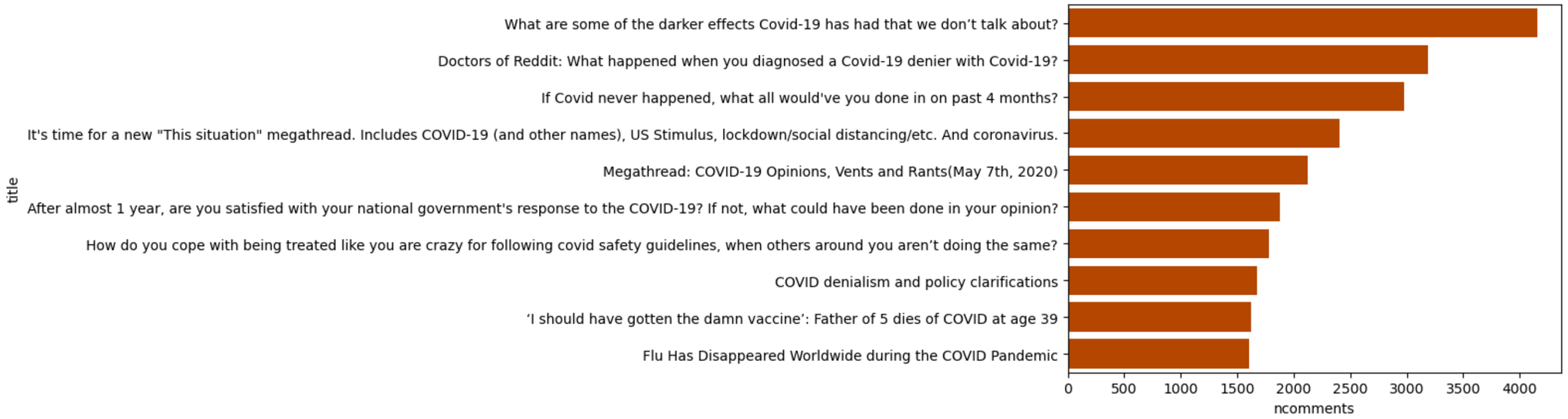
Análisis de la Percepción Pública sobre la Pandemia

Un estudio basado en datos de reddit con técnicas de Big Data

Contenido

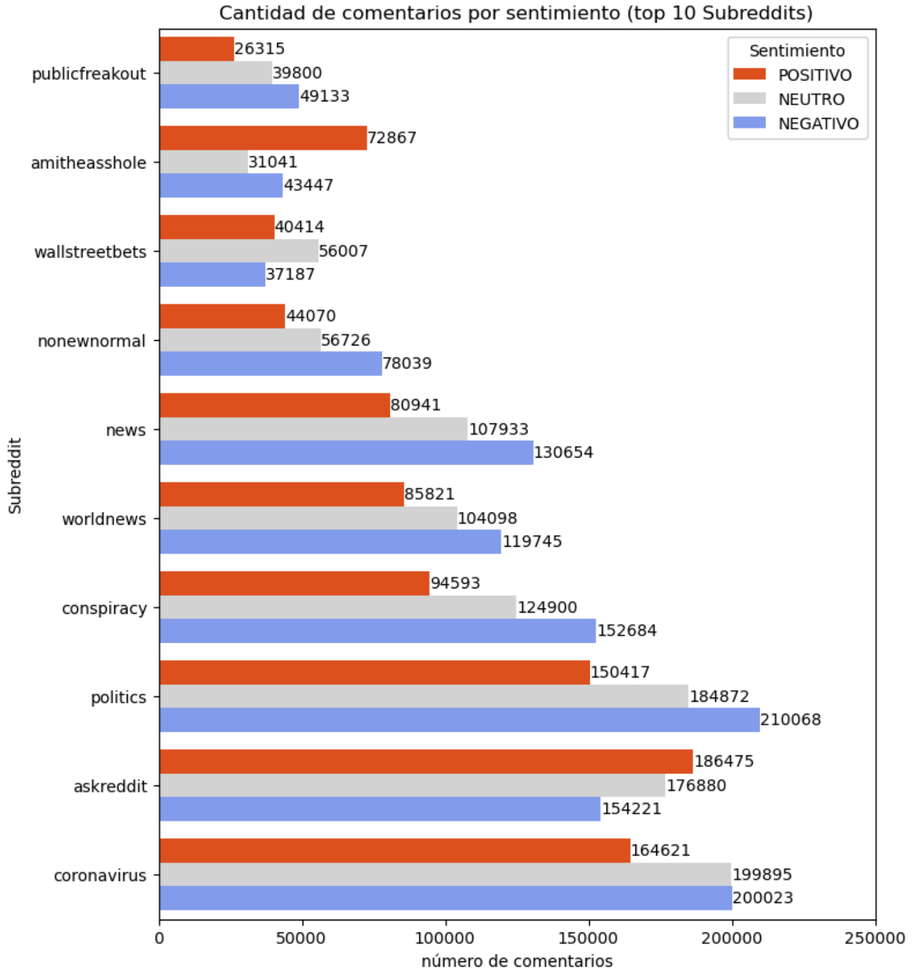
01 Análisis Exploratorio

Top 10 Posts con más Comentarios



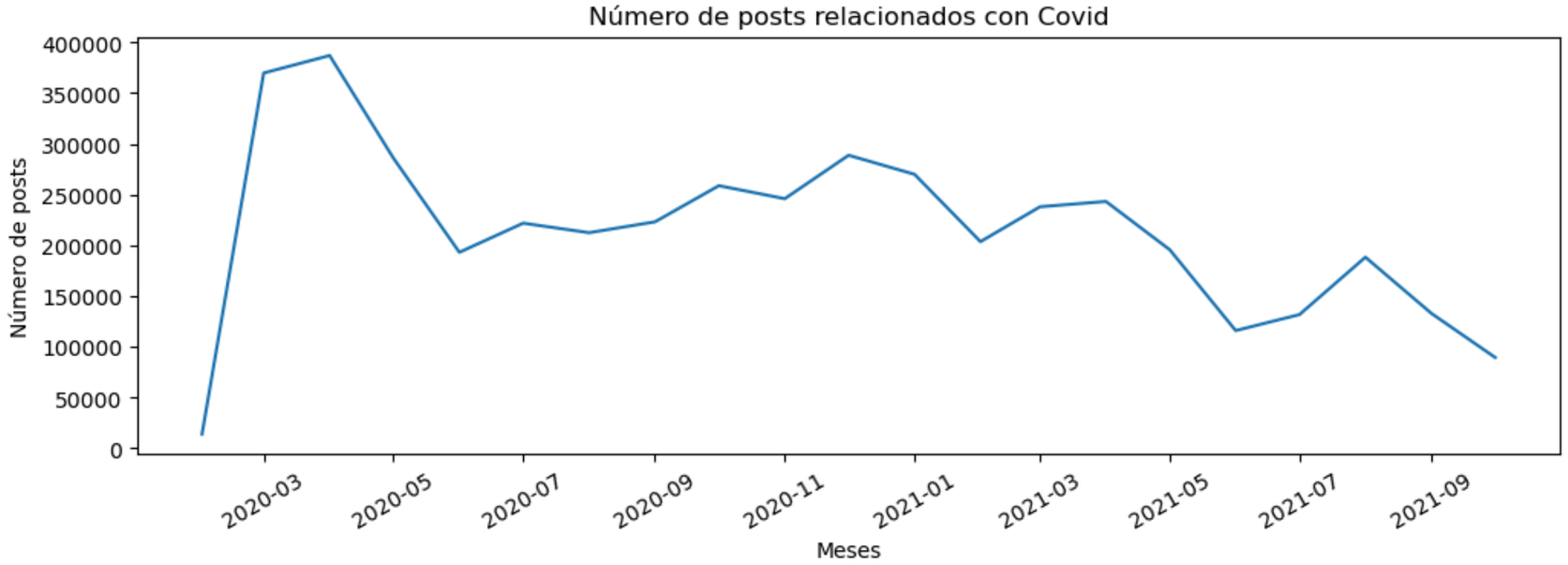
- Se realizó un agrupamiento por post_id para contar el número de comentarios de cada post.
- Se ordenaron los posts por el # de comentarios, seleccionando los 10 con más comentarios.

Comentarios por sentimiento



- En 8/10 comentarios predomina el sentimiento negativo.
- Categorización:
  - > 0.4 = POSITIVO
  - -0.4 < X < 0.4 = NEUTRO
  - < -.04 = NEGATIVO

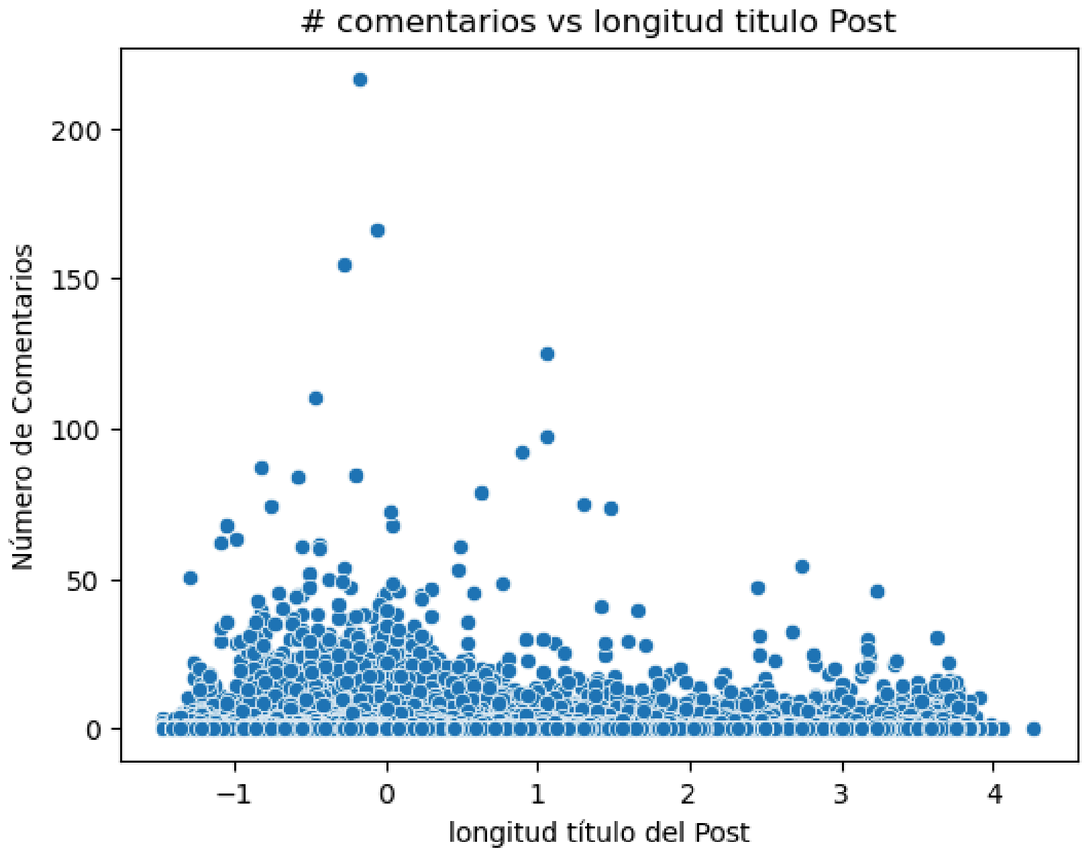
Distribución de Posts por Meses del Año



Se realizó un análisis sobre la cantidad de comentarios relacionados con COVID-19 en Reddit, agrupándolos por meses del año en que fueron publicados.

Relación entre Longitud del Post y el # de Comentarios

Este análisis mostró una relación directa entre la extensión de los títulos y la cantidad de comentarios generados, lo que sugiere que los títulos más largos podrían captar más la atención de los usuarios o generar mayor discusión.



02 Motivación y Desafíos

Motivación

La motivación de este proyecto fue analizar las interacciones generadas en Reddit durante la pandemia de COVID-19, donde los usuarios discutieron temas clave sobre el impacto global. Utilizamos técnicas como TF-IDF y análisis de sentimientos, junto con visualizaciones de grafos y scatterplots, para identificar subreddits relevantes y analizar la relación entre los sentimientos en los títulos de los posts y los comentarios. El objetivo principal fue obtener insights sobre la **percepción pública de la pandemia** y cómo los usuarios interactuaban, contribuyendo al análisis social de este evento histórico.

Desafíos

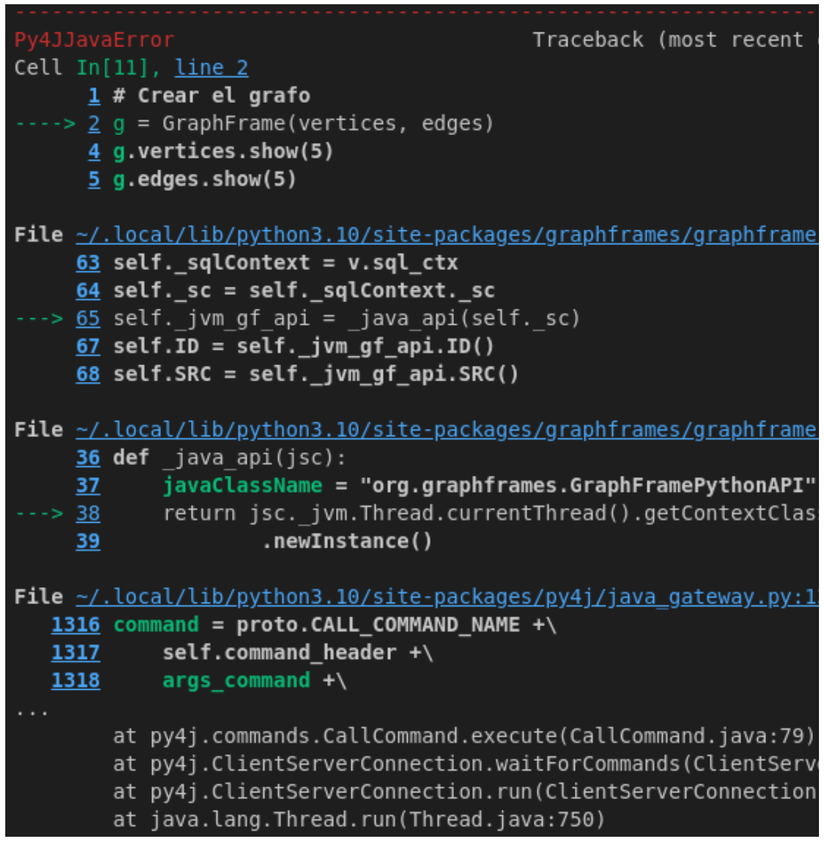
Manejo de grandes volúmenes de datos:
Utilizar herramientas como DuckDB y Spark fue esencial, pero optimizar las consultas y manejar los datos a gran escala sin saturar los recursos fue un desafío constante.

Aplicación de técnicas avanzadas como TF-IDF:
Implementar TF-IDF fue un desafío a la hora de encontrar un equilibrio entre la cantidad de términos utilizados y la relevancia que aportaban, evitando ruido o términos comunes que podían distorsionar el análisis.

Creación de grafo para obtener los dominios con más conexiones:
Uno de los principales desafíos fue la necesidad de seleccionar estratégicamente una porción del dataframe de comentarios que no contaba con muchos registros, para garantizar que la computadora pudiera generar el grafo sin sobrecargar el kernel.

Problemas encontrados

Al generar un grafo con GraphFrame, surgió un **Py4JJavaError** que impedía cargar la clase GraphFramePythonAPI debido a una incompatibilidad entre Java y PySpark.
Fue necesario utilizar NetworkX como alternativa para generar el grafo, evitando así la incompatibilidad que impedía la ejecución correcta del grafo.



03 Análisis de Costos

Almacenamiento

Dataset de posts (**2 GB**) y Dataset de comentarios (**13 GB**).

Servicios a utilizar:

Amazon S3

Costo estimado: $0.023 por GB al mes.

Costo mensual para 15 GB: 15 GB * $0.023 = $0.345/mes.

Procesamiento de datos con Databricks

Para el análisis con KMeans (creación de 3 clusters) y la generación de wordclouds

- Tiempo de procesamiento de datos y generación de clusters = 1 hora
- Costo Databricks Classic Jobs: $0.15 por nodo de trabajo.
- Para procesar 15GB de datos, es razonable utilizar 2 nodos.
- Costo de procesamiento: 2 nodos * $0.15/hora * 1 hora = $0.30 por procesamiento.

Generación de WordClouds

- Este proceso se completará en 2 minutos.
- Se utilizarán los mismos nodos de procesamiento (2 nodos)
- Costo para la generación de WordClouds: 2 nodos * $0.15/hora * 0.03 horas = $0.009.

04 Resultados

Análisis del Grafo: Top 20 Dominios con Más Conexiones

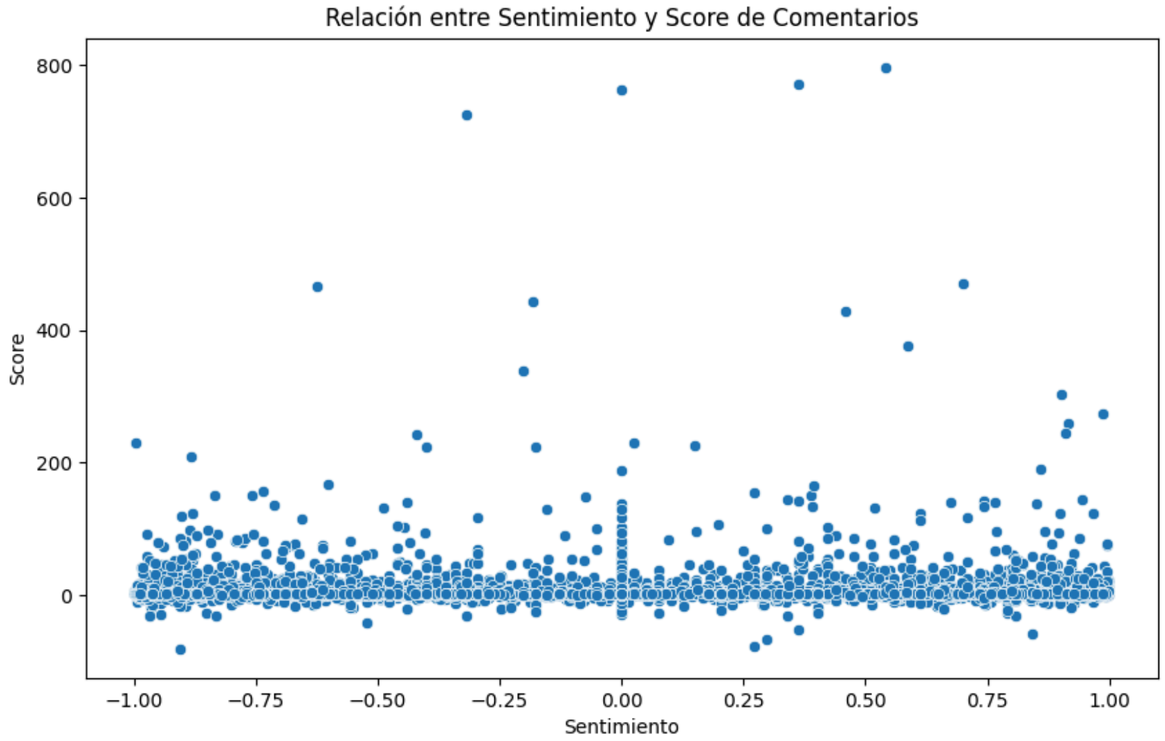
Se filtraron los datos para identificar las conexiones entre los usuarios y los dominios de las publicaciones.

Se etiquetaron únicamente los nodos que representan dominios (que contienen un punto)

20 dominios con más conexiones

Relación entre Sentimiento y Score de Comentarios

- No existe una correlación clara entre sentimiento y score.
- Mayor concentración alrededor del sentimiento neutral.
- Score alto en comentarios neutrales y positivos.
- Sentimientos negativos no se destacan por tener scores altos



Análisis de texto (Pipeline)

Se creó un pipeline para automatizar el flujo de procesamiento de los datos textuales:

Tokenización: Separación del texto en palabras individuales (tokens).

Eliminación de Stopwords: Se filtraron palabras comunes que no aportan valor al análisis (ej. "el", "de", etc.).

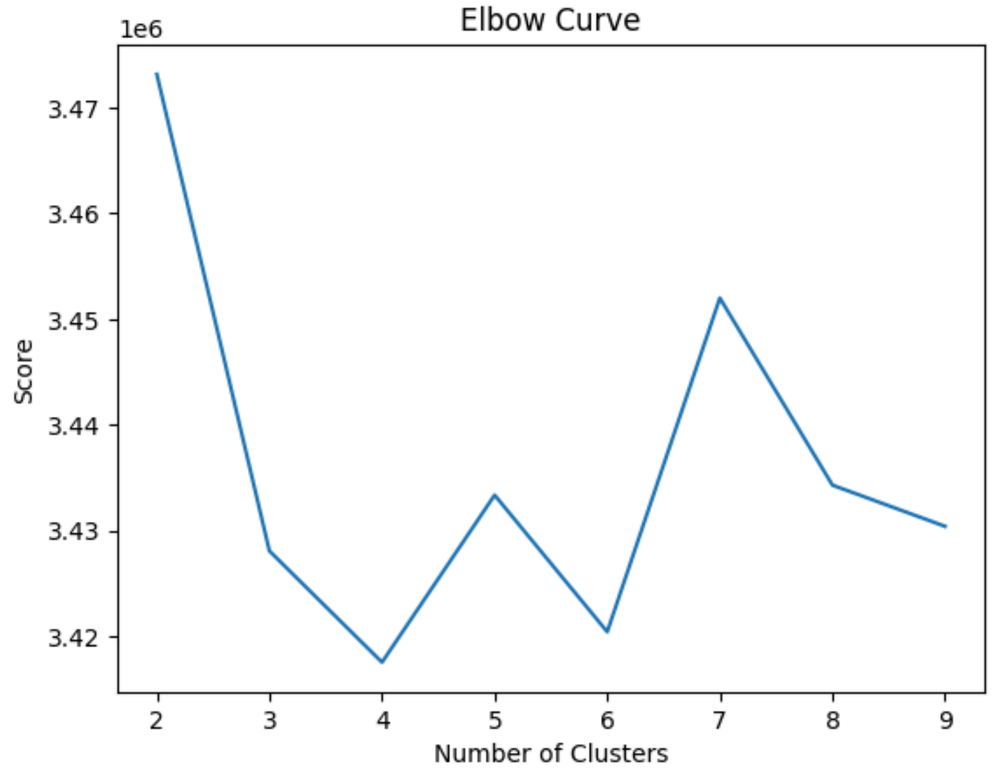
Vectorización: Se aplicó TF-IDF para asignar un valor numérico a cada palabra según su relevancia en el corpus.

Regresión Lineal: Se entrenó un modelo de regresión lineal para predecir el sentimiento basado en las características de las palabras (features).

Análisis TF-IDF en Subreddits más relevantes

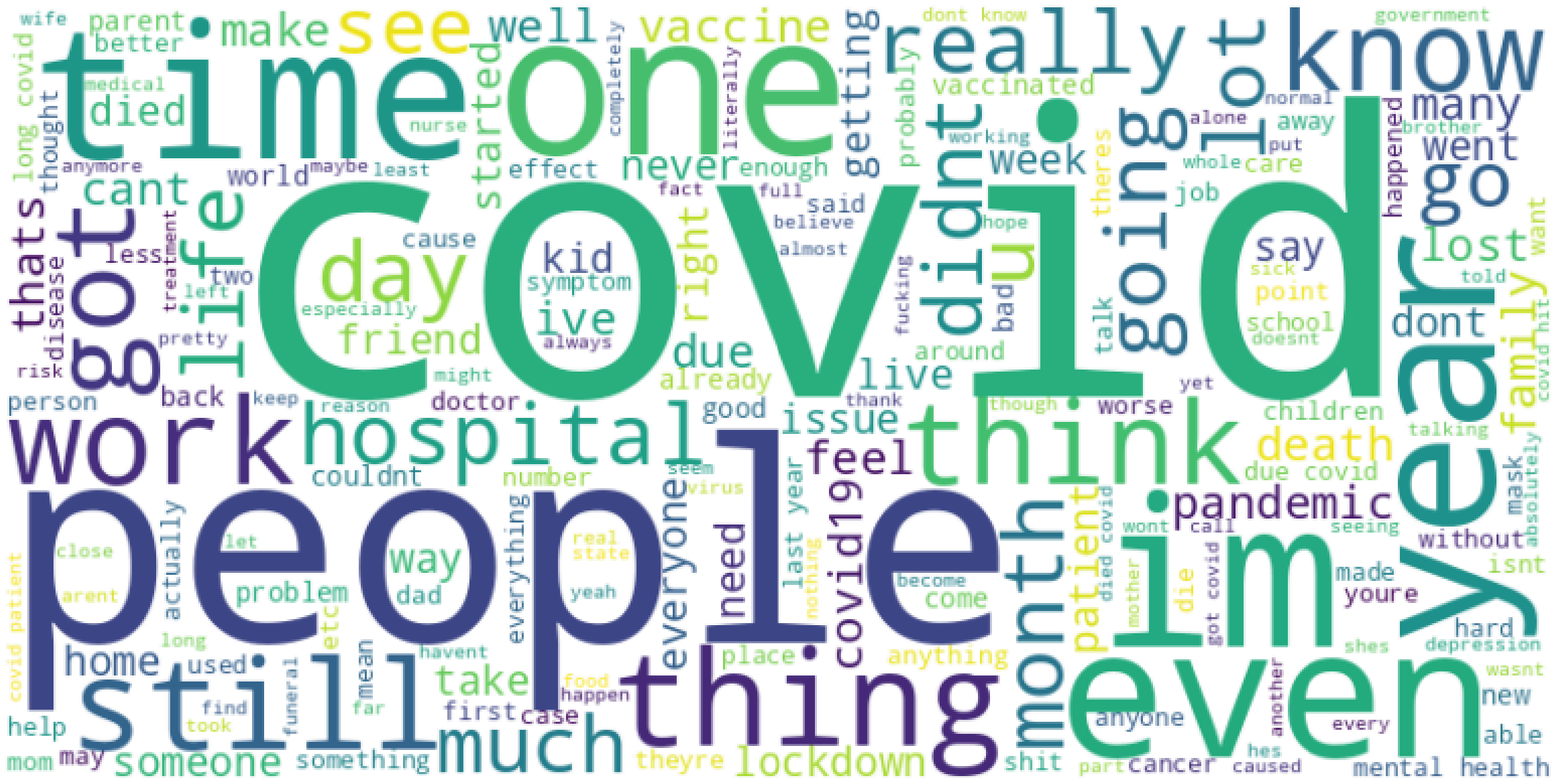
Se utilizó la técnica de vectorización de conteo de palabras para crear una matriz de frecuencias de términos.

Se aplicó IDF para resaltar aquellos términos que eran más representativos en los subreddits, penalizando aquellos que eran demasiado comunes.



Se definieron 3 clústers para agrupar por comunidades

Wordcloud de comentarios sobre la pandemia por comunidad



Primer grupo

Segundo grupo

Tercer grupo

Conclusiones

Optimización de recursos:
Para manejar grandes volúmenes de datos, se utilizaron herramientas como DuckDB y se seleccionaron subconjuntos de datos, permitiendo análisis eficientes sin sobrecargar el sistema.

Análisis de sentimientos y subreddits:
Técnicas como TF-IDF y el análisis de sentimientos revelaron patrones clave sobre los subreddits relevantes y cómo la comunidad percibió la pandemia, mostrando que el sentimiento no siempre está ligado a la popularidad.

Adaptación técnica:
Ante problemas como el Py4JJavaError con GraphFrame, se recurrió a NetworkX, mostrando la importancia de adaptarse a limitaciones tecnológicas para avanzar en el análisis.

Gracias por su atención