Proyecto Nivel I: Programación con Python

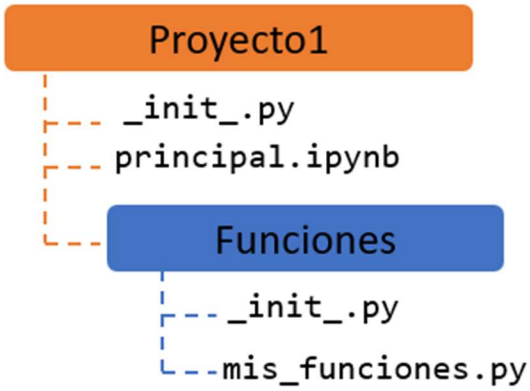
El proyecto tiene dos objetivos:

1. Normalizar un texto
2. Análisis de sentimiento a comentarios de usuarios

Parte 1: Normalización de un texto

El objetivo del proyecto es realizar un simple preprocesamiento del texto del archivo **"articulo.txt".** Este preprocesamiento se conoce como normalización. La normalización generalmente se refiere a una serie de tareas destinadas a poner todo el texto en igualdad de condiciones: eliminando la puntuación, convirtiendo los números a sus equivalentes de palabras, y así sucesivamente. La normalización pone todas las palabras en igualdad de condiciones, y permite que el procesamiento proceda de manera uniforme. Una vez normalizado el texto seremos capaces de analizar nuestros datos de una manera más fácil (aunque este no es el alcance de nuestro proyecto).

El proyecto debe tener la siguiente estructura (paquete):



En el programa **principal.ipynb (Jupyter Notebook)** se realizarán todas las importaciones de librerías y del módulo **mis_fuciones.py.** Además, del código necesario para alcanzar el objetivo del proyecto "Normalizar" el contenido del archivo texto **"articulo.txt"** y del texto **"sentweets_esp.txt"**. Se sugiere utilizar Pycharm para crear el paquete del Proyecto y el módulo **mis_funciones.py.**

El programa debe:

1. Leer el archivo texto (ver código anexo)
2. Contar cuantas palabras tiene el texto antes de preprocesarlo
3. Convertir el texto Normalizado a minúsculas
4. Identificar, mostrar y eliminar signos de puntuación
5. Identificar, mostrar y eliminar las letras acentuadas sustituirlas por la misma letra sin acento
6. Identificar, mostrar y eliminar los números
7. Identificar y mostrar todas las palabras que comience en Mayúscula
8. Identificar y mostrar todas las siglas encontradas en el texto, por ejemplo: SIDA, VIH, OMS, IAVI, GSK, etc.
9. Eliminar todas las palabras "stop words"[1] del texto. Se suministrar un archivo texto **"stopwords_español.txt"** con el listado de palabras en español.

[1] ¿QUÉ SON LAS STOPWORDS O PALABRAS VACÍAS?
En definitiva estas palabras no tienen un significado por si solas, sino que modifican o acompañan a otras, este grupo suele estar conformado por artículos, pronombres, preposiciones, adverbios e incluso algunos verbos. En el procesamiento de datos en lenguaje natural son filtradas antes o después del proceso en sí, no se consideran por su nulo significado.

10. Contar cuantas veces aparece cada una de las palabras que quedan en el texto una vez realizada la normalización.
11. Incluir el control de excepciones en los procesos donde abrimos los archivos de texto.

Parte 2: Análisis de sentimientos

1. Utilizando las funciones creadas anteriormente, debe normalizar el texto **sentweets_esp.txt**. Una vez obtenidas todas las palabras, debe calcular la puntuación de palabras positivas y negativas encontradas en el texto normalizado, para ello debe tomar en cuenta la lista de palabras de los archivos **positive_lex.txt** y **negative_lex.txt**. Recomendación no aplicar la función de eliminar letras acentuadas.
2. Crear dos funciones adicionales para eliminar las urls y las menciones a usuarios por ejemplo @usuario

Para programar los requerimientos antes enumerados debe crear las siguientes funciones en el módulo **mis_funciones.py**:

1. Cree una función que reciba como parámetro un texto y retorne el mismo texto en minúscula
2. Crear una o más funciones reciba como parámetro un texto, retorne los signos de puntuación encontrados en el texto y retorne el texto sin los signos de puntuación.
3. Crear una función que reciba como parámetros un texto, retorne las letras con acento que encontró en el texto y retorne un nuevo texto (donde sustituya la letra acentuada por la misma letra sin acento
4. Crear una función que recibe como parámetro un archivo texto, retorne los números encontrados en el texto y retorne el texto sin números.
5. Crear una función que reciba como parámetro un texto y retorne las palabras que comience en mayúscula encontradas en el texto.
12. Crear una función que reciba como parámetro un texto y retorne las siglas encontradas en el texto
6. Crear una función que reciba como parámetro un texto, muestre las palabras “stop words” encontradas y retorne un nuevo archivo sin las palabras “stop words”.
7. Crear una función que reciba como parámetro un texto, lo divida en palabras y cuente cuantas veces aparece las palabras que quedan en el texto una vez “normalizado”.
8. Crear una función que reciba como parámetro un texto y retorne el texto sin las urls
9. Crear una función que reciba como parámetro un texto y retorne el texto sin las menciones a usuarios @usuarioabc
10. Cree una función que calcule la puntuación total de palabras positivas y negativas encontradas en el texto. La puntuación de cada palabra aparece a la derecha en los archivos **positive_lex.txt** y **negative_lex.txt**.

El alumno debe entregar en la plataforma virtual, un archivo comprimido identificado: PROYECTO_1_ CODIGOCURSO_NOMBRE_APELLIDOS, este archivo debe contener los siguientes documentos:

1. El paquete o estructura de carpetas del Proyecto (archivo comprimido PAQUETE), que además de la estructura de carpetas y los archivos _init_.py, debe contener:
    2. El archivo **mis_funciones.py**
        1. Cada función debe estar documentada utilizando docsstrings: debe especificar que hace la función, que parámetros recibe y el tipo (entero, string, etc.) y que retorna indica el tipo (entero, string, etc.)
    3. El libro generado con Jupyter Notebook (**principal.ipynb**), en este libro deben aparecer:
        1. Importación de las librería utilizadas
        2. Importación del módulo **mis_funciones.py**
        3. Las llamadas a cada una de las funciones, explicar cada paso del proceso, mostrando el texto antes de llamar a cada función, los resultados de la función y el nuevo texto.
        4. Utilizando las celdas de tipo Heading o Markdown, debe documentar cada paso que aplica al texto, los resultados obtenidos.

5. Un apartado de conclusiones, las dificultades que se le presentaron durante el desarrollo del proyecto y lecciones aprendidas al final del libro.

ANEXO

Código para abrir los archivos articulo.txt, stopwords_español.txt

```
# Abrir archivo articulo.txt

      archivo = open('articulo.txt', encoding="utf8")
      articulo = archivo.read()
      articulo


# Abrir el archivo stopwords_español.txt y lo guarda en una lista

      archivo_sw = open('stopwords_español.txt', encoding="utf8")
      a_stop_words = archivo_sw.read()
      stop_words = a_stop_words.split() # Genera una lista con las stop_words


# Abrir el archivo negative_lex.txt y lo guarda en una lista

      archivo_sw = open('negative_lex.txt', encoding="utf8")
      p_negativas = archivo_sw.read()
      p_negativas = p_negativas.split()
      p_negativas # lista de palabras (posición impar) y puntos(posición par)

      ['abatir',
       '-0.554',
       'abochornado',
       '-0.25',
       'abochornar',
       '-0.5',…]

# Abrir el archivo positive_lex.txt y lo guarda en una lista

      archivo_sw = open('positive_lex.txt', encoding="utf8")
      p_positivas = archivo_sw.read()
      p_positivas = p_positivas.split()
      p_positivas # lista de palabras (posición impar) y puntos(posición par)


      ['acertado',
       '0.708',
       'admirable',
       '0.906',
       'admiración',
       '0.45'…]
```