



Procesamiento de Lenguaje Natural o Minería de textos

Tema 4: Aprendizaje Supervisado para clasificación de textos.

Objetivo: El participante identificará el conjunto de características textuales que permiten mejorar los modelos de aprendizaje supervisado para la clasificación de textos, a partir de los métodos existentes para ello y con la ayuda de las bibliotecas implementadas en Python.

Temario:

1. Preprocesamiento de textos
2. Extracción de características
3. Clasificación supervisada: Regresión Logística, Naïve Bayes, Máquinas de Vectores de Soporte

Lecturas:

Applied Text Analysis with Python / by Benjamin Bengfort, Rebecca Bilbro, Tony Ojeda : O'Reilly Media, Inc. [2018] 1 recurso en línea (xii, 334 páginas) : ilustraciones <https://www.oreilly.com/library/view/applied-text-analysis/9781491963036/>

Natural language processing recipes : unlocking text data with machine learning and deep learning using Python / Akshay Kulkarni, Adarsha Shivananda -- [Berkeley, California] : Apress, [2019].-- xxv, 234 páginas : ilustraciones

Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit 1st Edition / by Steven Bird, Ewan Klein, Edward Loper : O'Reilly Media, Inc. [2009] 1 recurso en línea (xi, 512 páginas) : ilustraciones <https://itbook.store/books/9780596516499>

Introducción

El Procesamiento del Lenguaje Natural (PLN) o *Natural Language Processing* (NLP) se centra en el análisis de las comunicaciones humanas y, en concreto, de su lenguaje. Dentro de la Inteligencia Artificial, el PLN es la principal tarea de la lingüística computacional. Un análisis automático sobre cualquier tipo de texto permite clasificar, organizar, buscar o descubrir información no explícita, agilizando tareas que se realizan manualmente e identificando los elementos más relevantes de un escrito.

El PLN consiste en transformar el lenguaje natural en un lenguaje formal, como el de la programación, que los ordenadores puedan procesar. Normalmente, parte de dividirlo en elementos (frases, palabras, etc.) e intentar entender las relaciones entre ellos.

Algunas de las áreas de investigación principales del PLN: Análisis de texto (Lectura), Generación de texto (Escritura), Reconocimiento de voz (Escuchar), Síntesis del habla (Hablar), etc.

Aplicaciones del Procesamiento del Lenguaje Natural

El PLN tiene aplicación en cualquier sector que disponga de grandes cantidades de información no estructurada:

Búsqueda avanzada de información

El análisis de texto permite detectar y recuperar automáticamente información específica en documentos de texto libre de cualquier sector.

Named-entity recognition (NER).

La detección de entidades (personas, lugares, marcas u otros términos) con aprendizaje automático es útil para detectar en qué contextos se mencionan determinadas palabras, por ejemplo, en documentos clínicos o legales.

Anonimización de documentos

Partiendo de la detección de entidades, se puede hacer un primer filtro sobre los datos personales, para asegurar la privacidad. Puede aplicarse los ámbitos de salud, justicia o seguridad.



Detección de topics, similitudes o anomalías en los textos

Con el análisis lingüístico, se detectan temas o patrones en la información, que nos indican ideas relevantes, relaciones, coincidencias o errores. Algo útil, por ejemplo, para la detección de plagio o el control de calidad de documentos.

Chatbots

El PLN es el primer paso en el desarrollo de los asistentes de voz o sistemas conversacionales, siendo esencial en la parte de comprensión del lenguaje.

Clasificación automática de documentos y mensajes

Se pueden etiquetar automáticamente textos según su temática u otras características. Es especialmente útil en ámbitos donde se maneja mucha información o se necesita hacerlo con rapidez, como el sector legal o el de la atención al cliente.

Análisis de sentimiento y de la opinión

Por las palabras que utilizamos, se pueden detectar opiniones acerca de un tema, una persona o un producto en publicaciones de redes sociales, comentarios de clientes o encuestas de clima.

Figure 1 Fuente: <https://www.iic.uam.es/inteligencia-artificial/procesamiento-del-lenguaje-natural/>

Ejemplos de **aplicaciones del PLN¹** de la vida real:

- Análisis de textos
 - Aplicaciones de búsqueda web (Google, Bing, etc.)
 - Errores gramaticales (Ofimática, editores de texto, etc.)
 - Minería de Opiniones (sistema de recomendación de Netflix)
 - Colocación automatizada de anuncios
 - Monitoreo de redes sociales
 - Aplicaciones de seguridad
- Generación de textos
 - La aplicación más exitosa son los datos a los sistemas de texto
- Análisis del habla (reconocimiento de voz)
 - Interfaces de sistemas de habla (sistemas de dictado, sistemas GPS y teléfonos móviles)
 - Enseñanza de un segundo idioma
 - Ayuda a personas con discapacidades

¹ Más aplicaciones, pueden leer en: Justicia de la T., M. d. (2017). Nuevas Técnicas de Minería de Textos: Aplicaciones (pág. 31-39). Universidad de Granada. Tesis Doctorales. Obtenido de <http://hdl.handle.net/10481/46975>



- Generación de habla
 - Ayuda a personas con discapacidades como *Stephen Hawking*
 - Interacción con dispositivos móviles a través de interfaces de PLN

Análisis de textos

Las aplicaciones de análisis de texto requieren transformar texto en alguna otra representación. La representación más común es el modelo de espacio vectorial que es el equivalente de traducir texto en un vector multidimensional de números. ¿Qué deberíamos usar como dimensiones? ¿Frecuencias de palabras, caracteres, lemmas, árboles, raíces o significados?

Recordemos las **herramientas más comunes para el procesamiento de textos**:

- Tokenizador
- Stemmer
- Etiquetador gramatical (POS)
- Lematizador
- Analizador de dependencias

Más allá del nivel sintáctico, análisis de alto nivel como:

- *Desambiguación del sentido de las palabras*, recuperar significados.
- *Reconocimiento de entidades nombradas*, identificar instancias de entidades tales personas o compañías.
- *Resolución de anáforas*, traducir pronombres o frases nominales en entidades.

Co – relación entre complejidad y calidad:

- El uso de herramientas más complejas no es garantía de mejores resultados para la mayoría de los problemas. Tal vez se deba a que las herramientas de nivel superior tienen menos precisión.
- La mayoría de los investigadores de PLN utilizan al menos un etiquetado POS. Es común utilizar lematización o análisis sintácticos.

Modelo de espacio vectorial

- Consiste en representar el texto como un vector de características y valores
- Las características comunes son:
 - "Palabras"
 - N-gramas de palabras (se refiere a la extracción de N "palabras" contiguas del texto)
 - N-gramas de caracteres (se refiere a la extracción de N "caracteres" contiguos del texto)
 - N-gramas sintácticos (se refiere a la extracción de N elementos conectados de un árbol sintáctico)



Ejemplo:

1. Representar las siguientes oraciones como vectores:
 - Somos héroes
 - Luis Pedro y Juan Luis son héroes
 2. Identificar le vocabulario:
 - a. [Somos, héroes, son, Luis, Pedro, y, Juan] ← Dimensiones del vector
 3. Traducir el texto en vectores usando vocabulario como dimensiones y frecuencias de palabras como valores:
 - Somos héroes → [1, 1, 0, 0, 0, 0, 0]
 - Luis Pedro y Juan Luis son héroes → [0, 1, 1, 2, 1, 1, 1]
 - Los valores comunes son:
 - Existencia (¿La característica está presente en el texto o no?) 1 si una característica específica está presente en un texto objetivo, 0 en caso contrario
 - TF - Frecuencia de término: conteo de apariciones de una característica específica en un texto de destino
 - IDF
- IDF: Frecuencia inversa de documentos
- Mide la cantidad de información que brinda la palabra
 - Los términos comunes tienen valores bajos y los términos raros tienen valores altos
 - Necesita un corpus de documentos para su cálculo

$$idf(t, D) = \log \frac{N}{|\{d \in D: t \in d\}|}$$

N es el número de documentos en el corpus
D es el conjunto de documentos
t es el término
El denominador representa la cantidad de documentos que contienen **t**

- TF-IDF
- TF-IDF:
- Es el producto de la frecuencia de los términos y la frecuencia inversa de los documentos
 - Es una estadística numérica que pretende reflejar la importancia de una palabra para un documento en una colección o corpus
- La selección de las características y sus valores es subjetiva
 - Modelo ampliamente utilizado en informática

Identificación de características de textos

- Toda la información se encuentra en el texto
- Las características pueden ser extraídas del texto con diferentes granularidades



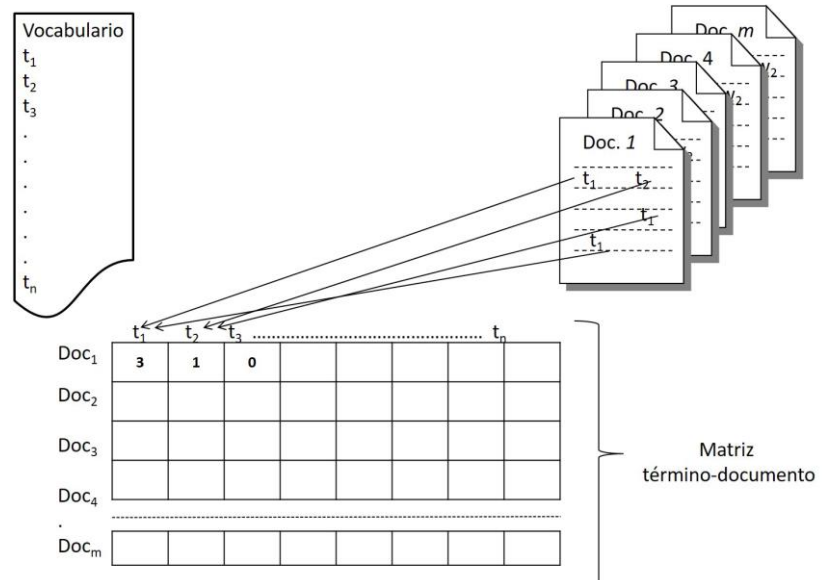
Tipos de características textuales:

- **Palabras**
 - Por mucho, la clase más común de características
 - Manejo de palabras comunes: palabras cerradas (*stop words*)
 - La palabra exacta
 - Normalización: hacer minúsculas vs. dejar como está
 - Raíces / Lematización
 - La palabra etiquetada
- **N-gramas:**
 - Según las tareas de clasificación, las características pueden venir desde adentro palabras y secuencias de palabras
 - bigramas, trigramas, *n-grams*: "Casa Blanca"
 - subsecuencias de caracteres en palabras: "ing", "ion", ...
- Características de las palabras: Capitalización
- Etiquetas de categorías gramaticales de una oración
- Estructura gramatical, análisis de oraciones
- Signos de puntuación
- Agrupación de palabras de significado similar, semántica
 - {comprar, adquirir}
 - {Sr., Sra., Dr., Prof.}; Números / Dígitos: fechas

Matriz término-documento

Un corpus puede ser representado por una matriz término-documento:

- Cada frecuencia de token individual (normalizada o no) se trata como una característica.
- El vector de todas las frecuencias de tokens para un documento dado se considera una muestra multivariada.



Extracción de características textuales

- Scikit-learn para extracción de características de texto:
http://scikit-learn.org/stable/modules/feature_extraction.html#text-feature-extraction
- Scikit-learn proporciona utilidades para extraer las características del texto:
 - Tokenización
 - Conteo de ocurrencias de tokens



- Normalización, esquemas de pesado
- **Vectorización** es el proceso general de convertir una colección de documentos de texto en vectores de características numéricas

Conjuntos de datos dispersos de alta dimensión

- Los documentos utilizan un subconjunto pequeño de palabras utilizadas en el corpus, la matriz resultante tendrá muchos valores de características que son ceros.
- Para poder almacenar dicha matriz en la memoria y además para acelerar las operaciones algebraicas de matriz/vector, scikit-learn permite usar una representación dispersa (como las implementaciones disponibles en el paquete `scipy.sparse`).



0	0	0	0	9	0
0	8	0	0	0	0
4	0	0	2	0	0
0	0	0	0	0	5
0	0	2	0	0	0

Rows	Columns	Values
5	6	6
0	4	9
1	1	8
2	0	4
2	2	2
3	5	5
4	2	2

Ejercicio4(es)-Aprendizaje Supervisado.ipynb

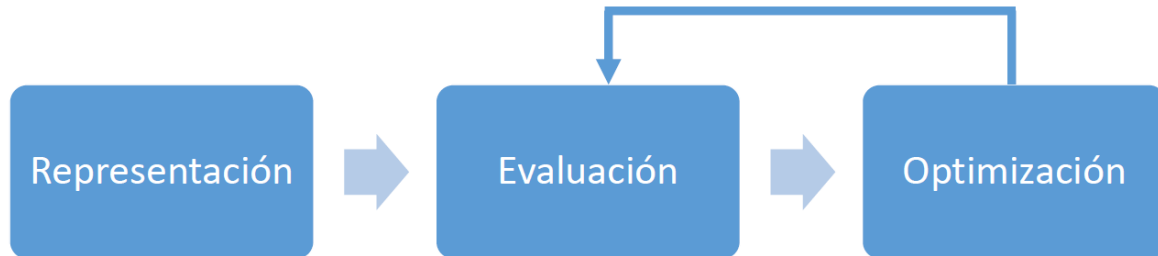
Según el propósito que persigan los algoritmos de aprendizaje automático, éstos se clasifican en dos categorías principales: aprendizaje supervisado y aprendizaje no supervisado. La principal diferencia entre estas dos familias se encuentra en los datos de entrenamiento. En el **aprendizaje supervisado** los resultados que se desean obtener del modelo son conocidos previamente. Siendo utilizados para guiar su entrenamiento. Por otro lado, en el **aprendizaje no supervisado** el resultado deseado no se utiliza durante el entrenamiento. En la mayoría de los casos tampoco se conoce previamente, siendo descubierto durante el proceso aprendizaje.

Aprendizaje supervisado: En el entrenamiento de los algoritmos de aprendizaje supervisado, además de los datos necesarios para realizar la predicción, es necesario disponer de una característica objetivo para cada una de las instancias.

- **Clasificación:** se desea *obtener una categoría*.
- **Regresión:** en estos se intenta *predecir un valor continuo*.

Aprendizaje no supervisado: A diferencia de los algoritmos de aprendizaje supervisado, en los no supervisados no es necesario *disponer de la respuesta* correcta en los datos de entrenamiento. Ya que no se busca la reproducción de un resultado conocido, sino el *descubrimiento de nuevos patrones o resultados*.

Flujo básico de un esquema de aprendizaje automático



Elegir:

- Representación de características
- Clasificador a utilizar

Elegir:

- ¿Qué criterio distingue un clasificador bueno o malo?

Elegir:

- ¿Cómo buscar la configuración que da mejores clasificaciones para este criterio de evaluación?

¿Qué hacemos?

- Representación del texto.
- Entrenar el modelo de aprendizaje automático.
- Evaluar los resultados de predicción obtenidos.
- Refinar ciclo para la obtención de mejores resultados; repetir proceso.



Generalización, sobreajuste y falta de ajuste

La **generalización** se refiere a la capacidad de un algoritmo para proporcionar predicciones precisas para datos nuevos que no se habían visto anteriormente.

Supuestos:

- Los datos futuros no vistos (conjunto de prueba) tendrán las mismas propiedades que los conjuntos de entrenamiento actuales.
- Se espera que los modelos que son precisos en el conjunto de entrenamiento sean precisos en el conjunto de prueba.
- Pero eso puede no ocurrir si el modelo entrenado se ajusta muy específicamente al conjunto de entrenamiento.



Los modelos que son demasiado complejos para la cantidad de datos de entrenamiento disponibles **se sobreajustan** y no es probable que se generalicen bien a los nuevos ejemplos. Los modelos que son demasiado simples, que ni siquiera funcionan bien en los datos de entrenamiento, les **falta ajuste** y tampoco es probable que se generalicen bien.

Clasificación supervisada²:

- Naïve Bayes
- Regresión Logística
- Máquinas de Vectores de Soporte

En clasificación, la tarea principal es elegir la etiqueta de clase correcta para un input dado. Cada input se considera aisladamente de los otros y las etiquetas están definidas con antelación. Aprender un modelo de clasificación sobre propiedades ('características') y su importancia ('pesos') de ejemplos etiquetados:

- X: Conjunto de atributos o características $\{x_1, x_2, \dots, x_n\}$
- Y: Una etiqueta de 'clase' de un conjunto de etiquetas $Y = \{y_1, y_2, \dots, y_k\}$
- Aplicar el modelo en nuevas instancias para predecir la etiqueta.

Paradigmas:

- Single label vs. Multi-label
 - $|Y| = 2$: Clasificación binaria o booleana. Existen sólo dos clases (pertenece o no)
 - $|Y| > 2$: Clasificación multi-clase. Existen más de dos clases posibles (etiquetas)
- Orientada a documentos vs. Orientada a categorías
 - En la *clasificación orientada a documentos*, se busca clasificar un conjunto de textos, para otorgar a cada uno una categoría.
 - En la *clasificación orientada a categorías*, se quiere encontrar todos los documentos que pertenecen a una categoría o categorías determinadas.
 - Sólo es posible cuando todas las categorías y documentos se conocen de antemano.
- Clasificación *hards* vs. *soft*
 - Hard: sistemas completamente automáticos
 - Soft: sistemas semiautomáticos
 - Muchos clasificadores otorgan un valor entre $[0,1]$ para cada documento relacionado con una etiqueta. Se llaman CSV (*categorization status value*), y alguien tiene que establecer un *threshold* para acabar el proceso de decisión.

Clasificador Bayes Ingenuo (Naïve Bayes)

- Clasificador probabilístico
- Se llaman "ingenuo" porque asumen que las características son independientes dada la clase.

² Extra: <https://sitiobigdata.com/2019/12/24/clasificacion-de-aprendizaje-automatico-supervisado/>



- Es uno de los algoritmos de aprendizaje inductivo más eficientes y eficaces para el aprendizaje automático y la minería de datos (Zhang, 2004)³.
- Altamente eficiente para aprender y predecir, pero el rendimiento de generalización puede ser peor que los métodos de aprendizaje más sofisticados.
- Puede ser competitivo para algunas tareas.

Entre la amplia variedad de clasificadores existentes del método de Bayes, se puede citar:

- **Bernoulli**: características binarias
- **Multinomial**: características discretas
- **Gaussian**: características continuas / de valor real
 - Estadísticas calculadas para cada clase: media, desviación estándar

Regresión Logística

- Del inglés, *logistic regression*, a pesar de su nombre, es un modelo lineal para la clasificación en lugar de la regresión.
- Se utiliza para predecir la probabilidad de una variable dependiente categórica.
- La variable dependiente es una variable binaria que contiene datos codificados como 1 – 0, sí – no, abierto – cerrado, etcétera. Es útil para modelar la probabilidad de un evento ocurriendo como función de otros factores.
- El análisis de regresión logística se enmarca en el conjunto de Modelos Lineales Generalizados (GLM por sus siglas en inglés) que usa como función de enlace la función *logit*. Las probabilidades que describen el posible resultado de un único ensayo se modelan, como una función de variables explicativas, utilizando una función logística.

$$f(x) = \frac{L}{1 + e^{-k(x-x_0)}}$$

Donde:

- x_0 : valor del punto medio del sigmoide
 - L : máximos valor de la curva
 - k : radio de crecimiento logístico o inclinación de la curva
- El modelo de regresión logística binaria tiene extensiones a más de dos niveles de la variable dependiente: las salidas categóricas con más de dos valores se modelan mediante regresión logística multinomial.
 - La regresión logística requiere tamaños de muestra bastante grandes. La razón por la cual es ampliamente utilizada, a pesar de los algoritmos avanzados como redes neuronales profunda, es porque es muy eficiente y no requiere demasiados recursos computacionales que hacen que sea asequibles ejecutar la producción.

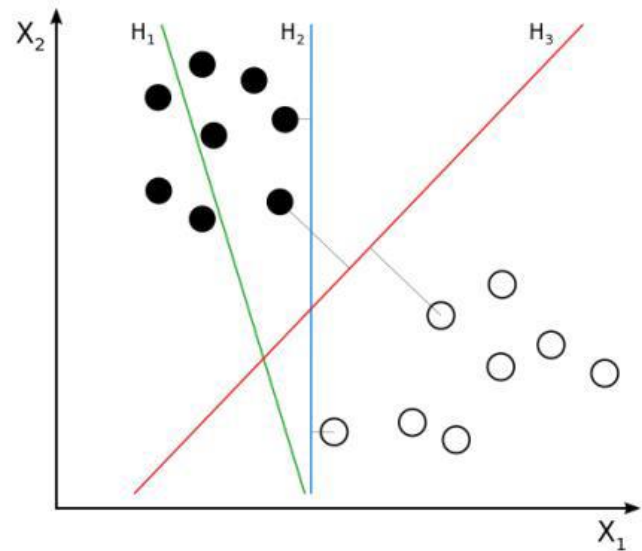
³ Zhang, H. (2004). *The Optimality of Naïve Bayes*.

Máquinas de Vectores de Soporte (SVM)

- Se utilizan para la detección de clasificación, regresión y valores atípicos.
- Dado un conjunto de ejemplos de entrenamiento, una SVM los representa como puntos en el espacio, separando las clases por la mayor distancia posible, observa en la figura.
- Cuando las nuevas muestras se ponen en correspondencia con dicho modelo, según su proximidad serán asignadas a una u otra clase, es decir, si un punto nuevo pertenece a una categoría o la otra.

Las ventajas de las SVM son:

- Son efectivos en casos donde el número de dimensiones es mayor que el número de muestras.
- Utilizan un subconjunto de puntos de entrenamiento en la función de decisión, llamados vectores de soporte, por lo que también es eficiente desde el punto de vista de la memoria.
- Son versátiles, se pueden especificar diferentes funciones del *Kernel*⁴ para la función de decisión. Se proporcionan núcleos comunes o se pueden definir núcleos personalizados para tipos de datos específicos.



H1 no separa las clases. H2 sí, pero H3 lo hace con la separación máxima

Fuente: https://es.wikipedia.org/wiki/M%C3%A1quinas_de_vectores_de_soporte

Las desventajas de las SVM incluyen:

- En caso de que el número de funciones es mucho mayor que el número de muestras se debe de evitar el ajuste excesivo al elegir las funciones del *Kernel* y el término de regularización es crucial.
- Las SVM no proporcionan estimaciones de probabilidad directamente, estas se calculan utilizando una costosa validación cruzada de cinco veces.

Para el empleo en el desarrollo de dicha técnica es importante esclarecer el parámetro C (fuerza de la regularización⁵), teniendo las siguientes las características:

⁴ Un kernel es una medida de similitud (producto punto modificado) entre puntos de datos. En los modelos el *kernel* por defecto es "rbf" (para función de base radial, otro tipo puede ser "polynomial") y como parámetros del kernel: gamma (γ) anchura del kernel RBF. Tradicionalmente C y gamma se optimizan al mismo tiempo.

⁵ Se refiere a regularización, a la capacidad de un algoritmo para proporcionar predicciones precisas para datos nuevos que no se habían visto anteriormente.



- Valores más grandes de C (menos regularización): significa que ajusta los datos de entrenamiento lo mejor posible y cada punto de datos individual es importante para clasificar correctamente.
- Valores más pequeños de C (más regularización): significa más tolerante a errores en puntos de datos individuales.

Para problemas de clasificación multi – clase (del inglés, *multi - class*) en un conjunto de datos como parte del conjunto de métodos de las SVM, se pueden citar los algoritmos: SVC (*C-Support Vector Classification*), NuSVC (*Nu-Support Vector Classification*) y LinearSVC (*Linear Support Vector Classification*). SVC y NuSVC implementan el enfoque de “one-against-one” (“uno contra uno”) (Knerr, Personnaz, & Dreyfus, 1990⁶) para la clasificación de clases múltiples. Por otro lado, LinearSVC implementa la estrategia multi - clase “one-vs-the-rest” (“uno frente al resto”), por lo que entrena a los modelos n_class (si solo hay dos clases, solo se entrena un modelo).

Para los métodos de SVM con *kernel*, la eficiencia (velocidad de tiempo de ejecución y uso de la memoria) disminuye a medida que aumenta el tamaño del conjunto de entrenamiento; y se necesita normalización de los datos de entrada y ajuste de parámetros. En este sentido, el LinearSVC su parámetro de *kernel* es lineal, teniendo como ventajas: simple y fácil de entrenar, predicción rápida, escala bien a conjuntos de datos muy grandes, funciona bien con datos dispersos y las predicciones son relativamente fáciles de interpretar. Sin embargo, para conjuntos de datos pequeños, otros modelos pueden tener un rendimiento de generalización superior.

Métricas de evaluación en modelos predictivos

El modelado predictivo funciona sobre el principio de retroalimentación constructiva: se construye un modelo, se obtiene retroalimentación de las métricas, se realiza mejoras y se continua hasta lograr una precisión deseable. **Las métricas de evaluación** explican el rendimiento de un modelo; aspecto fundamental del aprendizaje automático por su capacidad para discriminar entre los resultados del modelo. El objetivo no es construir un modelo predictivo y pensar que sea éste el mejor posible, es mejor crear y seleccionar un modelo que proporcione una alta precisión en los datos de muestra. La evaluación es importante para comprender la calidad del modelo o la técnica, para refinar los parámetros en el proceso iterativo de aprendizaje y para seleccionar el modelo o la técnica más aceptable de un conjunto dado de modelos o técnicas. La elección de la métrica depende completamente del tipo de modelo y el plan de implementación del modelo.

Matriz de confusión para tarea de predicción binaria

Las tareas de predicción binaria dan lugar a cuatro clasificaciones posibles para cada instancia: un verdadero positivo (TP, del inglés: *true positive*), un verdadero negativo (TN, del inglés: *true negative*), un

⁶ Knerr, S., Personnaz, L., & Dreyfus, G. (1990). Single-layer learning revisited: A stepwise procedure for building and training neural network. *Neurocomputing: Algorithms, Architectures and Applications*, NATO ASI, Berlin: Springer-Verlag.



falso positivo (FP, del inglés: *false positive*) o un falso negativo (FN, del inglés: *false negative*); situación que se puede representar como una matriz de confusión (también llamada tabla de contingencia).

En la figura, las clasificaciones que se encuentran a lo largo de la diagonal principal de la tabla son las clasificaciones correctas, es decir, los verdaderos positivos y los verdaderos negativos; sin embargo, los otros campos significan errores de modelo. Para un modelo perfecto se analizan los campos verdadero positivo y verdadero negativo, los otros campos se establecerían en cero.

Confusion Matriz		Observed			
		True	False		
Model Predicted	True	True Positive (TP)	False Positive (FP)	Positive Predictive Value	TP/(TP+FP)
	False	False Negative (FN)	True Negative (TN)	Negative Predictive Value	TN/(FP+TN)
		Sensitivity TP/(TP+FN)	Specificity TN/(FP+TN)	Accuracy = (TP+TN)/(TP+FP+FN+TN)	

Matriz de confusión y fórmulas de métricas de rendimiento

Exactitud

Una serie de métricas de rendimiento del modelo pueden derivarse de la matriz de confusión; la más común es la exactitud (del inglés, *accuracy*), aunque también existen otras métricas que son definidas a continuación, como parte de una matriz de confusión:

- *Valor predictivo positivo (precisión = 0.79)*: la proporción de casos positivos que se identificaron correctamente.
- *Valor predictivo negativo*: la proporción de casos negativos que se identificaron correctamente.
- *Sensibilidad o Exhaustividad (recall = 0.60)*: la proporción de casos positivos reales que están correctamente identificados.
- *Especificidad*: la proporción de casos negativos reales que están correctamente identificados.

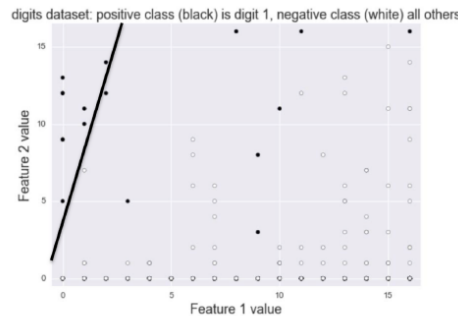
	Predicción negativa	Predicción positiva	
Etiqueta negativa	TN = 400	FP = 7	
Etiqueta positiva	FN = 17	TP = 26	
			N = 450

La exactitud (*accuracy = 0.95*) de la clasificación es el número de predicciones correctas realizadas, como una proporción de todas las predicciones realizadas. Esta es la evaluación más común de la métrica para problemas de clasificación, también es la más mal utilizada. Realmente solo es adecuado cuando hay un

número igual de observaciones en cada clase (lo que rara vez es el caso) y que todas las predicciones y errores de predicción son igualmente importantes, lo que a menudo no es el caso (Brownlee, 2016⁷).

Equilibrio entre precision y recall

Alta precision,
bajo recall

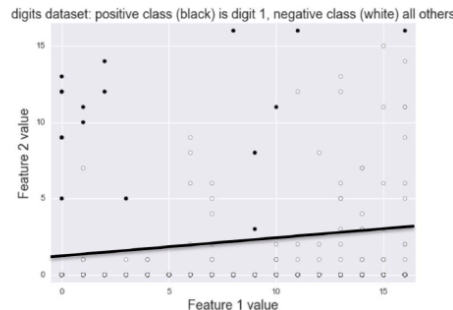


TN = 435	FP = 0
FN = 8	TP = 7

$$\text{Precision} = \frac{TP}{TP+FP} = \frac{7}{7} = 1.00$$

$$\text{Recall} = \frac{TP}{TP+FN} = \frac{7}{15} = 0.47$$

Baja precision,
alto recall



TN = 408	FP = 27
FN = 0	TP = 15

$$\text{Precision} = \frac{TP}{TP+FP} = \frac{15}{42} = 0.36$$

$$\text{Recall} = \frac{TP}{TP+FN} = \frac{15}{15} = 1.00$$

Valor F1

El Valor-F (denominada también F-score o medida-F) en estadística es la medida de precisión que tiene un *test*. Se emplea en la determinación de un valor único ponderado de la precisión y la exhaustividad (*recall*) (Beitzel., 2006⁸). Se calcula:

$$F1 = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} = \frac{2 * TP}{2 * TP + FN + FP}$$

El puntaje de F1 es necesario cuando se desea buscar el equilibrio entre *precision* y *recall*, alcanzando su mejor valor en 1 (*precision* y *recall* perfectas) y el peor en 0.

Evaluación multi-clase

Para generalizar esto a las clases múltiples, asumiendo que se tiene un clasificador *One-vs-All (OvA)* (Uno contra todo) o multi - clase, el puntaje F1 se puede calcular de dos formas diferentes: promedio micro y promedio macro:

⁷ Brownlee, J. (2016). *Metrics To Evaluate Machine Learning Algorithms in Python*. Obtenido de Machine Learning Mastery: <https://machinelearningmastery.com/metrics-evaluate-machine-learning-algorithms-python/>

⁸ Beitzel., S. M. (2006). *On Understanding and Classifying Web Queries (Ph.D. thesis)*.



- En “micro” se calcularía el rendimiento, por ejemplo, la precisión, a partir de los positivos verdaderos individuales, negativos verdaderos, falsos positivos y falsos negativos del modelo de la clase k .

$$PRE_{micro} = \frac{TP_1 + \dots + TP_k}{TP_1 + \dots + TP_k + FP_1 + \dots + FP_k}$$

- En “macro”, se promedian los rendimientos de cada clase individual.

$$PRE_{macro} = \frac{PRE_1 + \dots + PRE_k}{k}$$

Ejercicio4(es)-Aprendizaje Supervisado.ipynb

Actividad Independiente:

El discurso de odio se define comúnmente como cualquier comunicación que menosprecia a una persona o un grupo en función de algunas características. En el año 2019 se celebró la competencia: SemEval-2019 *International Workshop on Semantic Evaluation* (<https://alt.qcri.org/semeval2019/>) planteándose 12 tareas. De la Task 5: “Multilingual detection of hate speech against immigrants and women in Twitter (*hatEval*)” se planteó lo referente al discurso de odio en redes sociales, en específico la red social Twitter (<https://competitions.codalab.org/competitions/19935>).

- Trabajar con la tarea A, dejando a libre escoger uno de los 2 idiomas.
- Realice diferentes pruebas (mínimo 3). Anote los resultados obtenidos por cada una de ellas, y asuma diferentes características en el entrenamiento del clasificador binario.

Sugerencias:

Para el preprocesamiento de los textos, puede:

- Estandarizar el texto a minúsculas.
- Eliminar las menciones a usuarios (@user)
- Eliminar las url's
- Eliminar los emojis
- Las abreviaturas, contracciones y slangs sustituirlas por el texto equivalente
- Eliminar palabras funcionales
- Verificar si existen cifras numéricas, las cuales pueden ser reemplazadas por algún término o eliminarlas.
- Tratamiento con los hashtags
- Eliminar caracteres raros y especiales
- Eliminar signos de puntuación
- Estandarizar las secuencias de varios espacios en blanco, tabuladores y saltos de línea
- Entre otras...



Posibles características para tenerse en cuenta:

- N-gramas de caracteres
- N-gramas de palabras
- N-gramas de etiquetas POS
- N-gramas de saltos de palabras (skip-gram)
- N-gramas de palabras funcionales
- N-gramas de símbolos de puntuación
- Entre otras...

Conclusiones

- El análisis de texto es utilizado actualmente por muchas aplicaciones comerciales en este momento. Las grandes compañías como Oracle, Google, Microsoft y Facebook tienen sus propios grupos de investigación de PLN, por lo que es un buen momento para trabajar en esta área.
- El aprendizaje automático es ampliamente utilizado, tanto por investigadores como por ingenieros.