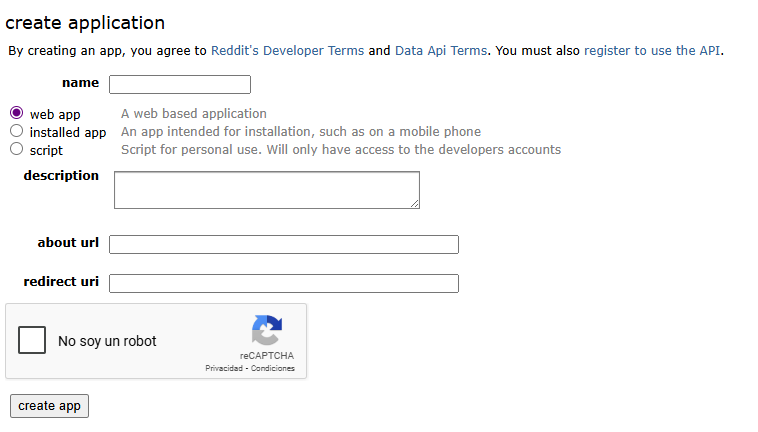
***Proyecto final analsis de datos Discusiones en Reddit sobre Donald Trump***

1. Para esto, haremos uso del API de Reddit, tenemos que tener una cuenta creada, luego seguiremos este enlace <https://www.reddit.com/prefs/apps>
2. Aqui adentro, crearemos una app de desarrollador, presionando el botón de Create a app, al presionarlo, nos saldrá esto



Le daremos un nombre, marcaremos script y en redirect uri pondremos esto: <http://localhost:8080>

Una vez creada, aparecerá un panel como este



Donde la parte mas importante es el client id (esta debajo del personal use script) y Client secret, esto nos servirá para conectarnos a la API de Reddit, también es importante obtener lo siguiente: <plataforma>:<nombre\_app>:<versión> (by /u/<tu\_usuario\_de\_reddit>)

Que en síntesis se vería algo asi: PC:analisis\_sentimiento\_py:v1.0 (by /u/United-Prompt-8786)

Así como también la contraseña, que es la que usamos para nuestro usuario de Reddit

Un vez tenemos esto, podemos pasar a jupyter para realizar la búsqueda y limpieza respectiva

1. Usaremos un código similar a este para los 250.000 datos, esta búsqueda tardará mucho dependiendo el dispositivo

import praw

import pandas as pd

import os

from getpass import getpass

# --- 1. Configura tus Credenciales ---

CLIENT\_ID = "7ZAfjRQrmwzZOYesFaVW-A"

CLIENT\_SECRET = "-sDcb8fx3u1zTl0g4wvrbRpma6Wxjg"

USER\_AGENT = "PC:analisis\_sentimiento\_py:v1.0 (by /u/United-Prompt-8786)"

REDDIT\_USERNAME = "United-Prompt-8786"

REDDIT\_PASSWORD = getpass('Ingresa tu contraseña de Reddit: ')

# --- 2. Conexión a la API de Reddit ---

try:

reddit = praw.Reddit(

client\_id=CLIENT\_ID,

client\_secret=CLIENT\_SECRET,

user\_agent=USER\_AGENT,

username=REDDIT\_USERNAME,

password=REDDIT\_PASSWORD,

)

print(f"Conexión exitosa como /u/{reddit.user.me()}")

except Exception as e:

print(f"Error de conexión: {e}")

raise

# --- 3. Extracción de Datos ---

all\_comments\_data = []

target\_count = 250000

# ----- CAMBIO 1: AÑADIMOS MÁS SUBREDDITS -----

subreddits\_to\_search = ['politics', 'worldnews', 'news', 'PoliticalDiscussion', 'Conservative', 'democrats']

keywords = 'trump OR election OR biden OR vote'

print(f"Iniciando búsqueda de {target\_count} comentarios...")

for subreddit\_name in subreddits\_to\_search:

if len(all\_comments\_data) >= target\_count:

break

subreddit = reddit.subreddit(subreddit\_name)

print(f"\nBuscando en r/{subreddit\_name}...")

# ----- CAMBIO 2: AUMENTAMOS EL LÍMITE DE POSTS A BUSCAR -----

for submission in subreddit.search(keywords, limit=300): # Límite aumentado a 300

if len(all\_comments\_data) >= target\_count:

break

print(f" -> Procesando post: '{submission.title[:50]}...'")

submission.comments.replace\_more(limit=0)

for comment in submission.comments.list():

if len(all\_comments\_data) >= target\_count:

break

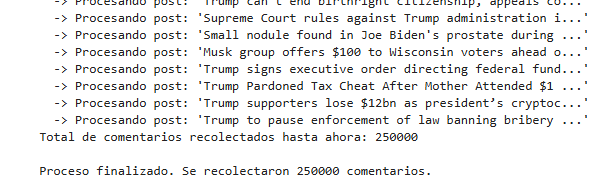
if comment.body and comment.body != '[deleted]' and comment.body != '[removed]':

all\_comments\_data.append({'texto': comment.body})

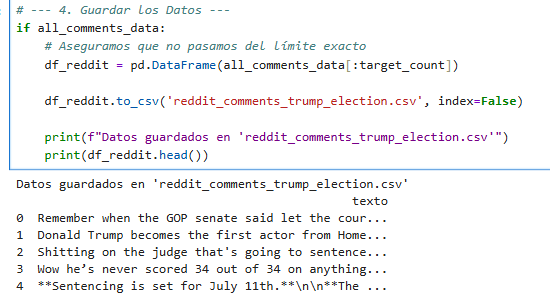
print(f"Total de comentarios recolectados hasta ahora: {len(all\_comments\_data)}")

print(f"\nProceso finalizado. Se recolectaron {len(all\_comments\_data)} comentarios.")

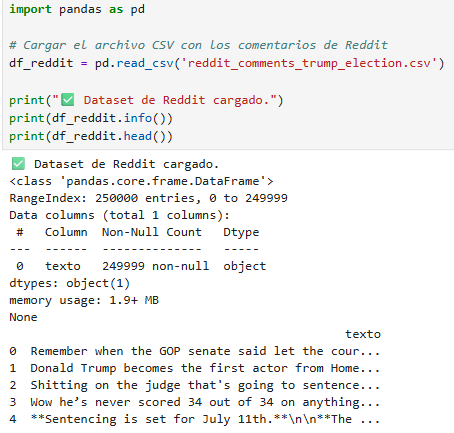
Obtendremos algo como esto:



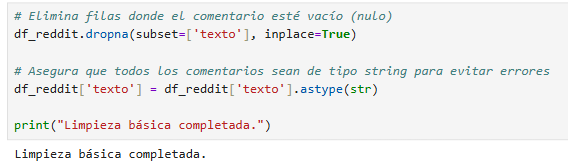
1. Procedemos a guardarlos



1. Lo cargamos como dataframe en pandas, para realizar la limpieza



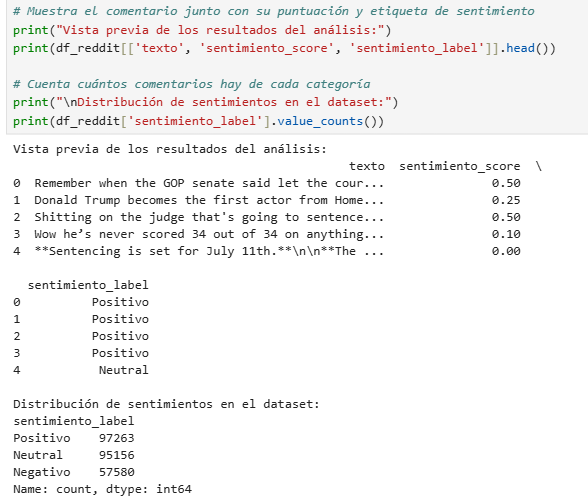
1. Limpieza básica



1. Realizamos el análisis de sentimiento con textblob, esto tardará



1. Echamos un vistazo a lo que hicimos



1. Con el siguiente código, lo enviaremos a Mongo Atlas

import pymongo

import pandas as pd

# --- 1. Configuración de la Conexión ---

MONGO\_URI = "mongodb+srv://juan:12345@cluster0.agcdm8h.mongodb.net/"

dataframe\_a\_subir = df\_reddit

try:

# --- 2. Conectar al Cliente de MongoDB ---

client = pymongo.MongoClient(MONGO\_URI)

client.admin.command('ping')

print("¡Conexión a MongoDB Atlas exitosa!")

# --- 3. Seleccionar Base de Datos y una Nueva Colección ---

db = client['proyecto\_futbol']

collection = db['sentimiento\_reddit'] # Colección final para los datos analizados

# --- 4. Preparar y Cargar los Datos ---

datos\_para\_mongo = dataframe\_a\_subir.to\_dict(orient='records')

print(f"Limpiando la colección '{collection.name}'...")

collection.delete\_many({})

# Usamos el método de lotes que ya sabemos que funciona para grandes volúmenes

batch\_size = 10000

total\_docs = len(datos\_para\_mongo)

print(f"Iniciando inserción de {total\_docs} documentos en la colección '{collection.name}'...")

for i in range(0, total\_docs, batch\_size):

batch = datos\_para\_mongo[i:i + batch\_size]

collection.insert\_many(batch)

print(f" -> Lote {i//batch\_size + 1} insertado ({i + len(batch)}/{total\_docs} documentos)")

print("¡Carga del análisis de sentimientos a MongoDB completada!")

except Exception as e:

print(f" Error durante el proceso: {e}")

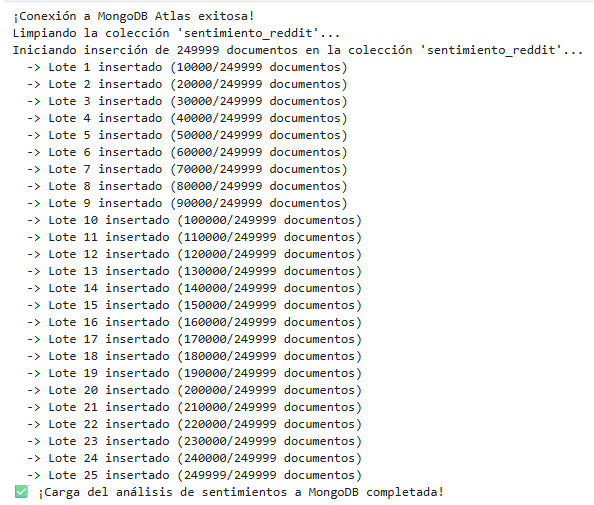
finally:

# Cerramos la conexión

if 'client' in locals():

client.close()

Obtendremos como resultado algo similar a esto



El paso de los datos se realizó mediante lotes, para no perder conectividad con Mongo Atlas, ya que, si lo realizamos todo a la vez, este bloqueará la solicitud