Ejercicio de POO (7 puntos (3 puntos+4 puntos)) Creación de nuestra estructura POO

1) En la red social Twitter, cada usuario es propietario de una cuenta (UserAccount) en la que, básicamente, se especifica un alias (que cumple las funciones de identificador único) y un email de contacto. En la cuenta, además, se incluye el conjunto de tweets que el propietario va publicando a lo largo del tiempo.
Como la cantidad de mensajes que maneja la red es inmensa, una característica original de Twitter es que cada usuario puede seleccionar la información que le interesa recibir. De esta manera, el propietario de una UserAccount puede convertirse en seguidor (follower) de otros usuarios, mostrando su interés en los tweets que ellos publiquen. Así, cada vez que un usuario publica un tweet, éste es incluido en el timeline de la UserAccount de cada uno de sus followers (es decir, el timeline se corresponde con el conjunto de tweets recibidos).
En base a estas especificaciones se solicita que:

a) Programe la clase UserAccount y su constructor. Incluya todos sus atributos (alias, email, tweets, followers, timeline) y establezca la visibilidad adecuada. Indica el tipo de datos de todos los atributos y parámetros del constructor y suponga que **ya tiene implementadas correctamente las clases Tweet y Email.**
   Justifique, brevemente, porqué ha seleccionado cada estructura de datos para los atributos.
   No es necesario realizar control de excepciones ni pruebas.

b) Implemente, en UserAccount, un método que permita a un usuario seguir a otro:
   - def follow(user2)
   - Al ejecutar "user.follow(user2)", el usuario user se convertirá en follower de user2.
   - Añada, si lo necesita, métodos auxiliares (por ejemplo, para manejar los followers de user2).
   - No es necesario realizar control de excepciones ni pruebas. Se debe indicar el tipo de datos que recibe cada método (con un comentario)

c) Implemente, en UserAccount, un método que permita a un usuario publicar un Tweet:
   - def tweet(tweet1)
   - Después de ejecutar el método "user.tweet(tweet1)", se deberá actualizar adecuadamente el atributo tweets de user. Además, todos los followers de user habrán recibido el tweet1 en su timeline.
   - Añada, si lo necesita, métodos auxiliares (por ejemplo, para manejar el timeline de los followers).

2) En la red social Twitter, la unidad básica de información se denomina Tweet. Un Tweet es creado en un instante de tiempo concreto (time), contiene un mensaje (message) con un máximo de 140 caracteres de longitud y es publicado por un
usuario (conocido como sender). Además, existen dos tipos de Tweet especiales:

- DirectMessage: Los mensajes directos son Tweets que permiten comunicarse, de manera privada, a dos usuarios dentro de la red. Estos DirectMessage son como Tweets ya que contienen un mensaje (message), son publicados por un emisor (sender) y son creados en un instante de tiempo determinado (time); la única diferencia es que incluyen a otro usuario como receptor (receiver) del tweet.
- Retweet: Cuando un usuario lee un tweet interesante que le ha llegado a su timeline, y quiere reenviarlo a su lista de followers, crea un retweet. Este Retweet

es como un Tweet, es decir, el usuario que lo publica (sender) puede poner un mensaje (message) y lo crea en un tiempo determinado (time); la única diferencia es que el Retweet incluye una referencia al Tweet que se reenvía.

En base a estas especificaciones se solicita que:

a) Implemente las clases Tweet, Retweet y DirectMessage escogiendo la jerarquía más adecuada. Añada los atributos que se especifican en el enunciado y establezca su visibilidad.

- Reutilice todo el código que pueda. Para el atributo time, se recomienda utilizar la clase Date de la librería estándar de Python.
- Suponga que ya tiene implementada correctamente la clase UserAccount.

b) Implemente los constructores de las clases reutilizando al máximo todo el código disponible.

- Además, compruebe las restricciones de datos (por ejemplo, el constructor debería lanzar una excepción si el mensaje que se le pasa, contuviese más de 140 caracteres).
- Recuerde que la librería estándar tiene una función len(string) que devuelve la longitud de un string.

c) Implemente el método **__str__(self)** en las tres clases, reutilizando al máximo todo el código disponible. **Suponga que las clases date y UserAccount ya tiene este método implementado correctamente**.

d) Responda a las siguiente preguntas:

- ¿Deberá modificar los atributos timeline y tweets de la clase UserAccount (definida en el ejercicio 1) para que contenga elementos de la clase hija Retweet? Justifique su razonamiento y, si cree que hay que modificarlos, explique también cómo lo haría.
- ¿Deberá modificar el método def tweet(Tweet tweet1) de la clase UserAccount (definida en el ejercicio 1) para que pueda enviar también objetos de tipo Retweet? Justifique su razonamiento y, si cree que hay que modificarlo, explique también cómo lo haría.

Ejercicio de Análisis EDA de Dataset (3 Puntos(1punto+1punto+1punto))Tiempo estimado 30 minutos

El análisis de sentimiento es la minería contextual de texto que identifica y extrae información subjetiva en el material de origen y ayuda a una empresa a comprender el sentimiento social de su marca, producto o servicio mientras monitorea las conversaciones en línea.

Sin embargo, el análisis de los flujos de redes sociales generalmente se restringe solo al análisis básico de sentimientos y métricas basadas en conteo. Esto es similar a simplemente rascar la superficie y perderse esos conocimientos de alto valor que esperan ser descubiertos. Entonces, ¿qué debe hacer una marca para capturar esa fruta al alcance de la mano?

Para nuestro caso de Twitter, deberá explicar brevemente la estructura del conjunto de datos, Generar y analizar y visualizar el conjunto de datos usando Matplotlib, seaborn y Plotly para obtener tanta información como pueda.

Para más información acerca del análisis de sentimiento y el paso a paso
https://www.slideshare.net/abhishekkrthakur/approaching-almost-any-nlp-problem

1. Importar las necesidades para el análisis de datos (1 Punto)

a) Importe todas las librerías necesarias para realizar el análisis. Aquí os dejo las librerías que son necesarias para realizar el análisis.matplotlib en línea:

```
%matplotlib inline
from plotly import graph_objs as go
import plotly.express as px
import plotly.figure_factory as ff
from collections import Counter

from PIL import Image
from wordcloud import WordCloud, STOPWORDS, ImageColorGenerator


import nltk
from nltk.corpus import stopwords

from tqdm import tqdm
import os
import nltk
import spacy
import random
from spacy.util import compounding
from spacy.util import minibatch

import warnings
warnings.filterwarnings("ignore")

import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))
```

b) A continuación, realice una función auxiliar que genera colores aleatorios que se pueden usar para dar diferentes colores a sus gráficos.

2. Lectura de los datos (1 Punto)

a) Para la lectura de los datos, se han adjuntado varios ficheros csv que deberá leer para realizar el análisis posterior del dataset. Puede usar el siguiente código:

```
train = pd.read_csv('/kaggle/input/tweet-sentiment-extraction/train.csv')
test = pd.read_csv('/kaggle/input/tweet-sentiment-extraction/test.csv')
ss = pd.read_csv('/kaggle/input/tweet-sentiment-
extraction/sample_submission.csv')
```

b) Elimine los valores nulos del fichero Train (NAN simplemente lo eliminaremos)

3. Análisis EDA (1 Punto)

Antes de comenzar, veamos algunas cosas que ya sabemos sobre los datos y nos ayudarán a obtener más información nueva:

- Sabemos que selected_text es un subconjunto de texto

- Sabemos que selected_text contiene solo un segmento de texto, es decir, no salta entre dos oraciones. Por ejemplo: - Si el texto es 'Pasé toda la mañana en una reunión con un vendedor, y mi jefe no estaba contento con ellos. Mucha diversión. Tenía otros planes para mi mañana" El texto seleccionado puede ser "mi jefe no estaba contento con ellos. Mucha diversión' o 'Mucha diversión' pero no puede ser 'Mañana, vendedor y mi jefe,
- Gracias a esta discusión:https://www.kaggle.com/c/tweet-sentiment-extraction/discussion/138520 Sabemos que los tweets neutrales tienen una similitud jaccard del 97 por ciento entre el texto y selected_text
- También como se os enseña aquí https://www.kaggle.com/c/tweet-sentiment-extraction/discussion/138272, hay filas donde selected_text comienza entre las palabras y, por lo tanto, selected_texts no siempre tienen sentido y, dado que no sabemos si la salida del conjunto de pruebas contiene estas discrepancias o no, no estamos seguros de que el preprocesamiento y la eliminación de puntuaciones sea una buena idea o no.

Con esta premisa deberemos realizar las siguientes acciones:

a) Vamos a seleccionar train como conjunto de datos a analizar, para ello debe analizar la distribución de tweets en el conjunto y Dibujar un gráfico de embudo para una mejor visualización.

b) En el contexto de este examen, se nos pide predecir selected_text que es un subconjunto de texto, por lo tanto, las características más útiles para generar serían:
    i. Diferencia en el número de palabras de Selected_text y texto
    ii. Puntuaciones de similitud de Jaccard entre texto y Selected_text

   Por lo tanto, una función útil para el análisis del EDA que anteriormente definimos es la siguiente:

```
def jaccard(str1, str2):
    a = set(str1.lower().split())
    b = set(str2.lower().split())
    c = a.intersection(b)
    return float(len(c)) / (len(a) + len(b) - len(c))
```

   Nota: *Para quien no sepa que es Jaccard Similarity : https://www.geeksforgeeks.org/find-the-jaccard-index-and-jaccard-distance-between-the-two-given-sets/

c) Dada las siguientes conclusiones, proporcine el código necesario para llegar a ellas:
    a. Los tweets positivos y negativos tienen una alta curtosis y, por lo tanto, los valores se concentran en dos regiones estrechas y de alta densidad.
    b. Los tweets neutros tienen un valor de curtosis bajo y hay un aumento en la densidad cerca de los valores de 1

       Para aquellos que no saben:
       - La curtosis es la medida de qué tan pico está una distribución y cuánta propagación está alrededor de ese pico.
       - La asimetría mide cuánto se desvía una curva de una distribución normal

d) Podemos ver en la gráfica de puntuación de jaccard que hay pico para la gráfica negativa y positiva alrededor de la puntuación de 1. Eso significa que hay un grupo de tweets donde hay una gran similitud entre el texto y los textos seleccionados, si podemos encontrar esos grupos, entonces podemos predecir el texto de los textos seleccionados para esos tweets, independientemente del segmento.