María Alejandra Pérez

mperezo@eafit.edu.co

Enunciado

Realizar una tarea de clasificación para los tweets adjuntos en seleccionados y no seleccionados usando únicamente el campo de “texto”. Para esto es necesario, como mínimo, programar su propia red neuronal y usar otros dos clasificadores o combinaciones de los mismos. Se espera que todos los clasificadores tengan un mejor resultado que el hacer una clasificación aleatoria.

Proyecto 3

Clasificación de Tweets

**HERRAMIENTAS:**

* Spyder

Spyder es el desarrollo científico de PYthon EnviRonment:

* Un poderoso entorno de desarrollo interactivo para el lenguaje Python con funciones avanzadas de edición, pruebas interactivas, depuración e introspección.
* Un entorno de computación numérica gracias al soporte de IPython (intérprete de Python interactivo mejorado) y librerías populares de Python como NumPy (álgebra lineal), SciPy (procesamiento de señales e imágenes) o matplotlib (trazado interactivo 2D / 3D).

Spyder también puede usarse como una biblioteca que proporciona widgets potentes relacionados con consolas para sus aplicaciones basadas en PyQt, por ejemplo, puede ser usado para integrar una consola de depuración directamente en el diseño de su interfaz gráfica de usuario.

En el caso del proyecto, Spyder es el entorno de desarrollo seleccionado.

* NLTK, Natural Language Toolkit

El NLTK (Natural Language Toolkit) es una biblioteca de Procesamiento de Lenguaje Natural que utiliza el lenguaje de programación Python. NLTK es software libre, lo que permite a estudiantes y al personal académico realizar estudios con la herramienta sin necesidad de realizar una inversión económica. Esta herramienta es también de código abierto, lo que lo hace ideal para expandir sus funcionalidades en caso de necesitarlo. El hecho de estar implementada como una biblioteca Python reduce la curva de aprendizaje, y la acerca al mundo académico, cuya mayor parte de integrantes se encuentra familiarizado con este lenguaje de programación.

En el caso del proyecto se usa NLTK para descargar una lista de stop words en español del corpus, esta técnica se usa en la limpieza del texto.

* Scikit-Learn

Machine Learning in Python.

* Herramientas sencillas y eficientes para la minería de datos y análisis de datos.
* Accesible a todo el mundo, y reutilizable en diversos contextos.
* Construido en NumPy, SciPy y matplotlib.
* Código abierto, comercialmente utilizable - licencia BSD.

En el caso del proyecto se usa sklearn para aplicar Naive Bayes Classifier a través de MultinomialNB.

* Listas de Categorías

Estas listas de categorías son creadas con el fin de alimentar al clasificador y al modelo. Se tomaron estas categorías basados en una inspección de ciertos tweets donde se lograba distinguir a que hacen referencia y también en relación y en torno a cultura.

* Lista de Hashtags

Esta lista de hashtags es creada a partir de la observación de ciertos tweets y su frecuencia de uso de menciones en estos. Esto, al igual que las listas de categorías nos ayuda a alimentar al clasificador y al modelo.

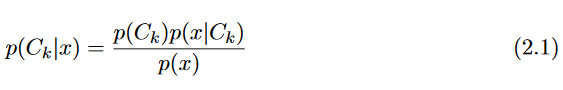
**CLASIFICADORES:**

La mayoría de autores o paginas recomiendan:

* Naive Bayes Classifier

Naive Bayes Classifier, son utilizados ampliamente en la actualidad para problemas de clasificación de textos, tanto en ámbitos académicos como industriales. Los caracteriza una gran simplicidad, que conlleva la ventaja de que son de fácil comprensión y por tanto de fácil uso. A pesar de esta simplicidad, los clasificadores de Bayes ingenuos son bastante eficaces y tienen una gran cantidad de aplicaciones prácticas.

Los clasificadores de Bayes ingenuos se basan en un modelo probabilístico, el Teorema de Bayes.



Según este teorema, la probabilidad de un suceso Ck, dada una situación x viene determinada por el número de veces que se ha producido el suceso Ck en esa situación x, dividido por el número de veces total que se ha encontrado esta situación x (Ecuacion. 2.1).

Aplicados al problema de la clasificación, nos permite inferir la posibilidad que tiene un elemento de pertenecer a cada una de las clases consideradas, teniendo en cuenta cada una de las características de este elemento como un suceso a priori. Los clasificadores de Bayes ingenuos reciben su nombre debido a que toman en cuenta la contribución de cada característica hacia la posibilidad de una clasificación de manera independiente, en contraposición a correlacionar varias características para realizar una estimación más precisa a la hora de clasificar un elemento.

Lo que se usó:

* Un clasificador creado a fuerza bruta, tomando las listas de categorías y la lista de hashtags. Se tomo la lista de categorías y la lista de hashtags y se miró en el archivo tweets si las palabras se encontraban en el texto del tweet y con un contador se fue sumando las veces q aparece y se divide por la longitud del tweet.
* MultinomialNB: se implementó el clasificador de Naive Bayes Multinomial con ayuda de la librería sklearn. Se le asigno el 20% al tamaño del test set y el 80% al tamaño del train set.



**RED NEURONAL:**

El algoritmo más común que se usa junto con una red neuronal es Back-Propagation.

* Back-Propagation

La propagación hacia atrás de errores o retropropagación (del inglés backpropagation) es un algoritmo de aprendizaje supervisado que se usa para entrenar redes neuronales artificiales. El algoritmo emplea un ciclo propagación – adaptación de dos fases. Una vez que se ha aplicado un patrón a la entrada de la red como estímulo, este se propaga desde la primera capa a través de las capas siguientes de la red, hasta generar una salida. La señal de salida se compara con la salida deseada y se calcula una señal de error para cada una de las salidas.

Las salidas de error se propagan hacia atrás, partiendo de la capa de salida, hacia todas las neuronas de la capa oculta que contribuyen directamente a la salida. Sin embargo, las neuronas de la capa oculta solo reciben una fracción de la señal total del error, basándose aproximadamente en la contribución relativa que haya aportado cada neurona a la salida original. Este proceso se repite, capa por capa, hasta que todas las neuronas de la red hayan recibido una señal de error que describa su contribución relativa al error total.

La importancia de este proceso consiste en que, a medida que se entrena la red, las neuronas de las capas intermedias se organizan a sí mismas de tal modo que las distintas neuronas aprenden a reconocer distintas características del espacio total de entrada.

Se puede pensar en una red neuronal como una función matemática compleja que acepta entradas numéricas y genera salidas numéricas. Los valores de las salidas se determinan por: los valores de entrada, el número de nodos de procesamiento ocultos, las funciones de activación de la capa oculta y de la salida y un conjunto de pesos y valores de bias.

Entrenar una red neuronal es el proceso de encontrar valores para los pesos y biases de modo que, para un conjunto de datos de entrenamiento con valores de entrada y salida conocidos, las salidas calculadas de la red coincidan estrechamente con las salidas conocidas.

**TRABAJO FUTURO O DE MEJORA:**

Mejora del sistema de clasificación: lo que ya se tiene planteado más una conexión a un servicio como Twitter, básicamente lo que se hace dentro del tweet es un proceso de limpieza para obtener palabras claves, como hashtags para retroalimentar la base de datos del clasificador, cada hashtag que se agrega va a mejorar el clasificador de esa categoría.

**BIBLIOGRAFIA:**

* <https://riull.ull.es/xmlui/bitstream/handle/915/3082/Analisis%20automatico%20de%20textos%20en%20espanol%20utilizando%20NLTK.pdf?sequence=1>
* <http://bbengfort.github.io/tutorials/2016/05/19/text-classification-nltk-sckit-learn.html>
* <http://iamtrask.github.io/2015/07/12/basic-python-network/>
* <https://github.com/dennybritz/nn-from-scratch>
* <https://www.kaggle.com/c/word2vec-nlp-tutorial/details/part-1-for-beginners-bag-of-words>
* <https://es.wikipedia.org/wiki/Propagaci%C3%B3n_hacia_atr%C3%A1s>
* <https://github.com/shreyans29/thesemicolon/blob/master/Text%20Analytics%20CV.ipynb>
* <https://pythonhosted.org/spyder/index.html>
* <http://scikit-learn.org/stable/>