al conjunto de entrenamiento.



## **Conclusiones:**

Basándonos en los resultados de ambos modelos y la observación de que en Twitter la gente tiende a quejarse y hablar de temas negativos, podemos concluir que la incorporación de una puntuación de negatividad como característica adicional no mejoró significativamente el rendimiento del modelo. Esto podría deberse a que la negatividad es una característica omnipresente en los tweets y, por lo tanto, no contribuye a la diferenciación efectiva entre distintas clases o temas de tweets. En otras palabras, la puntuación de negatividad puede ser menos informativa en un entorno donde la mayoría de los mensajes ya tienen un tono negativo.

Es posible que en una plataforma como Twitter, donde las quejas y los comentarios negativos son comunes, el valor agregado de considerar la "negatividad" como una característica es marginal en el mejor de los casos. Esto se ve reflejado en el hecho de que el modelo más simple, que no toma en cuenta la puntuación de negatividad, superó ligeramente al modelo más complejo en términos de precisión y precisión.

Por lo tanto, en contextos donde la negatividad es la norma más que la excepción, podría ser más útil centrarse en otras características o métodos para mejorar el rendimiento del modelo de clasificación. La incorporación de una puntuación de negatividad en tales casos podría no ser una estrategia eficaz para mejorar la capacidad del modelo para hacer predicciones más precisas.