Universidad del Valle de Guatemala
Data Science 1 - Sección 10
Christopher Kevin Sandoval García 13660
María Fernanda Estrada Cornejo 14198
*
*
20 de agosto de 2020

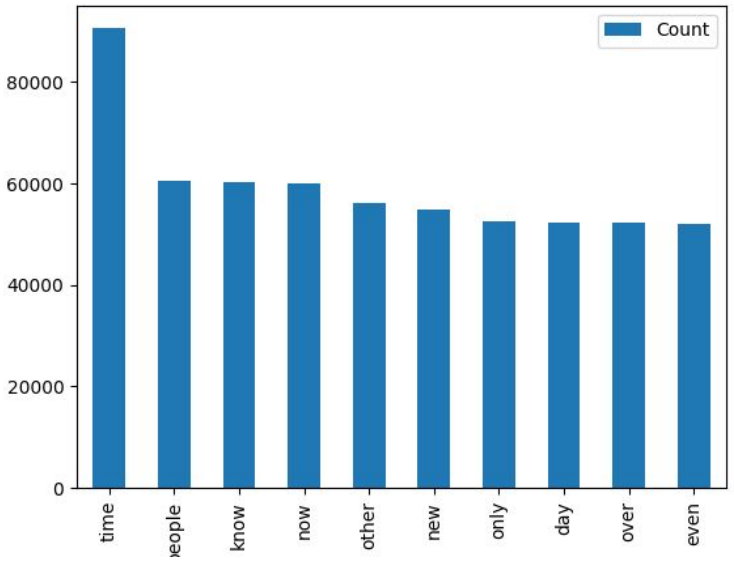
Laboratorio 4

Análisis exploratorio

Blogs

Frecuencia, qué palabra se repite más e histograma

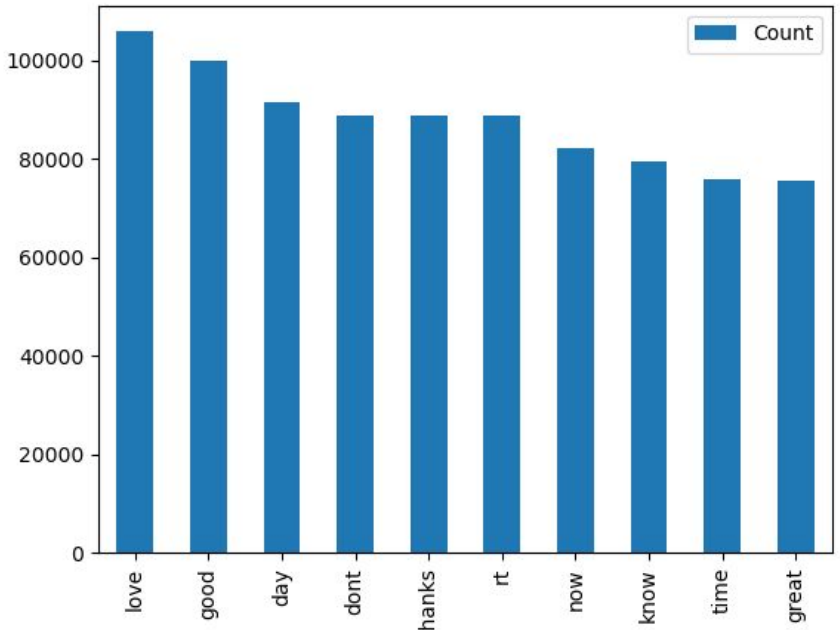
	Word	Count
0	time	90595
1	people	60474
2	know	60212
3	now	59898
4	other	56148
5	new	54782
6	only	52587
7	day	52316
8	over	52299
9	even	51999



Tweets

Frecuencia, qué palabra se repite más e histograma

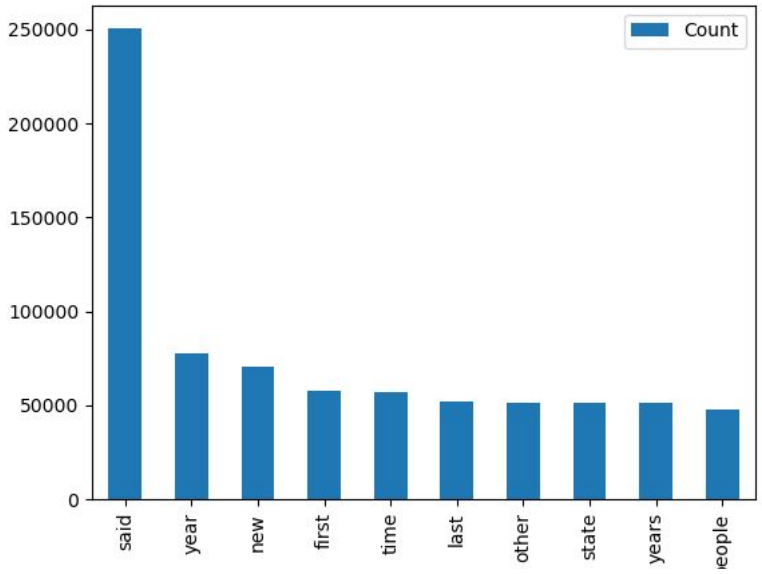
	Word	Count
0	love	105930
1	good	100024
2	day	91694
3	dont	88918
4	thanks	88813
5	rt	88796
6	now	82141
7	know	79419
8	time	75995
9	great	75585



Noticias

Frecuencia, qué palabra se repite más e histograma

	Word	Count
0	said	250379
1	year	77367
2	new	70791
3	first	57873
4	time	57059
5	last	52091
6	other	51664
7	state	51391
8	years	51052
9	people	47790



Limpieza y preprocesamiento de datos

Para la lectura de los datos, el encoding fue UTF-8. Los siguientes pasos se aplicaron para la limpieza de datos:

1. Convertir todo el texto a minúsculas
2. Eliminar caracteres especiales, como #, @, comillas o apóstrofes (de cualquier tipo), etc.
3. Eliminar urls
4. Eliminar emoticones si existen
5. Eliminar signos de puntuación, como !, comas, puntos, corchetes, etc.
6. Eliminar stopwords, como artículos, preposiciones y conjunciones
7. Eliminar números no significativos

Todos estos pasos quedaron documentados en el código. Como se utilizó Python, no se necesitaron módulos o paquetes extra, aparte de Pandas.

Generación de los n-gramas y cálculo de sus frecuencias y probabilidades

Primero, se calcularon los n-gramas solicitados, 3-grama. Para ello, se tokenizó (o separó por palabra) cada oración del archivo ya limpio para poder aplicarle la función de n-grama de la librería nltk. A esta función solamente se le pasaba el valor n y la lista del texto en tokens. Luego de esto, se determinó la frecuencia y probabilidad de cada n-grama con este mismo paquete. En la lista de n-gramas se buscan las palabras ingresadas y si se encuentra en uno o más n-gramas, se determina la probabilidad al encontrar la cantidad de n-gramas a las que pertenece esta frase.

A continuación un ejemplo de cómo se mira un 3-grama:

```
('years', 'thereafter', 'most')
('thereafter', 'most', 'oil')
('most', 'oil', 'fields')
('oil', 'fields', 'platforms')
('fields', 'platforms', 'named')
('platforms', 'named', 'after')
```

Algoritmo

Primero, se obtienen los n-gramas del texto completo. Luego, se busca la palabra o frase ingresada en estos n-gramas y se determinan todas las probabilidades que tienen. Se toman las probabilidades más grandes (top 5) y se muestran las posibles palabras que puede escribir el usuario (y con qué probabilidad).

Función de predicción

Para una palabra

```
Inserte las palabras: true
{'palabra': 'love', 'probabilidad': 0.0323943661971831}
{'palabra': 'story', 'probabilidad': 0.021126760563380 28}
{'palabra': 'self', 'probabilidad': 0.016901408450704224}
{'palabra': 'nature', 'probabilidad': 0.015492957746478873}
{'palabra': 'god', 'probabilidad': 0.011267605633802818}
```

Para dos palabras

```
Inserte las palabras: most people
{'palabra': 'know', 'probabilidad': 0.05673758865248227}
{'palabra': 'think', 'probabilidad': 0.04964539007092199}
{'palabra': 'reading', 'probabilidad': 0.02127659574468085}
{'palabra': 'dont', 'probabilidad': 0.02127659574468085}
{'palabra': 'write', 'probabilidad': 0.02127659574468085}
```