Para generar un informe sobre la información de la red social de facebook necesitamos generar los siguientes requerimientos

**Ubicaciones:**

1. Conectar con el archivo **location\_history.json** de la carpeta location
2. Cambiamos el nombre de las columnas a ciudad, longitud, latitud,timestamp ubicación
3. Para transformarla fecha a un formato más comprensible vamos al menú seleccionamos agregar columna y seleccionamos columna personalizada con el nombre fecha ubicación y en la fórmula **#datetime(1970,1,1,0,0,0)+#duration(0,0,0,[timestamp\_ubicación])**

**Nota:** El nombre que se encuentra en los corchetes es el nombre del campo en timestamp

4. En el panel de la derecha de propiedades cambiamos el nombre a la consulta a Ubicación

5. Vamos a la pestaña inicio, cerramos y aplicamos

6. En la sección visualización seleccionamos mapa que se encuentra en la primera columna y cuarta fila

7. En las propiedades de la visualización arrastramos latitud y longitud verificando que en su configuración se encuentre no resumir

8. En la opción formato de la visualización configuramos el titulo “Ubicaciones” con una fuente número 20 y centrado

9. Activamos la propiedad sombra y mapa de calor en formato para determinar las áreas de mayor frecuencia

10. Usar el modo enfoque de la visualización para mayor detalle y luego regresar

**Intereses:**

1. De la carpeta Your Topics importar el archivo your\_topics.json

a. Cambiar el nombre de la consulta a intereses

b. Cambiar el nombre de la columna a intereses

2. De la carpeta other\_logged\_information importar el archivo ads\_interests.json

. Cambiar el nombre de la consulta a intereses publicitarios

a. Cambiar el nombre de la columna a intereses publicitarios

3. En la pestaña inicio de nuestra vista de informe buscamos la opción “Más objetos visuales” de AppSource que es la tienda donde podremos descargar más visualizaciones de las disponibles en la interfaz de power BI desktop

4. Buscamos la visualización nube de palabras “Word Cloud” y pulsamos sobre el botón agregar y esperamos hasta que en la sección de visualizaciones tengas nuestro nuevo objeto visual

5. Arrastramos los intereses en la visualización

* · En las propiedades de la visualización desactivamos Girar el texto

· En la pestaña de general podemos configurar la cantidad de palabras que deseamos observar y el tamaño de la fuente (25 a 30 %)

· Realizamos una nueva nube de palabras para los intereses publicitarios con una cantidad de palabras

o Desactivamos girar texto

**Comportamiento de comentarios por mes**

· Importamos el archivo posts\_and\_comments.json de la carpeta comments\_and\_reactions e importar

· Cambiamos el nombre de la consulta a publicaciones y comentarios

· En el proceso de transformación vamos a desplegar con el botón de la parte derecha de la columna reactions\_v2.data en 3 ocasiones de tal forma que aparezca reaction y actor

· Eliminamos la columna reactions\_v2.attachments con clic derecho quitar

· La columna que tiene como registros la palabra **like**  le cambiamos el nombre a a reacción en ingles

· La columna de actor que aparece el nombre del dueño de la cuenta la quitamos también (reactions\_v2.data.reaction.actor)

· La columna que tiene como nombre reactions\_v2.title la vamos a separar con clic derecho dividir columna por delimitador y seleccionamos el espacio y cada aparición del delimitador

· Quitamos las columnas que no nos aportan mucho como la que tiene la (A,el, un) o el nombre del dueño de la cuenta teniendo en cuenta que solo tenemos una cuenta en análisis

· A la columna que tiene (gustó) le cambiamos el nombre a reacción en español

· A la columna que tiene los datos como (comtanrio,foto) le cambiamos el nombre a tipo de contenido

· Con control clic seleccionamos las columnas que tienen el nombre y apellido de nuestro amigo en Facebook y al pulsa clic derecho usamos combinar columnas y en el separador usamos espacio y el nombre de la nueva columna amigo, finalmente el resto de columnas innecesarias las quitamos

· Vamos al menú agregar una nueva columna, columna personalizada y con el nombre fecha reacción agregamos la formula **#datetime(1970,1,1,0,0,0)+#duration(0,0,0,[reactions\_v2.timestamp])**

· Esa nueva fecha la vamos a duplicar con clic derecho duplicar y pulsamos clic derecho sobre la columna duplicada y en la opción transformar seleccionamos mes para que aparezca del número del mes y en el nombre de la columna asignamos mes

· Repetimos el paso anterior, pero seleccionamos nombre mes en la opción de transformar y en la columna también nombre mes

· Repetimos el paso anterior, pero seleccionamos año en la opción de transformar y en la columna año

· En la pestaña de inicio pulsamos sobre inicio junto a cerrar y aplicar

· Agregamos una nueva visualización de líneas en la que en el eje arrastramos el mes (número) en leyenda el nombre del mes y en valores el recuento del mes

· En las opciones de la visualización en el titulo del gráfico “Comportamiento de comentarios por mes” y en el formato etiqueta de datos

· Agregamos una nueva visualización de filtro y arrastramos el campo año

**Análisis sentimientos:**

El análisis de sentimiento es el proceso de determinar el tono emocional detrás de una serie de palabras. La herramienta de análisis de sentimiento es una técnica automatizada para extraer información significativa de los clientes, relacionada con sus actitudes, emociones y opiniones. Para ello nos apoyaremos de la librería [sentiment-analysis-spanish · PyPI](https://pypi.org/project/sentiment-analysis-spanish/)

1. descargar e Instalar [**python**](https://www.python.org/downloads/)
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)
   1. python -m pip install pandas
   2. python -m pip install matplotlib
   3. python -m pip install sentiment-analysis-spanish
   4. python -m pip install keras tensorflow
   5. python -m pip install sklearn
3. Para probar sobre nuestra terminal el análisis de sentimientos escribiremos en la consola
   1. python
   2. from sentiment\_analysis\_spanish import sentiment\_analysis
   3. sentiment = sentiment\_analysis.SentimentAnalysisSpanish()
   4. print(sentiment.sentiment("me gusta la tómbola es genial"))
   5. print(sentiment.sentiment("me parece terrible esto que me estás diciendo"))
4. Para conectarnos a una hoja de cálculo de google lo que debemos hacer es sobre la hoja de cálculo debemos ir a archivo y publicar en la web y seleccionamos todo el documento y en el tipo xls. El enlace generado lo copiamos (https://drive.google.com/file/d/130N5YXoWp0LH7gd853jF7Q5AH-XZmWNK/view?usp=sharing)
5. En power Bi vamos a obtener desde una web y pegamos el enlace y nos conectamos para posteriormente transformar datos
6. Si tenemos filas iniciales en blanco podemos usar la primera fila como encabezado o quitar dichas filas
7. A la columna que tiene el texto le llamaremos contenido y en la pestaña transformar seleccionamos la opción ejecutar script de python
8. Pegamos el script :

# 'dataset' contiene los datos de entrada para este script

nombre\_columna\_analizar="Comentarios"

nombre\_nueva\_columna="sentimiento"

########################################

from sentiment\_analysis\_spanish import sentiment\_analysis

sentiment = sentiment\_analysis.SentimentAnalysisSpanish()

import pandas as pd

coeficiente\_sentimiento=list()

for i in dataset[nombre\_columna\_analizar].index:

#valordecimal = decimal.Decimal(sentiment.sentiment(dataset[nombre\_columna\_analizar][i]))

texto\_sentimiento= sentiment.sentiment(dataset[nombre\_columna\_analizar][i])

valordecimal =f'{texto\_sentimiento:.2f}'

coeficiente\_sentimiento.append(valordecimal)

dataset[nombre\_nueva\_columna] = pd.DataFrame(coeficiente\_sentimiento,columns=[nombre\_nueva\_columna],dtype ='string')

print(dataset)

1. Si notifica sobre privacidad de los datos pulsamos sobre continuar y sobre las listas de niveles de publicidad en todas dejamos público y continuar
2. Desplegamos sobre el botón junto a la columna de value las columnas que presentaran los sentimientos, verificamos que este las 3 seleccionadas y aceptar
3. Cambiamos el nombre de la columna y en la pestaña de transformar tipo de datos decimal de punto fijo debido a que se encuentra como texto para manipularlo de mejor forma y no podríamos comparar texto con valores para el siguiente paso
4. Agregamos una nueva columna personalizada con el nombre categoria sentimiento y usamos la fórmula

if [sentimiento] < 50 then "NEGATIVO" else if [sentimiento] >50 then "POSITIVO" else if [sentimiento]=50 then "NEUTRAL" else null

**Nota:** No olvidar que entre los corchetes va el nombre del campo que contiene el coeficiente de sentimientos

**Nota:** si deseamos leer estos numeros en notacion cientifica podemos acceder a https://tulengua.es/numeros-texto/

**Amigos con mayor interacción**

· Arrastramos un gráfico de columna apiladas al eje arrastramos el campo amigo de publicaciones y en leyenda tipo de contenido y en valores el recuento de reacción en español

· En la sección filtros vamos a desplegar el campo amigo y en el filtro seleccionamos top N superior y 5 elementos y en por valor arrastramos reacción en español con la propiedad recuento

· Arrastramos el campo amigo de la sección de campos de la tabla publicaciones y en el filtro básico del campo amigo seleccionamos todos los campos menos el que está en blanco

· Cambiamos el título en el formato a amigos con mayor interacción

Análisis de sentimientos en comentarios

Finalmente en la pestaña seleccionamos el tema parque

· Importar de la carpeta comments and reaction importar el archivo comments

· La columna comments\_v2.data la expandimos con el botón de la columna unas 3 veces hasta que aparezcan los comentarios

· Quitamos la columna comments\_v2.attachments

· Quitamos comments\_v2.data.comment.author borramos