



OPTATIVA

Recuperación **Avanzada** de Información

Dr. J. Federico Medrano
@jfedemedrano

Unidad N° 3 – Parte 1

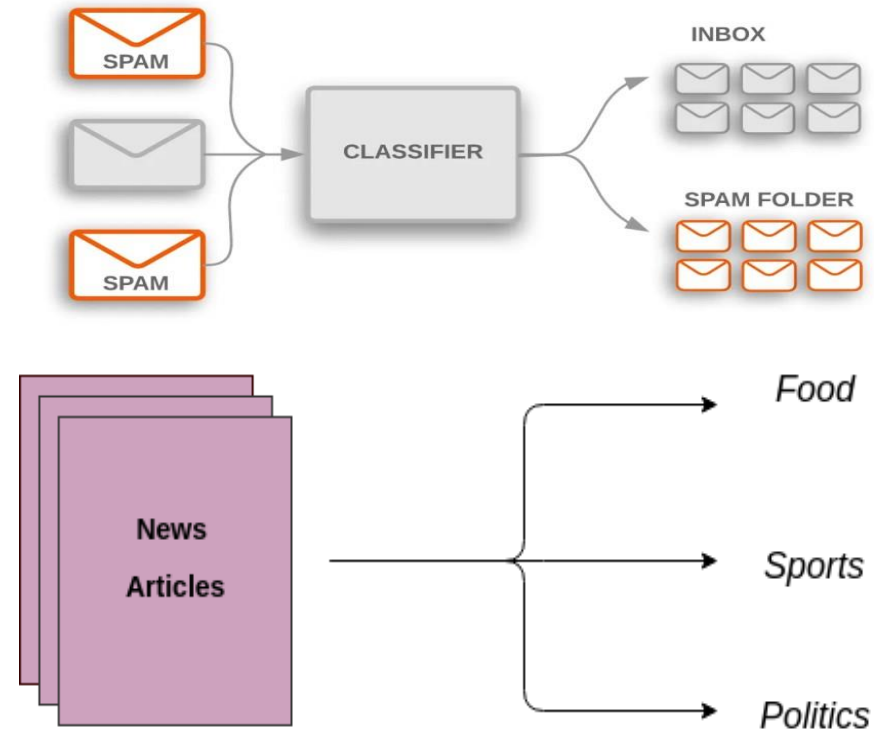
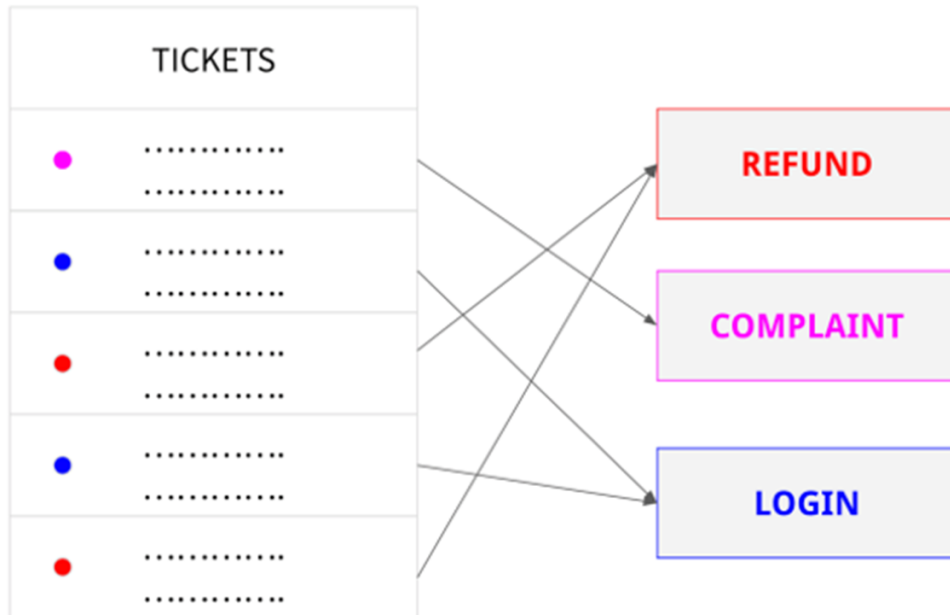
Temas a desarrollar

- Clasificación automática.
- Introducción a métodos de aprendizaje automático.
- Métricas para evaluar un clasificador/ Métricas para evaluar la recuperación.
- Aplicación de librerías para evaluar un sistema de recuperación
- Diseño de experimentos y evaluación de la recuperación.
- La búsqueda de respuestas.

Clasificación automática de Textos

Introducción

- La clasificación de texto es el proceso de clasificar o categorizar los textos sin formato en grupos predefinidos.
- En otras palabras, es el fenómeno de etiquetar los textos no estructurados con sus etiquetas relevantes que se predicen a partir de un conjunto de categorías predefinidas. Por ejemplo, la clasificación de texto se utiliza para filtrar correos electrónicos deseados y no deseados.

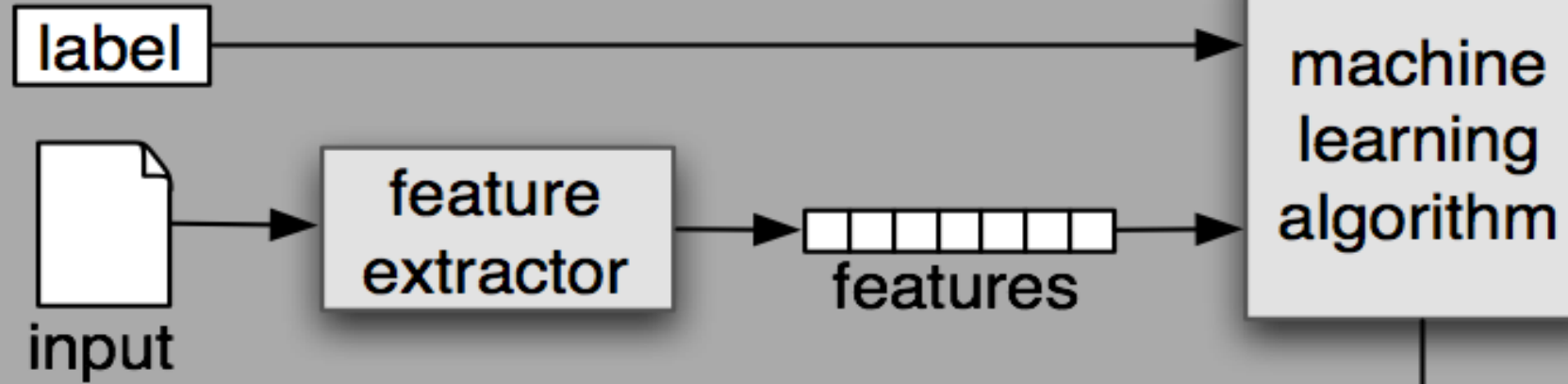


Introducción

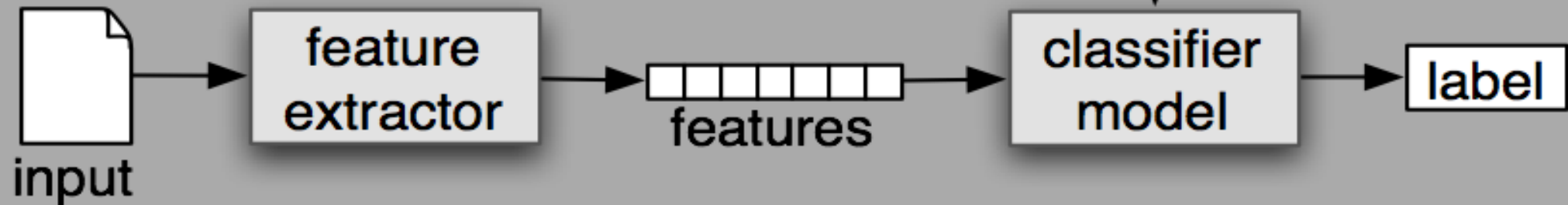
- Hoy en día, la clasificación de texto se utiliza con una amplia gama de servicios digitales para **identificar los sentimientos de los clientes**, **analizar discursos** de líderes políticos y empresarios, **monitorear el odio** y el **acoso** en plataformas de redes sociales, y más.
- De manera similar a un algoritmo de clasificación que se ha entrenado en un conjunto de datos tabulares para predecir una clase, la clasificación de texto también utiliza aprendizaje automático supervisado.
- También puede realizar clasificación de texto sin utilizar el aprendizaje automático supervisado. En lugar de algoritmos, se puede diseñar un sistema manual **basado en reglas** para realizar la tarea de clasificación de texto.

Clasificación mediante ML

(a) Training



(b) Prediction



Introducción

- La clasificación automática puede definirse como la acción ejecutada por un sistema artificial sobre un conjunto de elementos para **ordenarlos en clases o categorías**.
- Si bien los elementos a clasificar pueden ser de cualquier tipo, es la clasificación automática de textos una de las áreas de investigación que ha cobrado mayor importancia en los últimos años debido, en parte, a los grandes volúmenes de textos digitales que se almacenan en bases de datos empresariales, páginas *web* y redes sociales.

Sistemas de clasificación

- La organización automática de objetos en general, basada en las semejanzas entre ellos, puede ayudar a manejar grandes volúmenes de tales objetos, agrupándolos en clases, categorías o clusters de objetos parecidos entre si. Básicamente podemos distinguir dos grandes formas de abordar la organización automática: la ***clasificación no supervisada*** (o *clustering*) y la categorización o ***clasificación supervisada***.
- Los sistemas de **clasificación automática de tipo supervisado** son capaces de identificar la clase o categoría adecuada para un documento determinado, después de una fase de aprendizaje o entrenamiento, durante la cual el sistema aprende las características que definen las diferentes categorías (para ello, los programas de ordenador elaboran modelos o patrones de cada una de las clases o categorías contempladas)

Sistemas de clasificación

- El primer paso para realizar la tarea de Clasificación Automática de Textos consiste en obtener los atributos que describan el texto a clasificar, así como transformarlos a una representación adecuada para ser utilizados por los algoritmos de Aprendizaje Computacional/Automático. A este paso previo se le llama extracción de características.

- ☐ Pre-procesamiento

- ☐ Indexado (Modelo Vectorial)

- ☐ Reducción de dimensionalidad

- ☐ Umbral de frecuencia de documento

Extracción de características

- Pre-procesamiento

- Eliminación de etiquetas/cabeceras HTML, XML...
- Eliminación de palabras vacías
- Lematización/Stemming de palabras

- Indexado (Modelo Vectorial)

- En el modelo vectorial, los documentos son representados por vectores de palabras y una colección de documentos son representados por una matriz A (palabra por documento), donde cada entrada representa las ocurrencias de una palabra en un documento
- Existen muchos caminos para determinar el peso a_{ik} de la palabra i en el documento k

$$A = \begin{pmatrix} a_{11} & \cdots & a_{1m} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nm} \end{pmatrix},$$

Donde a_{ik} es el peso de la palabra i en el documento k .

Indexado, esquemas de ponderación

- Ponderado Booleano

- Este es el esquema más simple y consiste en asignar 1 a a_{ik} si la palabra ocurre en el documento y 0 en otro caso.

- Ponderado por frecuencia de palabra

- Otro esquema simple es usar la frecuencia de la palabra en el documento, $a_{ik} = f_{ik}$
Donde f_{ik} es la frecuencia de la palabra i en el documento k .

- Ponderado TFXIDF

- Los esquemas previos no toman en cuenta la frecuencia de la palabra en todos los documentos en la colección. Una aproximación bien conocida para calcular pesos de palabras es el TFXIDF (Term Frequency x Inverse Document Frequency), el cual asigna el peso a la palabra i en el documento k en proporción al número de ocurrencias de la palabra en el documento, y en proporción inversa al número de documentos en la colección en la que la palabra ocurre al menos una vez

TF-IDF . Ejemplo

- Donde f_{ik} es la frecuencia de la palabra i en el documento k , N el número de documentos en la colección y n_i el número de documentos en los que i aparece.

$$a_{ik} = f_{ik} \times \log\left(\frac{N}{n_i}\right)$$

$$tf(n) = \sum_{D1} (n)$$

La frecuencia de aparición de un término (n) en un documento (D1) es la suma de las ocurrencias de dicho término

$$TF-IDF_{(n,d)} = TF_{(n,d)} \times IDF_{(n)}$$

Peso de un término (n) en un documento (d)

Frecuencia de aparición de un término (n) en un documento (d)

Factor IDF de un término (n)

$$IDF_{(n)} = \log_{10} \frac{N}{DF_{(n)}} + 1$$

También suele utilizarse el logaritmo en base 2, su función es conseguir un coeficiente bajo, fácil de manejar

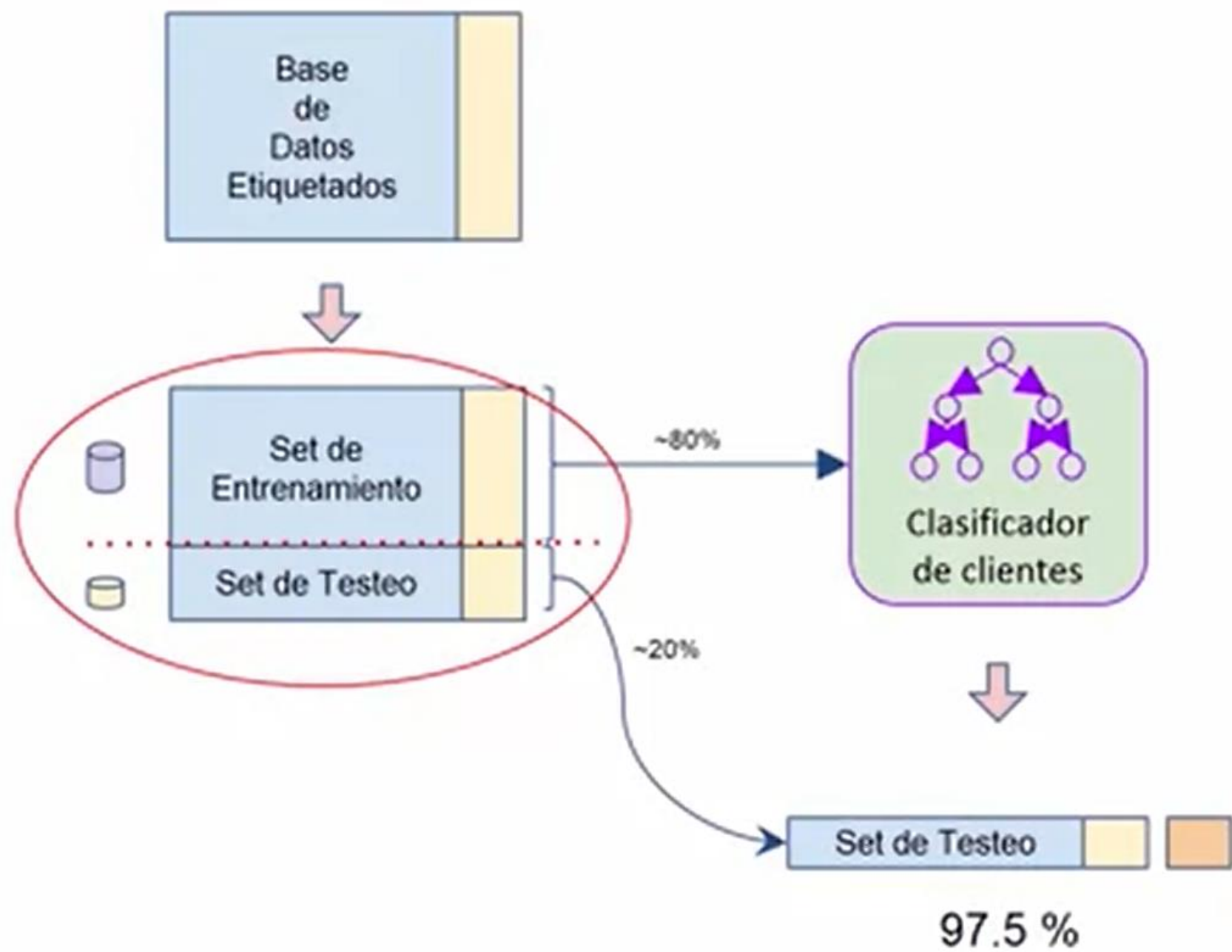
N es el número total de documentos de la colección.

DF (Document Frequency) es el número documentos en los que aparece el término (n) a lo largo de toda la colección

Factor correctivo

Algoritmos de Clasificación

- Naive Bayes
- Support Vector Machines
- K-Nearest Neighbor
- Discriminant Analysis



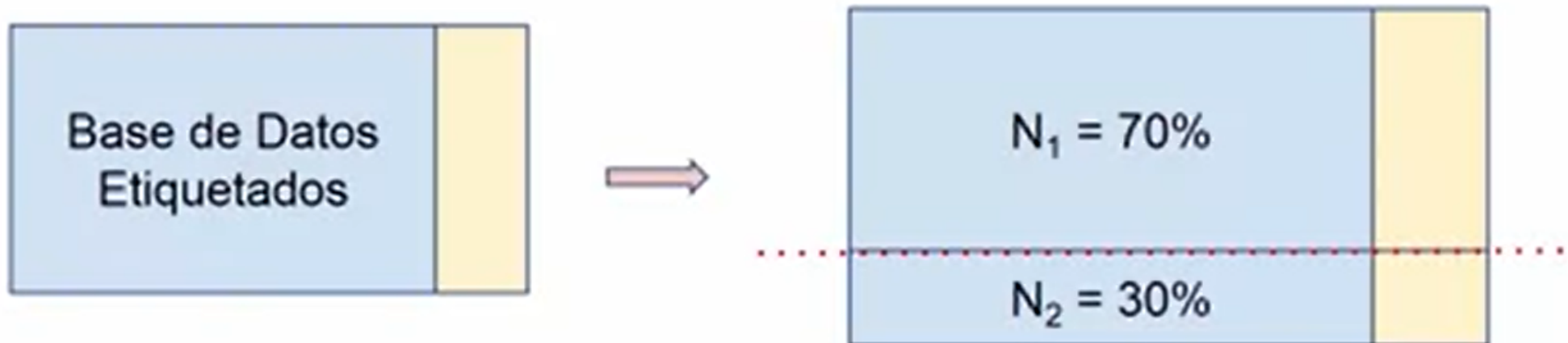
Etapas de entrenamiento y validación

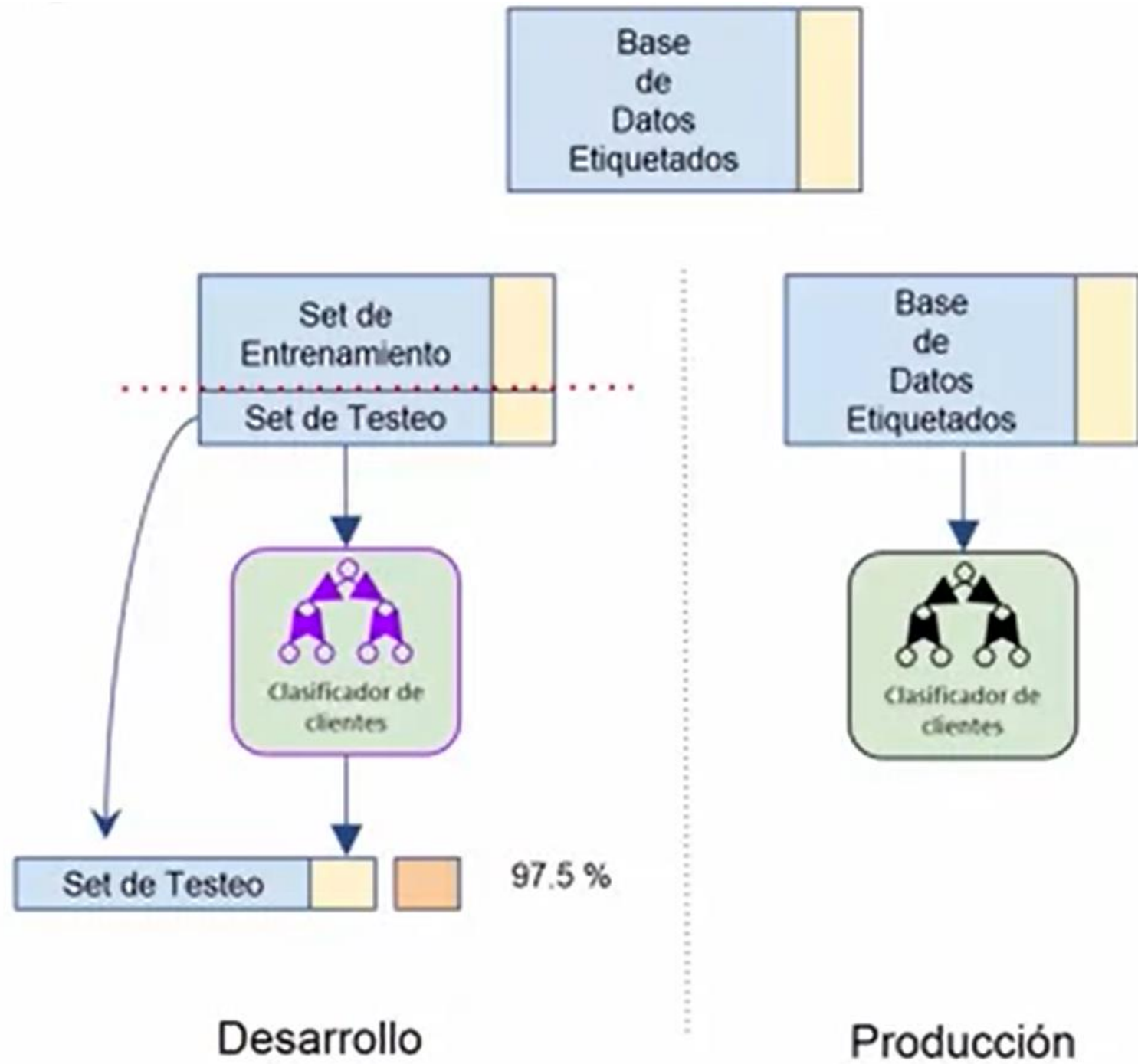
- Etapas de entrenamiento y validación. Para medir la calidad de los modelos predictivos deben definirse apropiadamente las etapas de entrenamiento y validación. De esta manera se asegura que las predicciones sean precisas y robustas.
- ❑ Datos de entrenamiento (training dataset): conjunto de datos que se emplea para construir el modelo.
- ❑ Datos de validación (test dataset): conjunto de datos que se usa para validar el modelo obtenido en la etapa anterior.

Partición del conjunto

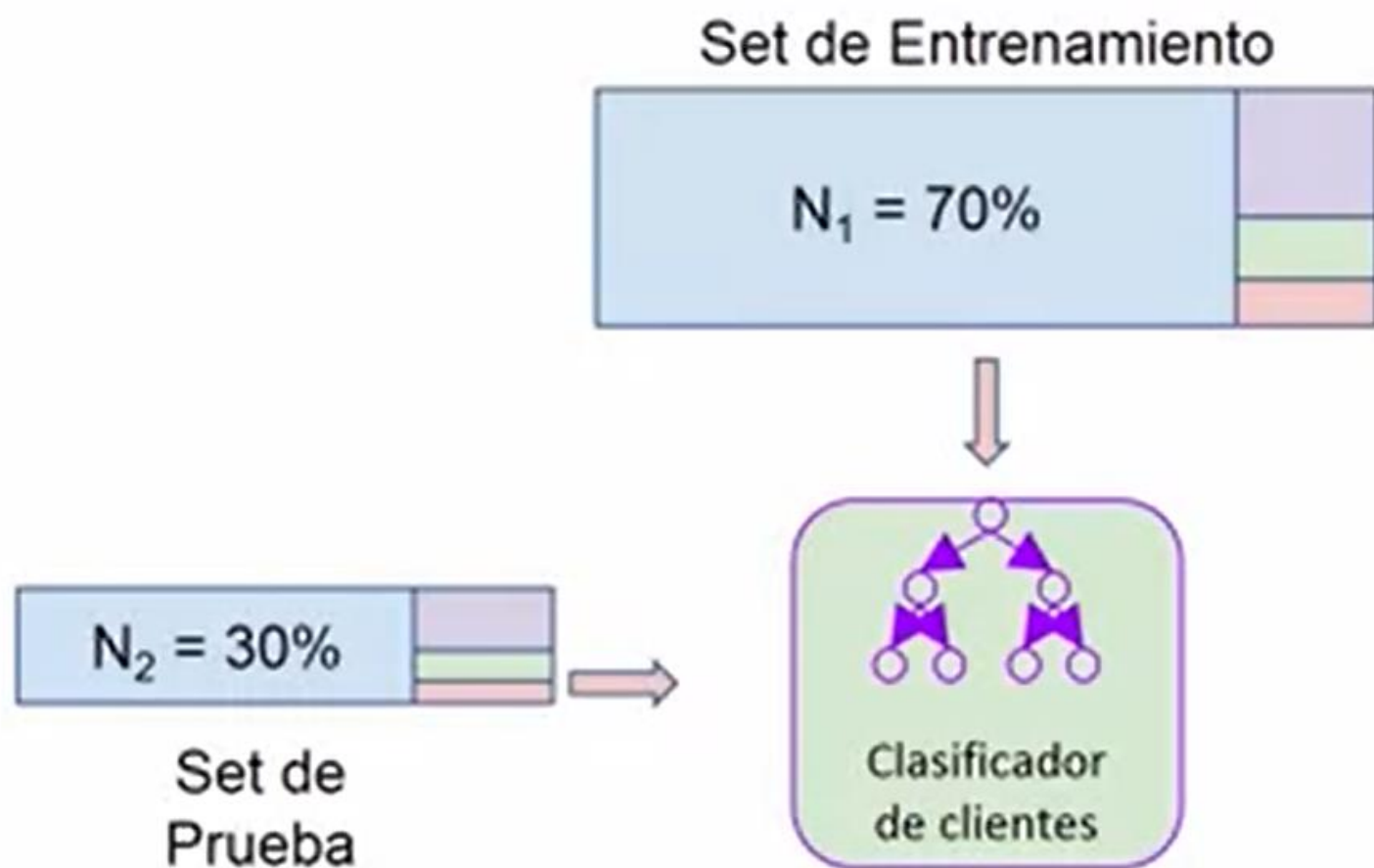
1. Hold Out

- Particionar el conjunto de datos en tamaños N_1 y N_2 (Ej: 70% - 30%)





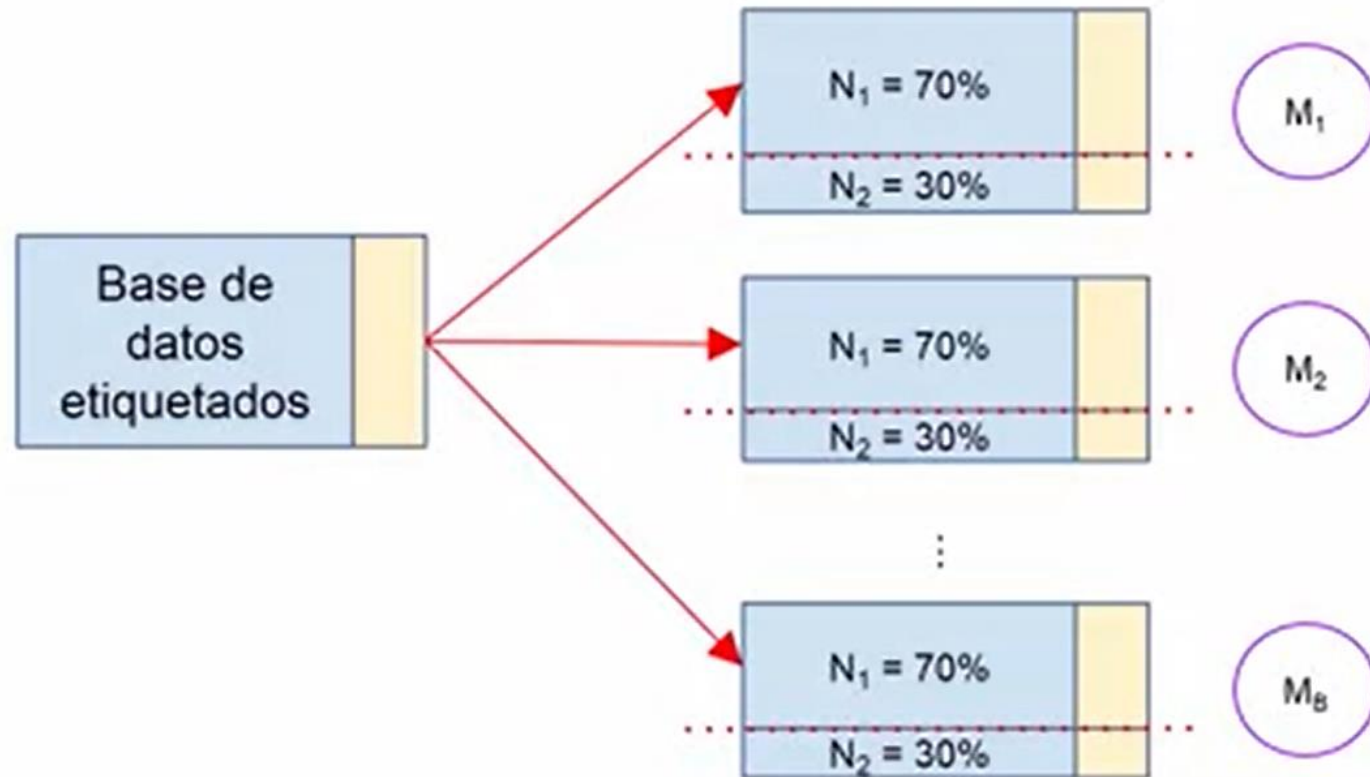
- Mantiene las proporciones de las clases.



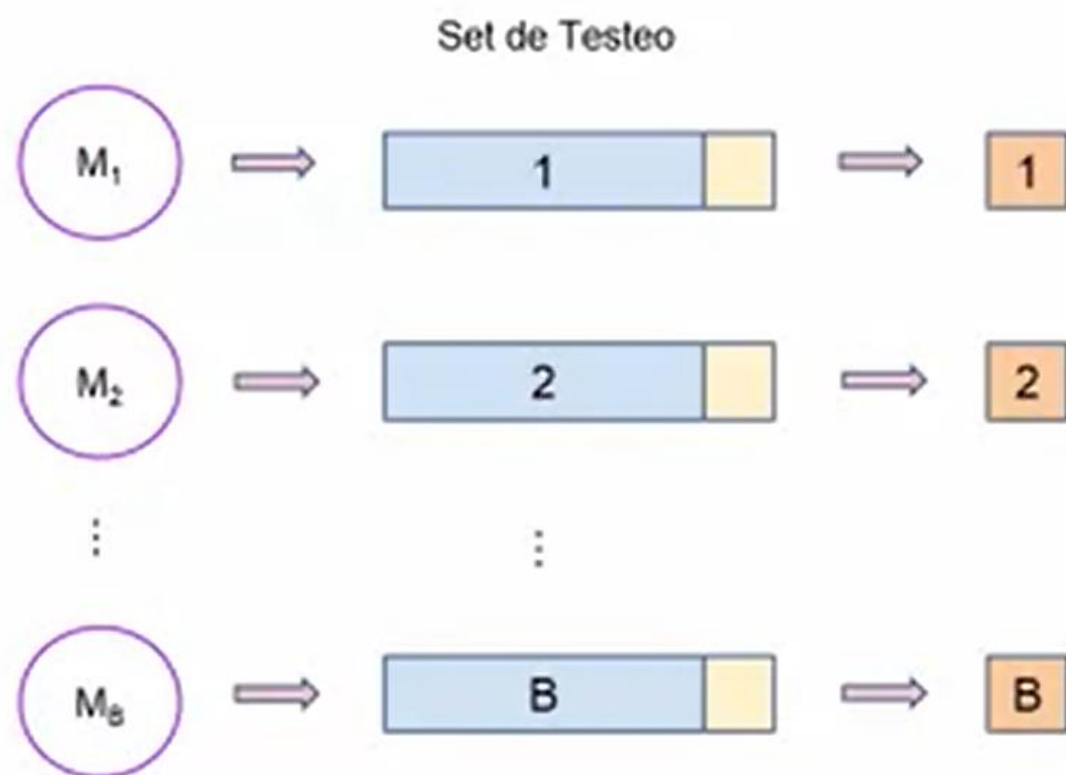
Partición del conjunto

2. Random Sub-Sampling

- Realizar Hold Out B Veces



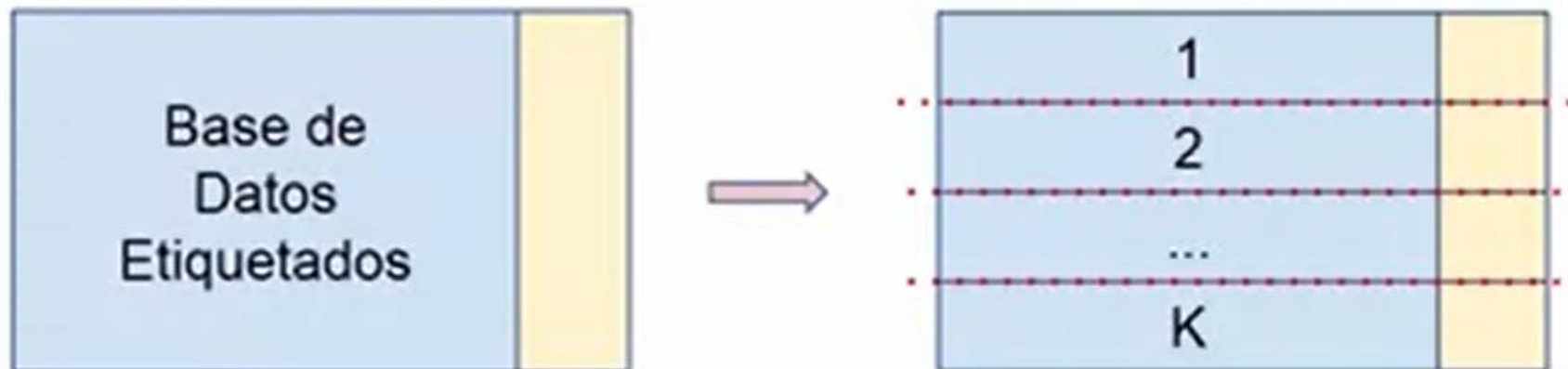
2. Random Sub-Sampling:



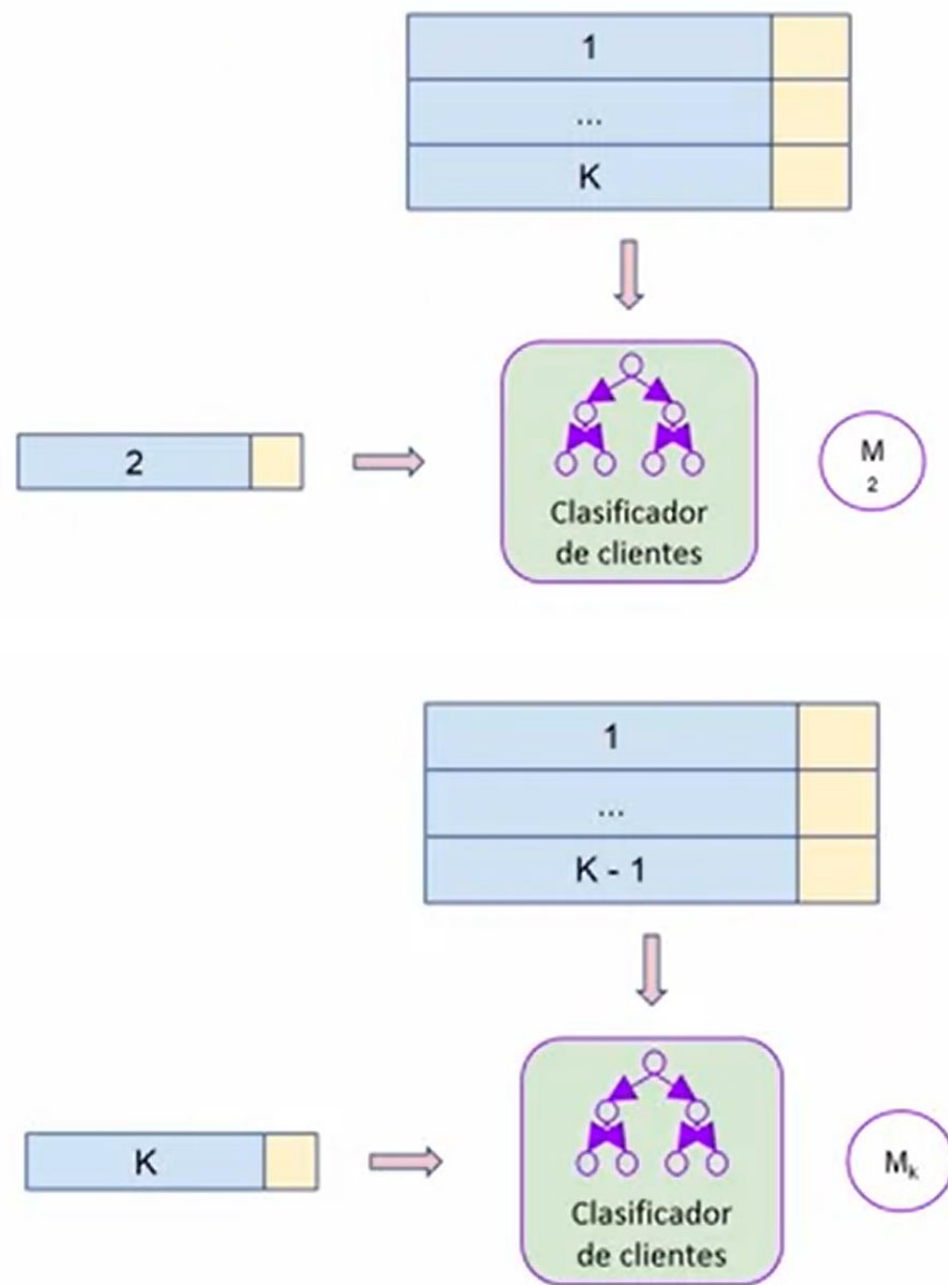
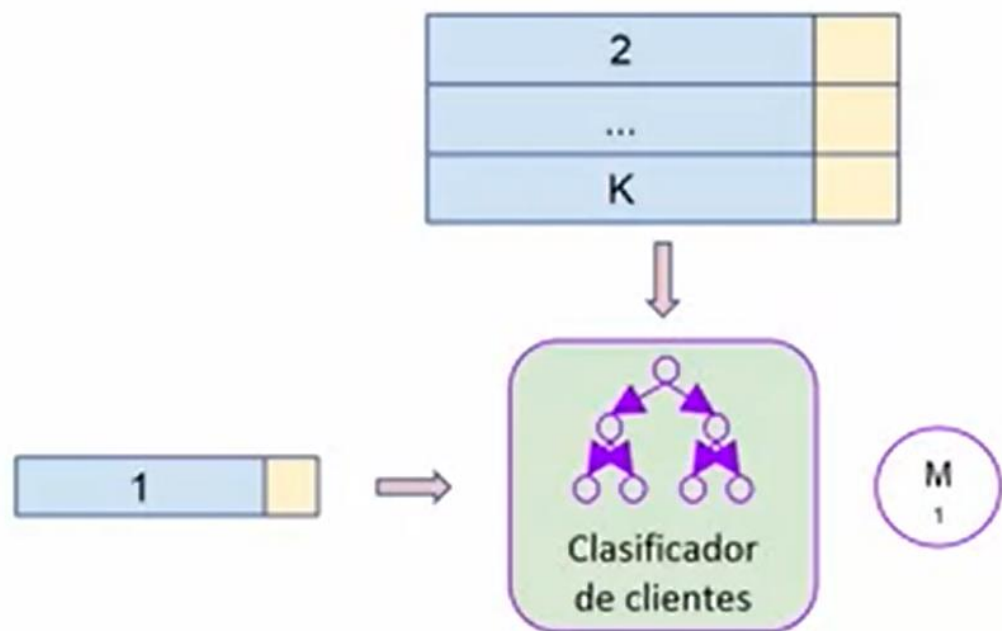
Partición del conjunto

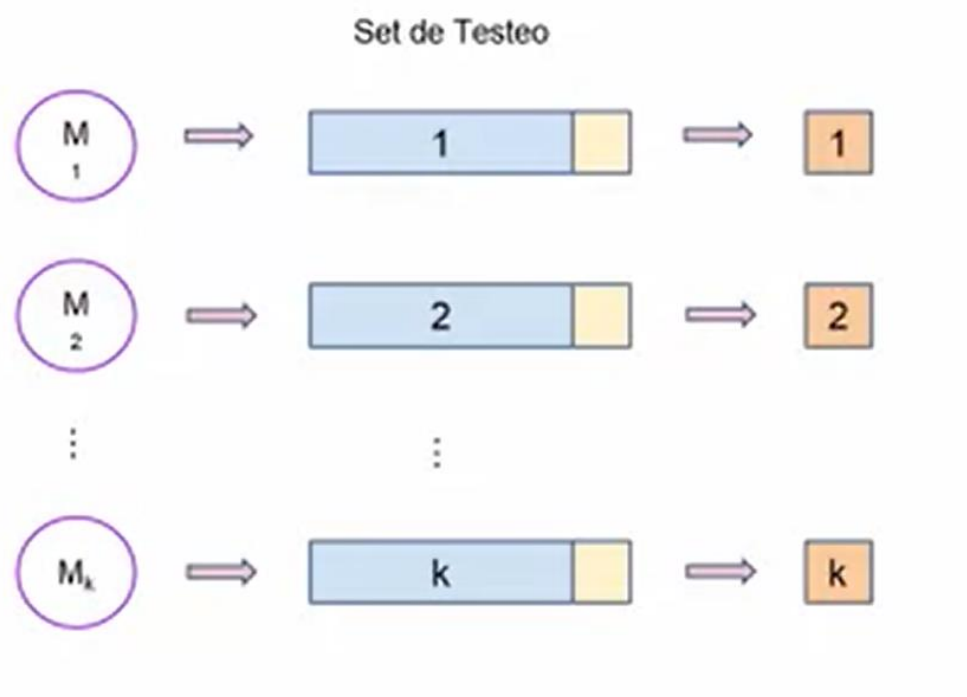
3. K-Fold Cross Validation

- El conjunto de datos se particiona en K particiones (folds) disjuntos de aproximadamente el mismo tamaño

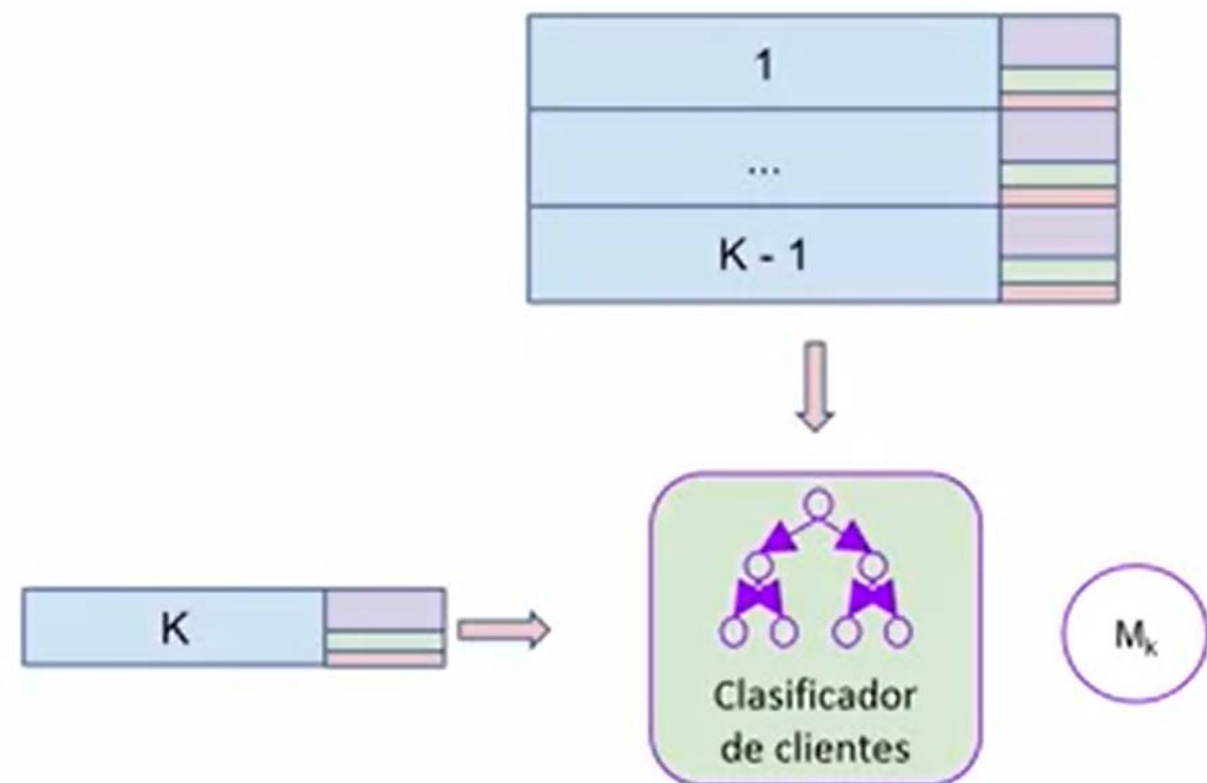


- Con cada uno de los folds se crea un modelo clasificador usando a los $K-1$ restantes.





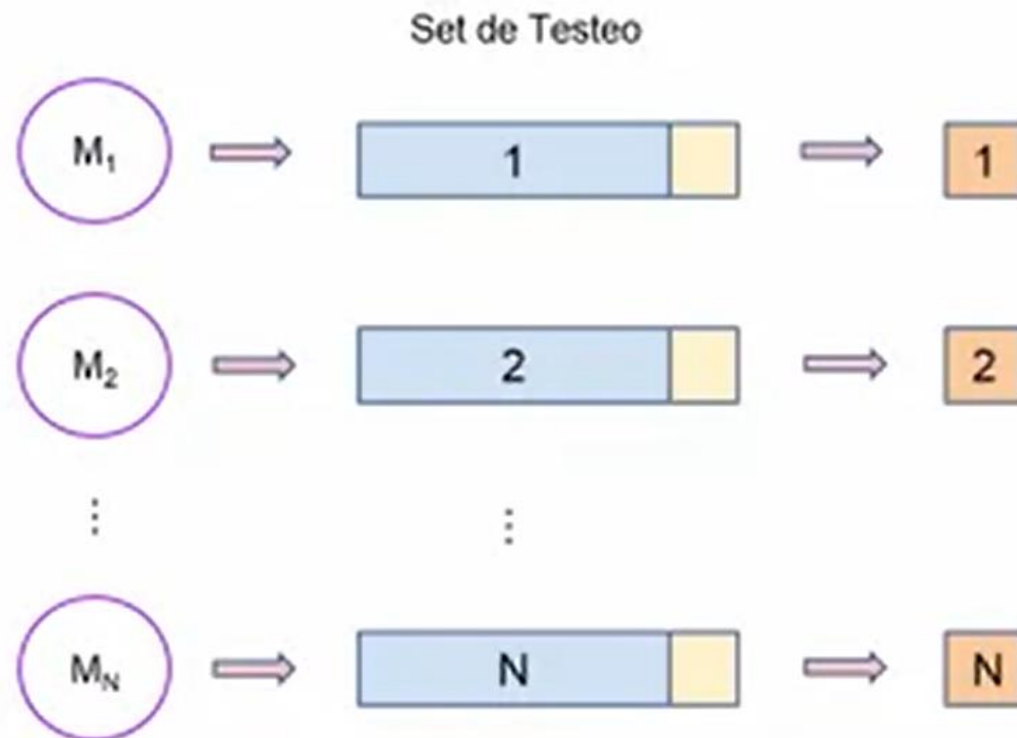
3. Stratified K - Fold Cross Validation



Partición del conjunto

4. Leave-one out

○ Caso particular con $k = N$



Evaluación

- La medida de evaluación empleada depende, entre otros factores, del contexto de aplicación o de la tarea a evaluar.
 - **Clasificación.** Normalmente, se emplea como medida la precisión, que se obtiene como la proporción de instancias clasificadas correctamente.
 - No obstante, pueden considerarse otras medidas o estadísticos como los que se enumeran a continuación.
 - Para la definición de algunos de ellos es importante la matriz de confusión.

Matriz de Confusión

- Supongamos que se dispone de k clases, $C_1, C_2, \dots, C_k$, y sean
 - a_i el número de ejemplos que pertenecen a la clase C_i ,
 - p_j el número de ejemplos que el clasificador clasifica como perteneciente a la clase C_j ,
 - n_{ij} el número de ejemplos de la clase C_i que se clasifican como pertenecientes a la clase C_j , y
 - N el número total de ejemplos.
- La disposición de estos valores en forma de tabla recibe el nombre de **matriz de confusión**.

Matriz de Confusión

		Clase predicha						
		C_1	C_2	$\dots$	C_j	$\dots$	C_k	a_i
Clase real	C_1	n_{11}	n_{12}	$\dots$	n_{1j}	$\dots$	n_{1k}	a_1
	C_2	n_{21}	n_{22}	$\dots$	n_{2j}	$\dots$	n_{2k}	a_2
	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
	C_i	n_{i1}	n_{i2}	$\dots$	n_{ij}	$\dots$	n_{ik}	a_i
	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
	C_k	n_{k1}	n_{k2}	$\dots$	n_{kj}	$\dots$	n_{kk}	a_k
	p_j	p_1	p_2	$\dots$	p_j	$\dots$	p_k	N

Matriz de Confusión

		PREDICCIÓN	
		POSITIVO	NEGATIVO
CLASE REAL	POSITIVO	True Positives (TP)	False Negatives (FN)
	NEGATIVO	False Positives (FP)	True Negatives (TN)

Ejemplo

	C_1	C_2	C_3	C_4	a_i
C_1	130	74	2	6	212
C_2	96	99	6	16	217
C_3	3	4	207	4	218
C_4	6	12	4	177	199
p_j	235	189	219	203	845

Accuracy - Exactitud

- Esta métrica es la base uno. Indica el número de elementos clasificados correctamente en comparación con el número total de artículos.
- No funciona bien con las clases desequilibradas que pueden tener muchos elementos de la misma clase e incluir algunas otras clases.

$$\text{accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

True Positive (TP) rate

- También conocida como **Recall o Sensitivity**, es la proporción de ejemplos que fueron clasificados como pertenecientes a la clase entre todos los ejemplos que verdaderamente pertenecen a esa clase. Es decir, es la **proporción de ejemplos que han sido bien clasificados**.
- Matemáticamente se obtiene desde la siguiente expresión:

$$TP(C_i) = \frac{n_{ij}}{\sum_{j=1}^k n_{ij}} = \frac{n_{ij}}{a_i}$$

$$\text{recall} = \frac{TP}{TP + FN}$$

True Positive (TP) rate. Ejemplo

	C_1	C_2	C_3	C_4	a_i
C_1	130	74	2	6	212
C_2	96	99	6	16	217
C_3	3	4	207	4	218
C_4	6	12	4	177	199
p_j	235	189	219	203	845

$$TP(C_1) = \frac{130}{212} = 0.613$$

$$TP(C_2) = \frac{99}{217} = 0.456$$

$$TP(C_3) = \frac{207}{218} = 0.951$$

$$TP(C_4) = \frac{177}{199} = 0.889$$

False Positive (FP) rate

- Es la proporción de ejemplos que fueron clasificados como pertenecientes a la clase, pero que en realidad pertenecen a otra clase. Es decir, es la **proporción de ejemplos que han sido mal clasificados**.
- Matemáticamente se obtiene desde la siguiente expresión:

$$FP(C_i) = \frac{\left(\sum_{j=1}^k n_{ij}\right) - n_{ii}}{\left(\sum_{j=1}^k a_j\right) - a_i}$$

False Positive (FP) rate. Ejemplo

	C_1	C_2	C_3	C_4	a_i
C_1	130	74	2	6	212
C_2	96	99	6	16	217
C_3	3	4	207	4	218
C_4	6	12	4	177	199
p_j	235	189	219	203	845

$$FP(C_1) = \frac{96 + 3 + 6}{217 + 218 + 199} = 0.165$$

$$FP(C_2) = \frac{74 + 4 + 12}{212 + 218 + 199} = 0.173$$

$$FP(C_3) = \frac{2 + 6 + 4}{212 + 217 + 199} = 0.019$$

$$FP(C_4) = \frac{6 + 16 + 4}{212 + 217 + 218} = 0.040$$

Precision (Precisión)

- Es la proporción de ejemplos que realmente pertenecen a la clase de entre los que fueron clasificados como pertenecientes a la clase.
- Matemáticamente se obtiene desde la siguiente expresión:

$$Precision(C_i) = \frac{n_{ii}}{\sum_{j=1}^k n_{ij}}$$

$$\text{precision} = \frac{TP}{TP + FP}$$

Precision (Precisión). Ejemplo

	C_1	C_2	C_3	C_4	a_i
C_1	130	74	2	6	212
C_2	96	99	6	16	217
C_3	3	4	207	4	218
C_4	6	12	4	177	199
p_j	235	189	219	203	845

$$Precision(C_1) = \frac{130}{235} = 0.553$$

$$Precision(C_2) = \frac{99}{189} = 0.524$$

$$Precision(C_3) = \frac{207}{219} = 0.945$$

$$Precision(C_4) = \frac{177}{203} = 0.872$$

F-measure

- Media armónica entre Precision y Recall..
- Matemáticamente se obtiene desde la siguiente expresión:

$$F - measure(C_i) = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$$

F-measure. Ejemplo

	C_1	C_2	C_3	C_4	a_i
C_1	130	74	2	6	212
C_2	96	99	6	16	217
C_3	3	4	207	4	218
C_4	6	12	4	177	199
p_j	235	189	219	203	845

$$F(C_1) = 2 \cdot \frac{0.553 \cdot 0.613}{0.553 + 0.613} = 0.582$$

$$F(C_2) = 2 \cdot \frac{0.524 \cdot 0.456}{0.524 + 0.456} = 0.488$$

$$F(C_3) = 2 \cdot \frac{0.945 \cdot 0.950}{0.945 + 0.950} = 0.947$$

$$F(C_4) = 2 \cdot \frac{0.872 \cdot 0.889}{0.872 + 0.889} = 0.881$$

Otros estadísticos

- *Índice kappa*
- *Área bajo la curva ROC*
- *Mean absolute error (MAE)*
- *Root mean squared error (RMSE)*
- *Root relative squared error (RRSE)*

Evaluar un sistema de recuperación

Criterios de evaluación

- Después de plantear una búsqueda, lo primero que hace el documentalista es comprobar, a partir de los resultados, si la información que ha recuperado cumple las condiciones siguientes:
 - ☐ Se corresponde con la estrategia de búsqueda que ha planteado.
 - ☐ Sirve para solucionar la demanda de información del usuario, tanto por su calidad como por su cantidad.
- En definitiva, el documentalista intenta determinar la relevancia de los resultados obtenidos.

La relevancia desde el punto de vista del sistema de recuperación de información

- Cuando consultamos una base de datos o un buscador de Internet, obtenemos unos resultados en respuesta a una estrategia de búsqueda que hemos ejecutado previamente.
- En este apartado podemos hacernos dos preguntas básicas:
 - ❑ ¿Hasta qué punto hemos conseguido recuperar todos los documentos que existen en el sistema y que son relevantes para nuestra búsqueda?
 - ❑ ¿Hasta qué punto hemos conseguido recuperar la máxima cantidad de documentos relevantes, sin recuperar los que no lo son?

Relevancia

- Encontramos la respuesta a estas dos preguntas en dos índices básicos denominados:
- Índice de recuperación (o recall ratio).
- Índice de precisión (o precision ratio).

El índice de recuperación - *recall*

- El índice de recuperación mide la proporción de documentos relevantes recuperados, es decir, mide la efectividad y exhaustividad de la búsqueda realizada, o hasta qué punto después de interrogar el sistema hemos conseguido recuperar todos los documentos relevantes para nuestra consulta.
- Este índice se expresa en términos de proporcionalidad mediante la fórmula siguiente:

$$\text{Índice de recuperación} = \frac{\text{Número de ítems relevantes recuperados} \cdot 100}{\text{Número de ítems relevantes existentes en el sistema}}$$

El índice de precisión - *precision*

- El índice de precisión mide la exactitud de la búsqueda, o hasta qué punto se ha conseguido recuperar sólo los registros que debían ser recuperados, evitando hacerlo con los que no eran necesarios.
- Este índice se expresa en términos de proporcionalidad mediante la fórmula siguiente:

$$\text{Índice de precisión} = \frac{\text{Número de ítems relevantes recuperados} \cdot 100}{\text{Número de ítems totales recuperados}}$$

Interpretación

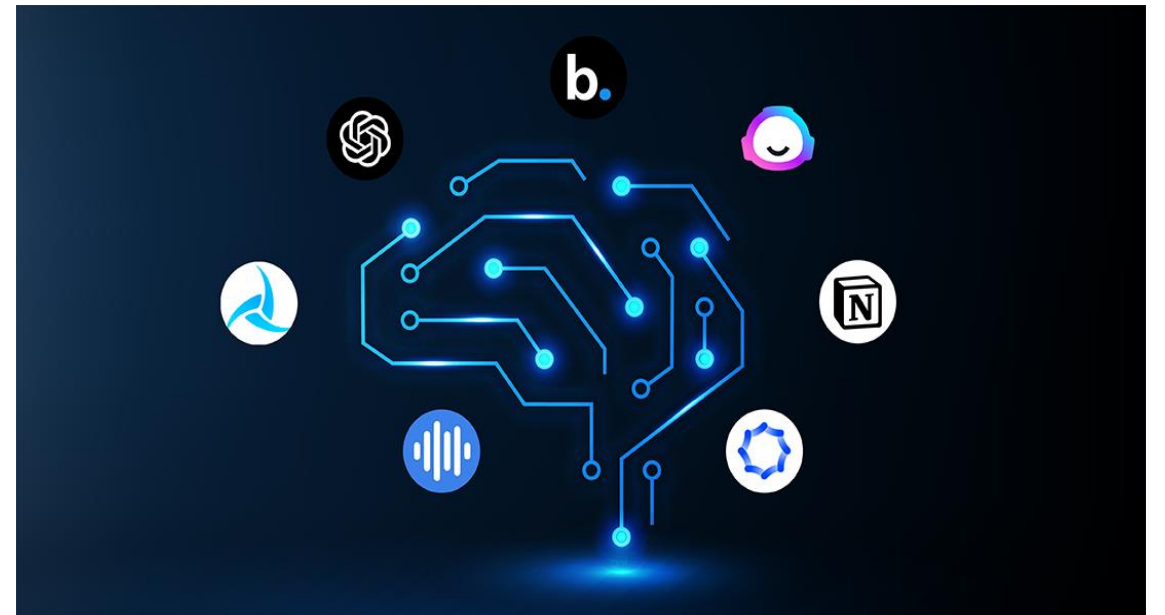
- El índice de recuperación se relaciona con el concepto silencio, mientras que el índice de precisión se relaciona con el concepto ruido.
- Hablamos de *silencio* en una búsqueda cuando no hemos conseguido recuperar todos los documentos pertinentes; o lo que es lo mismo, cuando hemos obtenido menos ítems de los que podíamos obtener.
- Hablamos de *ruido* en una búsqueda cuando no hemos conseguido eliminar los documentos no pertinentes (cuando hemos obtenido más ítems de los que deberíamos haber obtenido).

Interpretación

	Ítems recuperados	Ítems no recuperados
Relevantes	A	C
No relevantes	B	D

- **Grupo A:** engloba los ítems recuperados y relevantes para nuestra consulta.
- **Grupo B:** engloba los ítems recuperados y no relevantes para nuestra consulta.
- **Grupo C:** engloba los ítems que, a pesar de ser relevantes para nuestra consulta, no han sido recuperados.
- **Grupo D:** engloba los ítems que no han sido recuperados, pero que tampoco eran pertinentes para nuestra consulta.

Generación de Textos (Chatbots – Question Answering)



Introducción

- La **búsqueda de respuestas**, en inglés *Question Answering (QA)*, es un tipo de [recuperación de información](#). Dada una cierta cantidad de documentos (tales como [World Wide Web](#)), el sistema debería ser capaz de recuperar respuestas a preguntas planteadas en [lenguaje natural](#). QA es observado como un método que requiere una tecnología de [Procesamiento de lenguaje natural](#) más compleja que otros tipos de sistemas para la [Recuperación de documento](#), y, en algunos casos, se le observa como un paso por delante de la tecnología del [buscador](#).
- Un sistema de *question answering* es uno de los sistemas más complejos en torno a la recuperación de [información](#). Debemos de tener en cuenta que un sistema basado en el question-answering es mucho más difícil que un sistema normal que se encarga de buscar una información en una cantidad más o menos grande de [documentos](#), ya que estos debe extraer de dichos documentos un fragmento de [texto](#) (mínimo) que responda a una pregunta dada en lenguaje natural. Estos sistemas están muy ligados a los [buscadores web](#).

Introducción

- Por su parte, los sistemas QA tienen como objetivo encontrar de forma automatizada respuestas precisas a peticiones de información realizadas por los usuarios. No devuelven el documento completo, sino que pueden extraer el párrafo concreto que contiene la respuesta exacta.
- La RI obtiene documentos completos ordenados por relevancia, sin ningún tipo de tratamiento o análisis de los mismos, por lo que la respuesta puede no estar contenida en el texto. En cambio, un sistema de QA requiere un procesamiento del lenguaje más complejo, ya que debe extraer sólo el fragmento mínimo que responda a la pregunta realizada por el usuario. Es decir, un buen sistema de QA es capaz de "comprender" la pregunta para poder encontrar la respuesta concreta y específica. Como resultado, el sistema de QA extrae y construye respuestas concisas y exactas para la consulta realizada en lenguaje natural.

Introducción

Los grandes modelos de IA de imágenes y lenguaje, a veces llamados IA generativa o modelos básicos, han creado un nuevo conjunto de oportunidades para empresas y profesionales que realizan la creación de contenido.

- **Generación de contenido automatizada:** se pueden utilizar modelos de IA de imágenes y lenguaje de gran tamaño para generar contenido automáticamente, como artículos, publicaciones de blogs o publicaciones en redes sociales. Esta puede ser una valiosa herramienta para ahorrar tiempo a empresas y profesionales que crean contenido de forma regular.
- **Calidad de contenido mejorada:** el contenido generado por IA puede ser de mayor calidad que el contenido creado por humanos, debido a que los modelos de IA pueden aprender de una gran cantidad de datos e identificar patrones que los humanos tal vez no puedan ver. Esto puede resultar en contenido más preciso e informativo.
- **Mayor variedad de contenido:** los modelos de IA pueden generar una variedad de tipos de contenido, incluidos texto, imágenes y videos. Esto puede ayudar a las empresas y profesionales a crear contenido más diverso e interesante que atraiga a una gama más amplia de personas.
- **Contenido personalizado:** los modelos de IA pueden generar contenido personalizado según las preferencias de los usuarios individuales. Esto puede ayudar a las empresas y profesionales a crear contenido que tenga más probabilidades de ser de interés para su público objetivo y, por lo tanto, más probable que sea leído o compartido.

Introducción: Generative AI

La IA generativa puede hacer mucho.

- Es capaz de producir texto e imágenes, que abarcan publicaciones de blogs, códigos de programas, poesía y obras de arte (e incluso ganar concursos, **de manera controvertida**).
- El software utiliza modelos complejos de aprendizaje automático para predecir la siguiente palabra basándose en secuencias de palabras anteriores, o la siguiente imagen basándose en palabras que describen imágenes anteriores

ChatBot

- Un chatbot es un programa informático diseñado para simular conversaciones con usuarios humanos, especialmente a través de internet. Estos programas utilizan tecnologías de inteligencia artificial y procesamiento de lenguaje natural para comprender las preguntas o solicitudes de los usuarios y proporcionar respuestas automáticas que imitan la manera en que un humano respondería. Los chatbots pueden ser tan simples como programas basados en reglas que siguen un script muy estructurado, o tan complejos como los que utilizan algoritmos avanzados de aprendizaje automático y redes neuronales para aprender de las interacciones y mejorar sus respuestas con el tiempo.

Uso de ChatBots

Los chatbots se utilizan en una amplia variedad de aplicaciones, tales como:

- **Servicio al cliente:** Proporcionan asistencia inmediata a las preguntas de los clientes, pueden resolver problemas comunes, ayudar en la navegación de sitios web, y ofrecer información detallada sobre productos o servicios.
- **E-commerce:** Ayudan a los usuarios a encontrar productos, hacer recomendaciones personalizadas, procesar pedidos y gestionar devoluciones.
- **Educación:** Pueden funcionar como tutores virtuales o asistentes de aprendizaje, ofreciendo explicaciones, guiando estudios, y respondiendo preguntas frecuentes.
- **Entretenimiento:** Algunos chatbots ofrecen juegos interactivos, narran historias o actúan como compañeros conversacionales para entretener a los usuarios.
- **Salud:** Proporcionan soporte en la gestión de la salud, como recordatorios de medicación, consejos de salud, y en algunos casos, primeros auxilios psicológicos.

Tecnología usada en los ChatBots

Para funcionar efectivamente, los chatbots pueden utilizar varias tecnologías y métodos, incluyendo:

- **Procesamiento de lenguaje natural (NLP):** Permite al chatbot entender el lenguaje humano.
- **Modelos de respuesta basados en reglas:** El chatbot responde a entradas específicas utilizando un conjunto predefinido de instrucciones.
- **Aprendizaje automático y modelos de aprendizaje profundo:** Facilitan al chatbot la capacidad de aprender de las interacciones pasadas y mejorar con el tiempo.

Los chatbots han revolucionado la forma en que las empresas interactúan con sus clientes y continúan evolucionando rápidamente, incorporando nuevas tecnologías para hacer las conversaciones más naturales y efectivas.

ChatGPT

- ChatGPT es una aplicación de chatbot de inteligencia artificial desarrollado en 2022 por OpenAI que se especializa en el diálogo. El chatbot es un modelo de lenguaje ajustado con técnicas de aprendizaje tanto supervisadas como de refuerzo. Está compuesto por los modelos GPT-4 y GPT-3.5 de OpenAI.
- Su denominación es el acrónimo de Chat Generative Pre-trained Transformer (Chat Generativo Preentrenado Transformador).
- Chat GPT se basa en un modelo de entrenamiento único: el Aprendizaje por Refuerzo a partir de la Retroalimentación Humana (RLHF). Este método entrena al asistente haciendo que los humanos simulen conversaciones artificiales con él, y luego adapta sus respuestas en función de la precisión con la que reflejan el diálogo humano natural.



You

Necesito un script de Python empleando NLTK para tokenizar un texto

< 2/2 >



ChatGPT

Por supuesto, estaré encantado de ayudarte a crear un script en Python para tokenizar un texto usando la librería NLTK (Natural Language Toolkit), que es una de las librerías más utilizadas en procesamiento de lenguaje natural (NLP). Antes de comenzar, asegúrate de tener instalado Python y la librería NLTK. Si no la tienes instalada, puedes hacerlo ejecutando `pip install nltk` desde tu terminal o prompt de comandos.

Aquí tienes un script básico que tokeniza un texto en palabras y oraciones:

```
python  Copy code

import nltk
from nltk.tokenize import word_tokenize, sent_tokenize

# Descarga de los recursos necesarios de NLTK
nltk.download('punkt')

# Texto de ejemplo para tokenizar
texto = "Hola, ¿cómo estás? Este es un texto de ejemplo para tokenizar. ¡NLTK es"

# Tokenización por oraciones
oraciones = sent_tokenize(texto, language='spanish')
print("Oraciones tokenizadas:")
print(oraciones)

# Tokenización por palabras
palabras = word_tokenize(texto, language='spanish')
print("\nPalabras tokenizadas:")
```

```
print(oraciones)
print(palabras)
```

Explicación del código:

1. **Importaciones:** Importamos las funciones `word_tokenize` y `sent_tokenize` de NLTK que nos permiten tokenizar un texto en palabras y oraciones, respectivamente.
2. **Descarga de recursos:** Utilizamos `nltk.download('punkt')` para descargar los modelos necesarios para la tokenización. `punkt` es un tokenizador de oraciones pre-entrenado.
3. **Texto de ejemplo:** Definimos una variable `texto` que contiene el texto que queremos tokenizar.
4. **Tokenización por oraciones:** Usamos `sent_tokenize` para dividir el texto en oraciones. El parámetro `language='spanish'` asegura que la tokenización se ajuste a las características del español.
5. **Tokenización por palabras:** Con `word_tokenize`, dividimos el texto en palabras, utilizando también el parámetro `language='spanish'`.

Ejecución del script:

Cuando ejecutes este script, verás en la consola las oraciones y las palabras del texto de ejemplo tokenizadas. Esto es muy útil para procesos posteriores de análisis de texto como la lematización, el análisis sintáctico, la extracción de entidades, entre otros.

