Text Mining & Image Recognition
Proyecto Final

Instrucciones: A continuación verá dos ejercicios que debe completar para poder entregar su proyecto final. Podrá realizar su código en un Notebook con los dos ejercicios. Deberá entregar sus ejercicios por medio de github.

Problema 1 - Word Cloud:
Descargue el Dataset (de click aquí para descargar) el cual contiene aproximadamente 800,000 tweets de diversos temas.

Usando CoLab y expresiones regulares. Determine los 3 usuarios más populares dentro del dataset. Luego arme un corpus el cual contenga los siguientes elementos por cada usuario seleccionado:

- Content: Tweet.
- Metadata: ID, Timestamp, Length (este valor hay que calcularlo).

Posterior a tener sus 3 corpus creados, responda: ¿Razón por la que citan a ese usuario? para esto es necesario que extraiga el contexto de cada tweet y verifique cuales son las palabras que m ?as rodean al nombre de usuario. Para extraer un contexto valido y debido a la naturaleza del tipo de datos que están disponibles en nuestro dataset le recomendamos seguir los siguientes pasos:

1. Remover stopwords.
2. Realizar stemming y lemmatización.
3. Mostrar un wordcloud con el top 10 para cada usuario.

Problema 2 - Digist Recognizer:

MNIST es el conjunto de datos de básico similar a un "hola mundo"de facto de la visión por computadora.

Desde su lanzamiento en 1999, este conjunto de datos clásico de imágenes escritas a mano ha servido como base para los algoritmos de clasificación de evaluación comparativa. A medida que surgen nuevas técnicas de aprendizaje automático, MNIST sigue siendo un recurso confiable tanto para investigadores como para estudiantes.

En este dataset, el objetivo es identificar correctamente los dígitos de un conjunto de datos imágenes escritas a mano.
Los archivos de datos train.csv y test.csv contienen imágenes en escala de grises de dígitos dibujados a mano, de cero a nueve.

Cada imagen tiene 28 píxeles de alto y 28 píxeles de ancho, para un total de 784 píxeles en total. Cada píxel tiene un solo valor de píxel asociado, lo que indica la claridad u oscuridad de ese píxel, y los números más altos significan más oscuro. Este valor de píxel es un número entero entre 0 y 255, inclusive.

El conjunto de datos de entrenamiento, (train.csv), tiene 785 columnas. La primera columna, llamada .etiqueta", es el dígito que dibujó el usuario. El resto de las columnas contienen los valores de píxeles de la imagen asociada.

Cada columna de píxeles en el conjunto de entrenamiento tiene un nombre como pixelX, donde x es un número entero entre 0 y 783, inclusive. Para ubicar este píxel en la imagen, suponga que hemos descompuesto x como x = i * 28 + j, donde i y j son números enteros entre 0 y 27, inclusive. Entonces pixelX se ubica en la fila i y la columna j de una matriz de 28 x 28, (indexado por cero).

Por ejemplo, pixel31 indica el píxel que está en la cuarta columna desde la izquierda y la segunda fila desde la parte superior, como en el diagrama ascii a continuación.

Para este Ejercicio deberá implementar una red neuronal artificial y una red neuronal convolucional para realizar la clasificación de cada digito. Recuerde realizar el Split de prueba y entrenamiento para determinar que tan bueno es su algoritmo de clasificación, tome en cuenta que este es un problema multiclase, antes de iniciar deberá aplicar un reshape a el dataset de entrada para crear imágenes miniatura de 28x28. Notar que el problema tiene 10 salidas (dígitos del 0 al 9).

Al finalizar deberá comparar cual de los dos enfoques funciona mejor mostrando una tabla comparativa de métricas del análisis de clasificación según usted considere. Para esto deberá dividir el dataset en dos, una porción para entrenamient y una porción para prueba.

Puede utilizar colab para desarrollar este laboratorio o Anaconda.