

**“Análisis de sentimiento en twitter aplicando técnicas de procesamientos de lenguajes naturales y algoritmos de machine learning con Python”**

**Autor**: Estudiante. Rios, Leandro Agustin.

**Institución**: Colegio Universitario IES 21

**Carrera**: Tecnicatura superior en ciencia de datos e inteligencia artificial.

**Materia**: Procesamiento de lenguajes naturales II

**Docente**: Lic. Sufotinsky, Alejandro Julio

Córdoba, Argentina.

2022

# 1.Introducción

El análisis del sentimiento en los mensajes publicados en Twitter ofrece posibilidades de gran interés para evaluar las corrientes de opinión difundidas a través de este medio. Gracias a esto podemos saber qué tan positiva o negativa es la opinión de las personas con respecto a algo o alguien.

Al mismo tiempo, el manejo de grandes volúmenes de datos requiere de herramientas capaces de procesar automáticamente estos mensajes sin perder fiabilidad. Un método de trabajo con respecto a esto son las estrategias que se basan en los procesos de Aprendizaje Automático Supervisado. Su aplicación requiere integrar algunas herramientas del Procesamiento de Lenguajes Naturales (NLP) y tomar como punto de partida un corpus clasificado.

A lo largo del siguiente trabajo se desarrollarán de manera teórica los conceptos con los que vamos a estar trabajando y a través de la práctica esta aplicación veremos cuales son las estrategia que mejor se adaptan.

# 2. Marco teórico

En los últimos años el crecimiento de las técnicas de análisis de sentimiento ha crecido mucho para saber cual es la opinión pública que se tiene con respecto a alguien. Convirtiéndose en un tema de investigación (Eddy Sánchez-DelaCruz,2020). En la imagen que vemos a continuación se muestra la arquitectura de trabajo que diversos autores utilizan para realizar un proyecto de este tipo.



Figura 2.1: Metodología General de Análisis de Sentimiento. Información tomada de (Desh-pande y Rao, 2017)

Puede decirse que analizar el sentimiento en Twitter supone asignar a cada mensaje publicado un valor relacionado con la carga emocional que transmite. En relación a esta carga emocional se pueden distinguir algunos tipos de variables diferentes (Bravo-Marquez, 2014):

* Polaridad: indica si el mensaje tiene un sentimiento positivo o negativo. En algunos análisis se introduce una tercera categoría para clasificar los mensajes neutros. (Es en la cual nos estaremos basando).
* Intensidad: proporciona un valor numérico en relación con la intensidad del sentimiento. Se puede distinguir entre una intensidad positiva y una intensidad negativa.
* Emoción: clasifica el texto según los distintos tipos de emociones, como puede ser la alegría, la tristeza o la ira.

Elegir un algoritmo es un paso crítico en el proceso de aprendizaje automático, por lo que es importante que realmente se adapte al caso de uso del problema en cuestión. Específicamente para la tarea clasificación de opiniones tenemos los algoritmos: máquina de soporte vectorial (SVM) y K-vecinos más cercanos (Knn). Para el desarrollo de nuestra investigación hemos utilizado los algoritmos de aprendizaje supervisado (Denis Cedeno-Moreno,2020).

Podríamos decir que los algoritmos de clasificación son “*Básicamente consiste en que, para clasificar automáticamente una nueva muestra, se tiene en cuenta la información que se pueda extraer de un conjunto de objetos disponibles divididos en clases y la decisión de una regla de clasificación o clasificador.”*(Andres Felipe Jimenez, 2022).

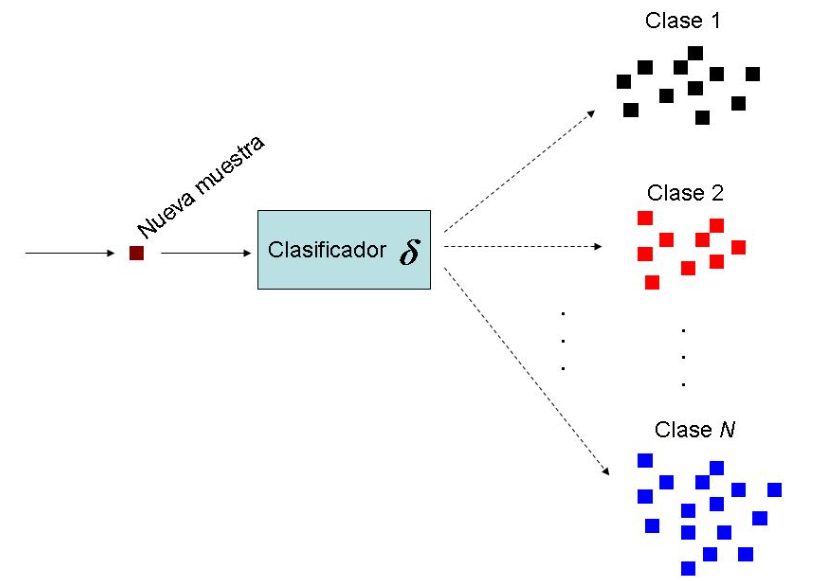


Figura 2.2 - Clasificación Supervisada - Reconocimiento de patrones - Información tomada de (Andres Felipe Jimenez, 2022).

Vamos a explicar brevemente los 2 algoritmos que utilizaremos. Empezaremos con Support Vector Machine (SVM) “...es un algoritmo de aprendizaje supervisado que se utiliza en muchos problemas de clasificación…El objetivo del algoritmo SVM es encontrar un hiperplano que separe de la mejor forma posible dos clases diferentes de puntos de datos. De la mejor forma posible implica el hiperplano con el margen más amplio entre las dos clases, representado por los signos más y menos en la siguiente figura. El margen se define como la anchura máxima de la región paralela al hiperplano que no tiene puntos de datos anteriores. El algoritmo sólo puede encontrar este hiperplano en problemas que permiten separación lineal; en la mayoría de los problemas prácticos, el algoritmo maximiza el margen flexible permitiendo un pequeño número de clasificaciones erróneas.” (1994-2022 The MathWorks, Inc.)

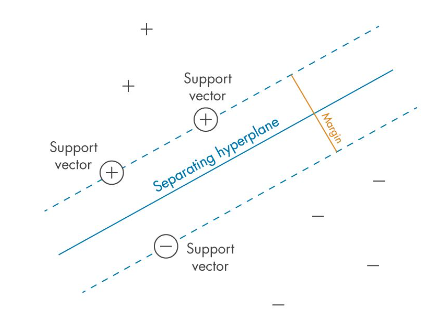


Figura 2.3 - Definición del “margen” entre clases: el criterio que los SVM intentan optimizar. - Imagen tomada de (1994-2022 The MathWorks, Inc.)

Por otro lado tenemos a kNN o k-nearest neighbors Este es un método de clasificación no paramétrico, que estima el valor de la función de densidad de probabilidad o directamente la probabilidad a posteriori de que un elemento x pertenece a la clase Cj a partir de la información proporcionada por el conjunto de prototipos. En el proceso de aprendizaje no se hace ninguna suposición acerca de la distribución de las variables predictoras (Información sacada de Wikipedia,2022)

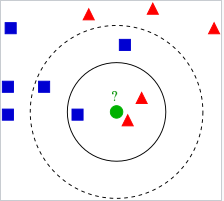


Figura 2.4 - Imagen extraída desde Wikipedia, 2022.

# 3. Marco metodológico

Como estuvimos hablando en la introducción, nosotros contamos con un corpus clasificado de tuits con opciones negativas, positivas y nuestras sobre la opinión popular de las personas sobre el político Argentino Javier Milei. Luego de realizar nuestro análisis tenemos la necesidad de poder saber a través de los tuits de los usuarios cuál es la opinión que se tiene sobre esta persona.

En primera instancia tenemos que poder realizar un algoritmo que pueda adaptarse a las necesidades de clasificar tuits en las categorías de positivos y negativos, esto tiene que estar realizado para el jueves 24 de noviembre de 2022.

A su vez, hay que ir probando distintas combinaciones de técnicas de NLP para poder elegir el modelo final que usaremos en el ambiente productivo. Fecha estimada para el 23 de noviembre de 2022.

Luego de una breve explicación de los fundamentos teóricos que vamos a aplicar, nos basamos en la metodología general del análisis de sentimiento, información tomada de (Desh-pande y Rao, 2017), para ir haciéndolo por las etapas que le corresponden al proyecto.

En una primera instancia, tal como dice la figura 2.1 es la recolección de datos. Para esta parte vamos a utilizar la API que Twitter tiene para poder bajar los tuits. Al tener acceso a esta información aplicamos algunos scripts para eliminar los caracteres especiales como por ejemplo las url, comas, hashtag, entre otros.

Al tener esto realizado nos encargamos de subir dicha información a una base de datos por medio de una API. Se creó una aplicación web que nos permitirá clasificar manualmente cada Tuit con el sentimiento que transmite. Brindándonos la posibilidad de descargar este corpus por medio de un archivo csv.

Cuando llegamos a la parte de procesamiento de lenguajes naturales, vamos a realizar un análisis de cuál es la mejor estrategia para tratar dicha información. Utilizaremos la librería de NLTK para realizarlo. También, analizaremos cual es la lista de stopwords más conveniente para este caso. Verificaremos si nos conviene utilizar por separado el lemmatizer o stemming, o porque no, buscar cual es la mejor combinación para este tipo de proyectos. A su vez, pasaremos todo el corpus que tenemos por una matriz TF-IDF con la cual conseguiremos la frecuencia relativa de las palabras. De esta forma, podremos saber la importancia que tiene cada palabra dentro del conjunto de datos. Esto nos servirá más adelante para poder entrenar el modelo.

Al finalizar con la etapa de preprocesamiento, ya podemos pasar a la parte de probar los algoritmos de clasificación y sus hiperparametros en la etapa de validación. Para esta parte utilizaremos de la librería de Sklearn un algoritmo llamado gridsearchcv que nos permitirá poder encontrar cual es la mejor combinación de los mejores hiperparametros. Solamente compararemos los algoritmos explicados teóricamente en el marco teórico, los cuales son k-NN y Support Vector Machine. En esta etapa se entrenó con 2 conjuntos de datos, uno de ellos solamente se aplicó en la parte de preprocesamiento el filtrado de StopWords y lemmatization, este conjunto lo llamaremos df\_SwL y el otro conjunto de datos fue con todas las funciones de preprocesamiento, en las cuales se aplicó filtrado de StopWord, lemmatization y Stemm, este conjunto lo llamaremos df\_Todo.

En primera instancia el df\_Todo, al finalizar las pruebas en la etapa de validación fue la siguiente. Con respecto a k-NN:

* Mejores hiperparametros:
  + metric: minkowski
  + 'n\_neighbors: 6
  + p: 2
  + weights: distance
* El accuracy: 0.7297596.

Con respecto a SVM:

* Mejores hiperparametros: C: 10
* El accuracy: 0.72370192

Por otro lado con el conjunto de datos df\_SwL los resultados obtenidos fueron los siguientes. Por el lado de k-NN, obtuvimos lo siguiente:

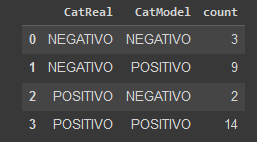
* Hiperparametros:
  + metric: minkowski
  + n\_neighbors: 5
  + p: 2
  + weights: uniform
* El accuracy: 0.7452884615384615

En el caso de SVM

* Hiperparametros: C: 10.0
* El accuracy: 0.720576923076923

Luego de obtener estos resultados en validación, decidimos que el modelo de producción va a ser un k-NN con los datos de df\_SwL.

Ya en la última instancia armamos el modelo de producción. El mismo se entrena con todos los datos del conjunto de df\_SwL implementando un algoritmo de k-NN con los siguientes hiperparametros: metric: minkowski, n\_neighbors: 5, p: 2 y weights: uniform. Para validar el funcionamiento del modelo de producción lo que hacemos es descargar un conjunto de datos nuevos desde twitter, clasificarlos manualmente y luego, compararlos con lo que el modelo pronóstico. De esta forma armamos una matriz de confusión. La misma informa que de un total de 28 tuits con los que hicimos las pruebas. Tenemos 12 en la categoría de negativos, dividiéndolos en 2 subcategorías. En una de ellas contamos con 3 que fueron negativos y 9 fueron falsos negativos. Por el lado de la categoría de los positivos tenemos un total de 16 tuits, de los cuales 14 de ellos fueron positivos y 2 de ellos falsos positivos.



3.1 Imagen obtenida del notebook utilizado para realizar las pruebas de código.

# 4. Conclusión

Luego de todo lo explicado en el marco metodológico, podemos llegar a la conclusión de que el modelo mejoraría su funcionamiento si lo entrenamos con más datos y nuestro accuracy subiría mucho más. Uno de los puntos a mejorar es tratar de bajar la cantidad de falsos negativos que dio el modelo.

De igual forma, para la cantidad de tuits que tenemos podemos decir que el modelo cumple con las expectativas que teníamos. Clasificó muchos de forma acertada.

# 5. Bibliografía

Bravo-Marquez, F.; Mendoza, M. & Poblete, B. (2014): “Meta-level sentiment models for big social data analysis”, Knowledge-Based Systems, vol. 69, no 1, pp. 86–99. http:// dx.doi.org/ 10.1016/j.knosys.2014.05.016

Denis Cedeno-Moreno1, Miguel Vargas-Lombardo21 GISES CIDITIC.(2020) Facultad de Ingeniería de Sistemas Computacionales, Universidad Tecnológica de Panamá,Panamá. https://revistas.utp.ac.pa/index.php/id-tecnologico/article/view/2833/3506

Eddy Sánchez-DelaCruz, Iván Meza, Aníbal Armando Herrera Contreras, 2020, "Modelo para Análisis de Sentimiento Utilizando Aprendizaje Automático", https://www.researchgate.net/publication/346521755\_Modelo\_para\_Analisis\_de\_Sentimiento\_Utilizando\_Aprendizaje\_Automatico

#### Andres Felipe Jimenez - Daniela Galindo - Paula Bastidas, “Informe Algoritmos de Clasificación”, 2022. https://rstudio-pubs-static.s3.amazonaws.com/871603\_89dc2180c83446f6aaa95b3a3788e6bc.html#:~:text=Porcentaje(%25)%20acierto-,%C2%BFQue%20es%20un%20Algoritmo%20de%20clasificacion%20supervisada%3F,patrones%20representantes%20de%20cada%20clase.

k vecinos más próximos, 2022, https://es.wikipedia.org/wiki/K\_vecinos\_m%C3%A1s\_pr%C3%B3ximos