

RESUMEN DE PQR

LIC.ANDRÉS PAZ

Cuando estamos trabajando en atención al cliente una de las principales virtudes de los usuarios es poder manifestarse sobre el servicio o productos que brinda la empresa, razón por la cual para el área de talento humano es importante gestionar y agilizar la solución a las peticiones, reclamos, solicitudes de los clientes, razón por la cual miraremos cómo realizar un resumen de textos mediante python basada en aprendizaje profundo. Para ello usaremos la librería [nltk](#) de python. A continuación vamos mirar como funciona la librería

- Lo primero que haremos será convertir los párrafos en oraciones
- Eliminar todos los caracteres especiales, detener palabras y números de todas las oraciones. Después del preprocesamiento, obtenemos las siguientes oraciones
- El siguiente paso consiste en la segmentación del texto en frases o palabras a esto se denomina tokenizar las oraciones
- Luego encontramos la frecuencia ponderada de apariciones de todas las palabras, podemos encontrar la frecuencia ponderada de cada palabra dividiendo su frecuencia por la frecuencia de la palabra más frecuente
- Reemplazar las frecuencias ponderadas de cada palabra en las frases
 - Así que sigue trabajando $1 + 0.20 = 1.20$
 - Sigue esforzándote $1 + 0.20 = 1.20$
 - Nunca te rindas $0.20 + 0.20 = 0.40$
- Ordenar oraciones en orden descendente de suma de tal forma que la frase que tiene mayor puntuación sigue creciendo e iterando

Ahora pasemos esto al algoritmo de python:

1. Descargar e Instalar [python](#)
2. Buscar en el menú inicio cmd y ejecutar como administrador y posterior a ello escribir "cd C:\Users\handres\AppData\Local\Programs\Python\Python39" donde handres es el nombre de usuario de su computadora. en caso de no existir la ruta verificar la ruta en power bi archivo->opciones y python y en la consola de comandos ejecutar los siguientes comandos (**Nota:** si pip install librería no funciona probar usando python -m pip install librería)
 - a. python -m install --user -U nltk #Librería de PNL
 - b. python -m install --user -U numpy
 - c. python -m nltk.downloader popular #instalamos los paquetes populares
3. Para conectarnos a un set de datos lo que debemos hacer es de datos abiertos El enlace generado lo copiamos (<https://www.datos.gov.co/resource/t7af-ejy7.csv>)
4. Pegamos el script :

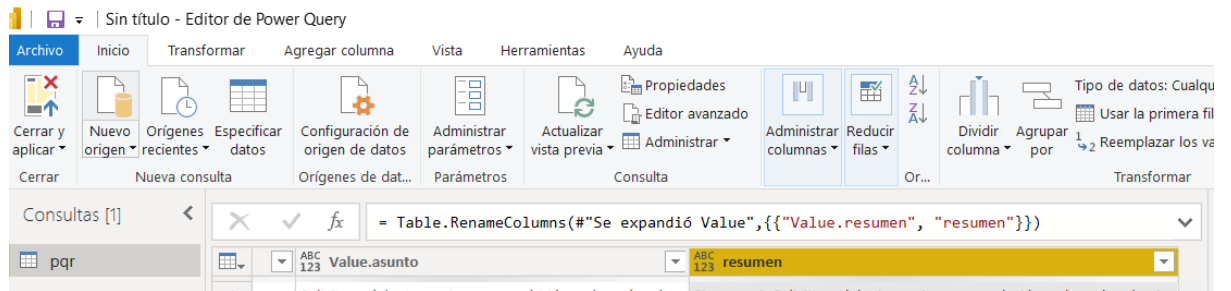
RESUMEN DE PQR

LIC.ANDRÉS PAZ

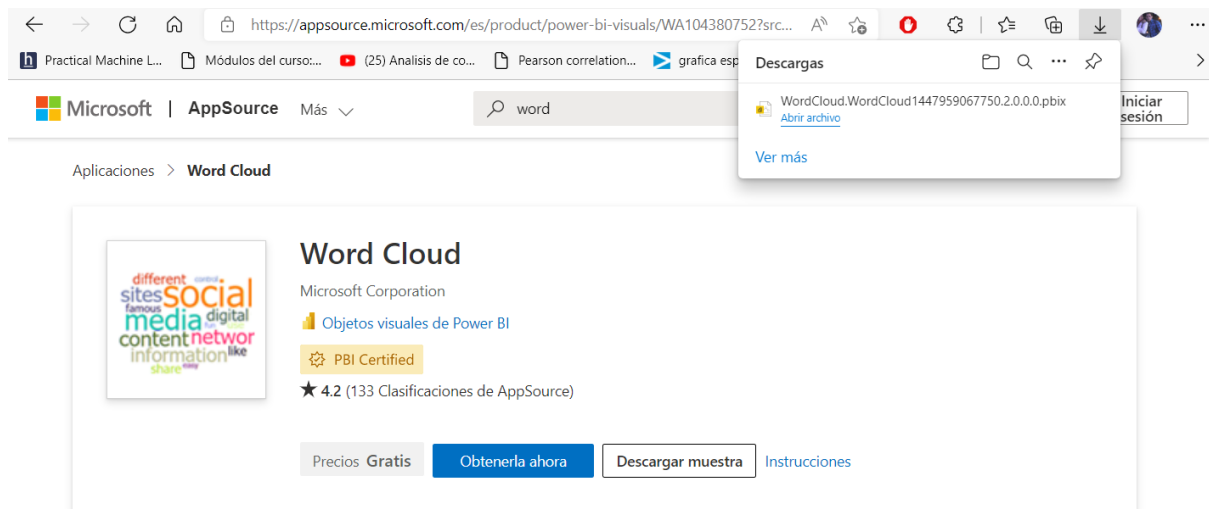
```
nombre_columna_analizar="asunto"
nombre_nueva_columna="resumen"
cantidad_palabras_resumen=30 #calculamos la puntuación solo para oraciones con menos
de n palabras porque no queremos oraciones tan largas
cantidad_oraciones_importantes=7 # n oraciones más importantes con el puntaje más alto
para mostrarse en pantalla
#####
import re
import nltk
import pandas as pd
#nltk.download('popular')
coeficiente_sentimiento=list()
for i in dataset[nombre_columna_analizar].index:
    article_text=dataset[nombre_columna_analizar][i]
    formatted_article_text = re.sub('[^a-zA-Z]', ' ', article_text )
    formatted_article_text = re.sub(r's+', ' ', formatted_article_text)
    sentence_list = nltk.sent_tokenize(article_text) #obtenemos las palabras de las oraciones
    stopwords = nltk.corpus.stopwords.words('spanish')
    word_frequencies = {}
    for word in nltk.word_tokenize(formatted_article_text):
        if word not in stopwords:
            if word not in word_frequencies.keys():
                word_frequencies[word] = 1
            else:
                word_frequencies[word] += 1
    maximum_frequency = max(word_frequencies.values())
    for word in word_frequencies.keys():
        word_frequencies[word] = (word_frequencies[word]/maximum_frequency)
    sentence_scores = {}
    for sent in sentence_list:
        for word in nltk.word_tokenize(sent.lower()):
            if word in word_frequencies.keys():
                if len(sent.split(' ')) < cantidad_palabras_resumen:
                    if sent not in sentence_scores.keys():
                        sentence_scores[sent] = word_frequencies[word]
                    else:
                        sentence_scores[sent] += word_frequencies[word]
    import heapq
    summary_sentences = heapq.nlargest(cantidad_oraciones_importantes,
sentence_scores, key=sentence_scores.get)
    summary = ' '.join(summary_sentences)
    if summary=="": #si no hay resumen
        summary='Sin resumir:' + article_text #Muestra el texto Sin resumir con el texto
completo del articulo
    coeficiente_sentimiento.append(summary)
dataset[nombre_nueva_columna] =
pd.DataFrame(coeficiente_sentimiento,columns=[nombre_nueva_columna],dtype ='string')
print(dataset)
```

RESUMEN DE PQR LIC.ANDRÉS PAZ

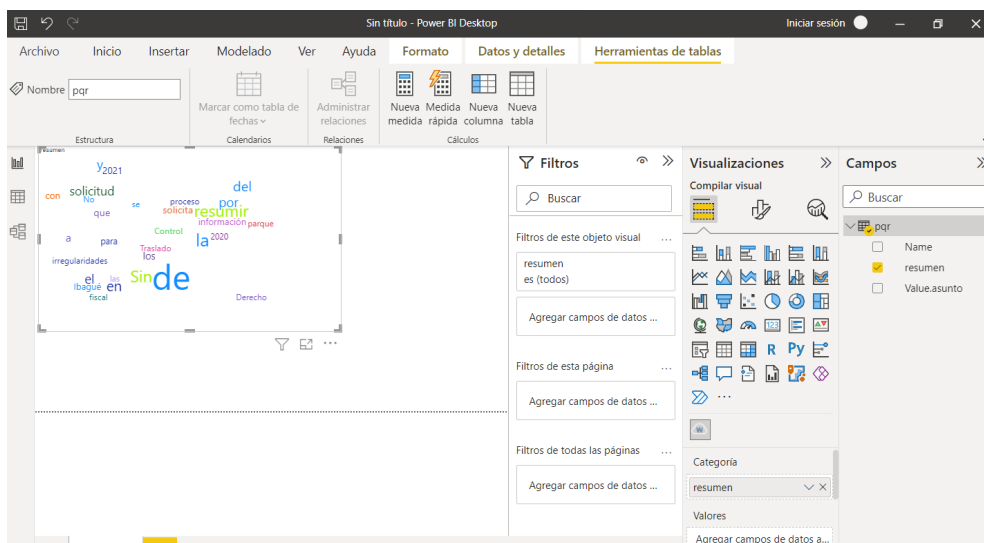
Modificamos el nombre de la consulta a pqr y el nombre de la columna value.resumen a resumen cerramos y aplicamos.



Descargamos la visualización de [nube de palabras](#)



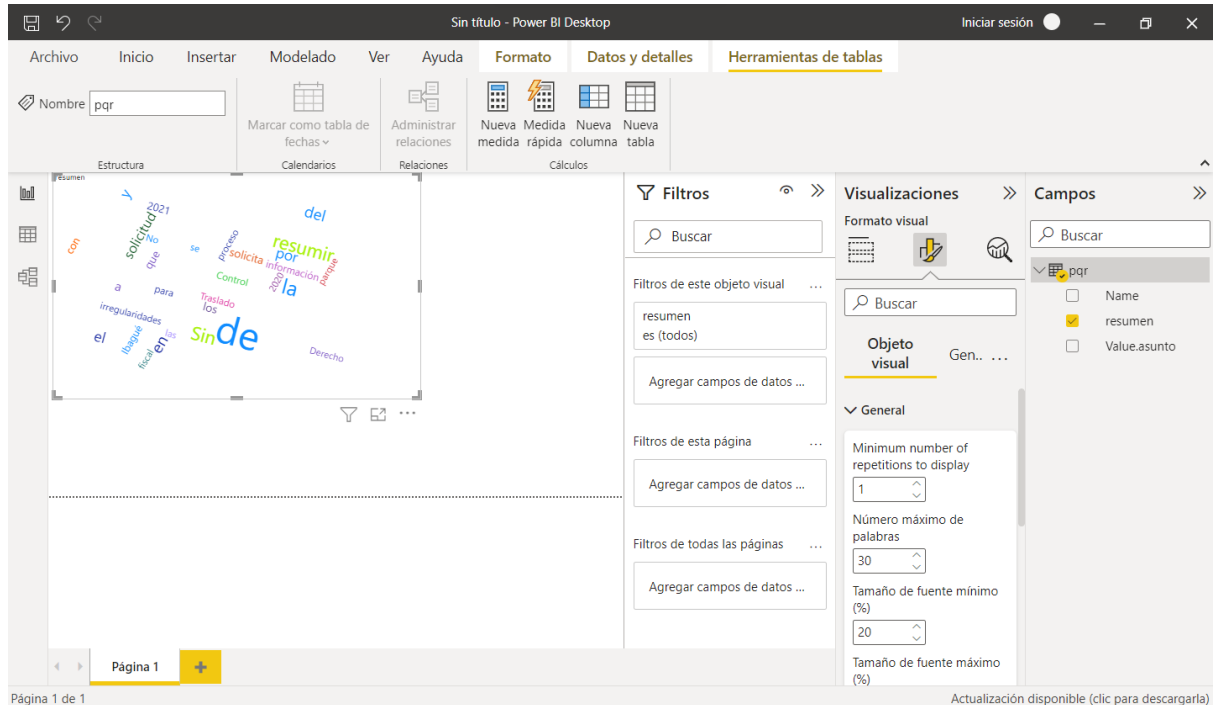
Ahora usaremos la visualización de nube de palabras para observar cuales son las palabras más frecuentes solicitadas por los ciudadanos



RESUMEN DE PQR

LIC.ANDRÉS PAZ

En las opciones de configuración de visualización tenemos la posibilidad de verificar cuantas palabras deseamos observar y muchas configuraciones más



Por ejemplo una de las palabras que podemos identificar que presenta relevancia es **IRREGULARIDADES** junto con el lugar denominado con la palabra **PARQUE** que nos permiten enfocar sobre que está sucediendo sobre la frecuencia en irregularidades y el parque de este lugar.

RESUMEN DE PQR
LIC.ANDRÉS PAZ

Referencias:

<https://pharos.sh/resumir-textos-con-nltk-en-python/>