

Procesamiento de Lenguaje Natural

Apunte de Clases (Borrador)

Felipe Bravo Márquez

Felipe Bravo Márquez

Ilustración Portada por Paulette Filla

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN, UNIVERSIDAD DE CHILE

GITHUB.COM/DCCUCHILE/CC6205

Apuntes de clases del curso de Procesamiento de Lenguaje Natural de la Universidad de Chile.

El formato del apunte fue tomado del template de Jasmine Hao.

Borrador, 3 de noviembre de 2023



Índice general

0.1	Interpolación Lineal	3
0.1.1	Estimación de los Valores λ	4
0.2	Modelos de Descuento (Katz Back-Off)	4
0.3	Historia	7
0.4	Conclusiones	7

La clasificación, que implica la asignación de un objeto a una categoría específica, desempeña un papel fundamental tanto en la inteligencia humana como en la artificial. En este contexto, la clasificación abarca diversas tareas, que van desde determinar qué letra, palabra o imagen se ha presentado a nuestros sentidos, hasta reconocer caras, voces, clasificar correos electrónicos o calificar tareas.

El propósito subyacente de la clasificación es tomar una única observación, identificar y extraer características relevantes de la misma, y, en última instancia, ubicarla en una de las categorías discretas predefinidas. En el procesamiento del lenguaje, la mayoría de las tareas de clasificación se abordan mediante enfoques de aprendizaje automático supervisado.

Este capítulo se basa en el material del curso de Daniel Jurafsky, al que se puede acceder a través del siguiente enlace¹.

Ejemplo 1: Clasificación de spam

Ejemplo 2: ¿Quién escribió los documentos Federalist?

- 1787-8: Ensayos anónimos intentaron convencer a Nueva York de ratificar la Constitución de EE. UU.: Jay, Madison, Hamilton.
- La autoría de 12 de las cartas está en disputa.
- 1963: Resuelto por Mosteller y Wallace mediante métodos bayesianos.

Ejemplo 3: ¿Cuál es el tema de este artículo médico?

¹<https://web.stanford.edu/~jurafsky/slp3/4.pdf>

Subject: **Important notice!**
From: Stanford University <newsforum@stanford.edu>
Date: October 28, 2011 12:34:16 PM PDT
To: undisclosed-recipients:;

Greats News!

You can now access the latest news by using the link below to login to Stanford University News Forum.

<http://www.123contactform.com/contact-form-StanfordNew1-236335.html>

Click on the above link to login for more information about this new exciting forum. You can also copy the above link to your browser bar and login for more information about the new services.

© Stanford University. All Rights Reserved.

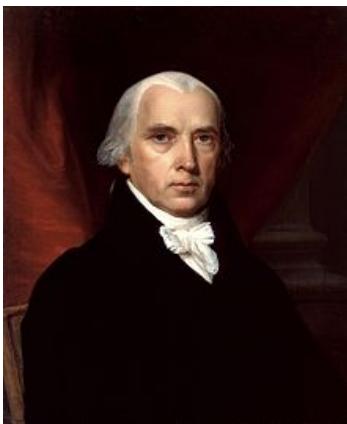


Figura 1: James Madison

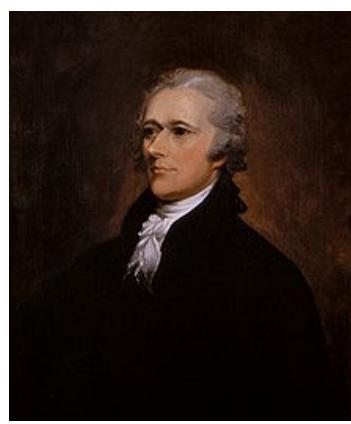


Figura 2: Alexander Hamilton

Ejemplo 4: ¿Reseña de película positiva o negativa?

- + ...personajes extravagantes y sátira **rico** aplicada, y algunos **grandes** giros de la trama.
- - Fue **patético**. La peor parte fue las escenas de boxeo...
- + ...salsa de caramelo **increíble** y almendras dulces y tostadas. ¡Me **encanta** este lugar!
- - pizza **horrible** y **ridículamente** cara...

¿Por qué el análisis de sentimientos?

- Película: ¿Esta reseña es positiva o negativa?
- Productos: ¿Qué opinan las personas sobre el nuevo iPhone?
- Sentimiento público: ¿Cómo está la confianza del consumidor?
- Política: ¿Qué opinan las personas sobre este candidato o tema?
- Predicción: Predecir resultados electorales o tendencias del mercado a partir del sentimiento.

Clasificación básica de sentimientos

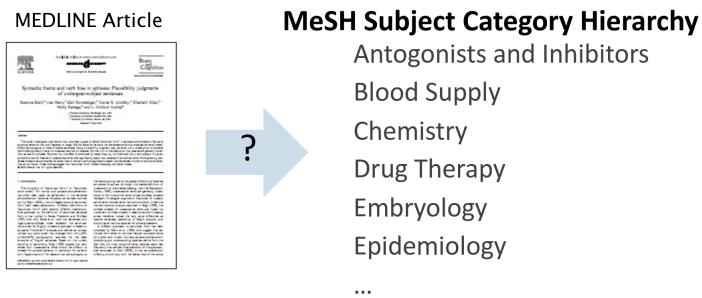
El análisis de sentimientos es la detección de actitudes.

- Tarea simple en la que nos enfocamos en esta clase.
 - ¿Es la actitud de este texto positiva o negativa?

Resumen: Clasificación de texto

La clasificación de texto se puede aplicar a varias tareas, incluyendo:

- Análisis de sentimientos



- Detección de spam
- Identificación de autoría
- Identificación de idioma
- Asignación de categorías, temas o géneros
- ...

0.1 Clasificación de texto: Definición

Entrada:

- Un documento d
- Un conjunto fijo de clases $C = \{c_1, c_2, \dots, c_J\}$

Salida: Una clase predicha $c \in C$

0.1.1 Métodos de clasificación: Reglas codificadas a mano

Reglas basadas en combinaciones de palabras u otras características.

- Spam: dirección-en-lista-negra O (“dólares” Y “has sido seleccionado”)
- La precisión puede ser alta si las reglas se refinan cuidadosamente por expertos
- Pero construir y mantener estas reglas es costoso

0.1.2 Métodos de clasificación: Aprendizaje automático supervisado

Entrada:

- Un documento d
- Un conjunto fijo de clases $C = \{c_1, c_2, \dots, c_J\}$
- Un conjunto de entrenamiento de m documentos etiquetados manualmente: $(d_1, c_1), (d_2, c_2), \dots, (d_m, c_m)$

Salida:

- Un clasificador aprendido $\gamma: d \rightarrow c$

Cualquier tipo de clasificador se puede utilizar:

- Naïve Bayes
- Regresión logística
- Redes neuronales
- k-vecinos más cercanos

0.1.3 Problemas de aprendizaje supervisado

- Tenemos ejemplos de entrenamiento $x^{(i)}, y^{(i)}$ para $i = 1, \dots, m$. Cada $x^{(i)}$ es una entrada, cada $y^{(i)}$ es una etiqueta.

- La tarea es aprender una función f que asigna las entradas x a las etiquetas $f(x)$.
- Modelos condicionales:
 - Aprender una distribución $p(y|x)$ a partir de ejemplos de entrenamiento.
 - Para cualquier entrada de prueba x , definir $f(x) = \arg \max_y p(y|x)$.

0.1.4 Modelos generativos

- Dados ejemplos de entrenamiento $x^{(i)}, y^{(i)}$ para $i = 1, \dots, m$, la tarea es aprender una función f que asigna las entradas x a las etiquetas $f(x)$.
- Modelos generativos:
 - Aprender la distribución conjunta $p(x,y)$ a partir de los ejemplos de entrenamiento.
 - A menudo, tenemos $p(x,y) = p(y)p(x|y)$.
 - Nota: Luego tenemos

$$p(y|x) = \frac{p(y)p(x|y)}{p(x)} \quad \text{donde} \quad p(x) = \sum_y p(y)p(x|y).$$

0.1.5 Clasificación con Modelos Generativos

- Dados ejemplos de entrenamiento $x^{(i)}, y^{(i)}$ para $i = 1, \dots, m$. La tarea consiste en aprender una función f que mapee las entradas x a las etiquetas $f(x)$.
- Modelos generativos:
 - Aprenden la distribución conjunta $p(x,y)$ a partir de los ejemplos de entrenamiento.
 - A menudo, tenemos $p(x,y) = p(y)p(x|y)$.
- La salida del modelo es:

$$\begin{aligned} f(x) &= \arg \max_y p(y|x) = \arg \max_y \frac{p(y)p(x|y)}{p(x)} \\ &= \arg \max_y p(y)p(x|y) \end{aligned}$$

0.2 Intuición del Bayes Ingenuo

El Bayes Ingenuo es un método de clasificación simple ("ingenuo") basado en la regla de Bayes.

- Se basa en una representación muy simple de un documento: *Bolsa de palabras*.

0.2.1 Aplicación de la Regla de Bayes a Documentos y Clases

Para un documento d y una clase c :

$$P(c|d) = \frac{P(d|c)P(c)}{P(d)}$$

0.3 Clasificador Bayes Ingenuo

- MAP significa "máximo a posteriori", que representa la clase más probable:

$$c_{\text{MAP}} = \arg \max_{c \in C} P(c|d)$$

- Para calcular la clase más probable, aplicamos la regla de Bayes:

$$= \arg \max_{c \in C} \frac{P(d|c)P(c)}{P(d)}$$

- Finalmente, podemos eliminar el denominador ya que permanece constante para todas las clases:

$$= \arg \max_{c \in C} P(d|c)P(c)$$

- Para clasificar el documento d , usamos la estimación MAP:

$$c_{\text{MAP}} = \arg \max_{c \in C} P(d|c)P(c)$$

- El documento d se representa como un conjunto de características x_1, x_2, \dots, x_n .
- El clasificador calcula la probabilidad condicional de las características dada una clase y la probabilidad a priori de la clase:

$$= \arg \max_{c \in C} P(x_1, x_2, \dots, x_n | c)P(c)$$

- El término $P(x_1, x_2, \dots, x_n | c)$ representa la "verosimilitud" de las características dada la clase.
- El término $P(c)$ representa la probabilidad ".a priori" de la clase.
- El clasificador Bayes Ingenuo [?] calcula la estimación MAP considerando las probabilidades de verosimilitud y a priori:

$$c_{\text{MAP}} = \arg \max_{c \in C} P(x_1, x_2, \dots, x_n | c)P(c)$$

- La probabilidad de las características dada la clase, $P(x_1, x_2, \dots, x_n | c)$, puede estimarse contando las frecuencias relativas en un corpus.
- La probabilidad a priori de la clase, $P(c)$, representa con qué frecuencia ocurre esta clase.
- Sin algunas suposiciones simplificadoras, estimar la probabilidad de cada posible combinación de características en $P(x_1, x_2, \dots, x_n | c)$ requeriría un gran número de parámetros y conjuntos de entrenamiento imposiblemente grandes.
- Por lo tanto, los clasificadores Bayes Ingenuo realizan dos suposiciones simplificadoras.

0.3.1 Suposiciones de Independencia del Bayes Ingenuo Multinomial

- Suposición de Bolsa de Palabras: asumimos que la posición de las palabras en el documento no importa.
- Suposición de Independencia Condicional: asumimos que las probabilidades de las características $P(x_i | c_j)$ son independientes dada la clase c_j .
- En el clasificador Bayes Ingenuo Multinomial, la probabilidad de un documento con características x_1, x_2, \dots, x_n dada la clase c se puede calcular como:

$$P(x_1, x_2, \dots, x_n | c) = P(x_1 | c) \cdot P(x_2 | c) \cdot P(x_3 | c) \cdot \dots \cdot P(x_n | c)$$

0.3.2 Clasificador Bayes Ingenuo Multinomial

- La estimación del Máximo A Posteriori (MAP) para la clase c en el clasificador Bayes Ingenuo Multinomial se calcula como:

$$c_{\text{MAP}} = \arg \max_{c \in C} P(x_1, x_2, \dots, x_n | c)P(c)$$

- Alternativamente, podemos escribirlo como:

$$c_{\text{NB}} = \arg \max_{c \in C} P(c_j) \prod_{x \in X} P(x|c)$$

- $P(c_j)$ representa la probabilidad a priori de la clase c_j .
- $\prod_{x \in X} P(x|c)$ representa la verosimilitud de las características x_1, x_2, \dots, x_n dadas la clase c .

0.3.3 Aplicación de los clasificadores Naive Bayes multinomiales a la clasificación de texto

El clasificador Naive Bayes multinomial para la clasificación de texto se puede aplicar de la siguiente manera:

$$c_{\text{NB}} = \arg \max_{c_j \in C} P(c_j) \prod_{i \in \text{positions}} P(x_i | c_j)$$

Donde:

- c_{NB} representa la clase predicha para el documento de prueba.
- C es el conjunto de todas las clases posibles.
- $P(c_j)$ es la probabilidad previa de la clase c_j .
- $\prod_{i \in \text{positions}} P(x_i | c_j)$ calcula la probabilidad de cada característica x_i en la posición i dada la clase c_j .
- El producto se toma sobre todas las posiciones de palabras en el documento de prueba.

0.3.4 Problemas al multiplicar muchas probabilidades

Multiplicar muchas probabilidades puede resultar en un desbordamiento de punto flotante, especialmente cuando se manejan probabilidades pequeñas. Por ejemplo, $0,0006 \times 0,0007 \times 0,0009 \times 0,01 \times 0,5 \times 0,000008 \dots$

Para solucionar este problema, podemos utilizar logaritmos, ya que $\log(ab) = \log(a) + \log(b)$. En lugar de multiplicar las probabilidades, podemos sumar los logaritmos de las probabilidades. Así, el clasificador Naive Bayes multinomial se puede expresar utilizando logaritmos de la siguiente manera:

$$c_{\text{NB}} = \arg \max_{c_j \in C} \left(\log(P(c_j)) + \sum_{i \in \text{position}} \log(P(x_i | c_j)) \right)$$

Al tomar logaritmos, evitamos el problema del desbordamiento de punto flotante y realizamos cálculos en el espacio logarítmico. El clasificador se convierte en un modelo lineal, donde la predicción es el argmax de la suma de pesos (logaritmos de probabilidades) y las entradas (logaritmos de probabilidades condicionales). Por lo tanto, Naive Bayes es un clasificador lineal que opera en el espacio logarítmico.

0.3.5 Aprendizaje del modelo Naive Bayes multinomial

El primer intento: Estimaciones de máxima verosimilitud

- Las probabilidades se estiman utilizando las frecuencias observadas en los datos de entrenamiento.
- La probabilidad previa de una clase c_j se estima como:

$$\hat{P}(c_j) = \frac{N_{c_j}}{N_{\text{total}}}$$

donde N_{c_j} es el número de documentos en la clase c_j y N_{total} es el número total de documentos.

- La estimación de la probabilidad de la palabra w_i dada la clase c_j se calcula como:

$$\hat{P}(w_i|c_j) = \frac{\text{count}(w_i, c_j)}{\sum_{w \in V} \text{count}(w, c_j)}$$

donde $w \in V$ representa una palabra en el vocabulario V .

- El denominador es la suma de las frecuencias de todas las palabras en el vocabulario dentro de la clase c_j .

0.3.6 Estimación de parámetros

Para estimar los parámetros del modelo Naive Bayes multinomial, seguimos estos pasos:

- Creamos un mega-documento para cada tema c_j concatenando todos los documentos de ese tema.
- Calculamos la frecuencia de la palabra w_i en el mega-documento, que representa la fracción de veces que la palabra w_i aparece entre todas las palabras en los documentos del tema c_j .
- La probabilidad estimada $\hat{P}(w_i|c_j)$ de la palabra w_i dada la clase c_j se obtiene dividiendo el recuento de ocurrencias de w_i en el mega-documento del tema c_j por el recuento total de palabras en el mega-documento:

$$\hat{P}(w_i|c_j) = \frac{\text{count}(w_i, c_j)}{\sum_{w \in V} \text{count}(w, c_j)}$$

Aquí, $\text{count}(w_i, c_j)$ representa el número de veces que la palabra w_i aparece en el mega-documento del tema c_j , y $\text{count}(w, c_j)$ es el recuento total de palabras en el mega-documento.

0.3.7 Probabilidades cero y el problema de las palabras no vistas

Consideremos el escenario en el que no hemos encontrado la palabra "fantástico." en ningún documento de entrenamiento clasificado como positivo (pulgar hacia arriba). Utilizando la estimación de máxima verosimilitud, la probabilidad $\hat{P}(\text{"fantástico"} | \text{positivo})$ se calcularía como:

$$\hat{P}(\text{"fantástico"} | \text{positivo}) = \frac{\text{count}(\text{"fantástico"}, \text{positivo})}{\sum_{w \in V} \text{count}(w, \text{positivo})}$$

En este caso, el recuento de la palabra "fantástico" en los documentos positivos es cero, lo que conduce a una probabilidad cero:

$$\hat{P}(\text{"fantástico"} | \text{positivo}) = \frac{0}{\sum_{w \in V} \text{count}(w, \text{positivo})} = 0$$

Sin embargo, las probabilidades cero no pueden eliminarse, independientemente de la evidencia adicional presente. Esto plantea un problema al calcular la estimación del máximo a posteriori (MAP), que se utiliza para la clasificación:

$$c_{\text{MAP}} = \arg \max_c \left(\hat{P}(c) \prod_i \hat{P}(x_i | c) \right)$$

Con una probabilidad cero para una palabra, toda la expresión se vuelve cero, independientemente de la otra evidencia.

0.3.8 Suavizado Laplaciano (Add-1) para Naïve Bayes

Manejo de probabilidades cero con el suavizado Laplaciano (Add-1):

- Para abordar el problema de las probabilidades cero, podemos utilizar la técnica de suavizado Laplaciano (Add-1).
- La estimación suavizada $\hat{P}(w_i | c)$ se calcula como:

$$\hat{P}(w_i | c) = \frac{\text{count}(w_i, c) + 1}{\sum_{w \in V} (\text{count}(w, c) + 1)}$$

- Aquí, se agrega un recuento adicional de 1 tanto al numerador como al denominador.
- El denominador se ajusta agregando el tamaño del vocabulario V para garantizar una normalización adecuada.
- Al hacerlo, evitamos las probabilidades cero y permitimos que cierta masa de probabilidad se distribuya a palabras no vistas.
- Esta técnica de suavizado ayuda a mitigar el problema de las palabras no vistas y evita la eliminación completa de ciertas clases durante la clasificación.

0.3.9 Naïve Bayes multinomial: aprendizaje

Aprendiendo el modelo Naïve Bayes multinomial:

- Para aprender los parámetros del modelo, necesitamos calcular los términos $P(c_j)$ y $P(w_k | c_j)$.
- Para cada clase c_j en el conjunto de clases C , realizamos los siguientes pasos:
 - Recuperamos todos los documentos $docs_j$ que pertenecen a la clase c_j .
 - Calculamos el término $P(w_k | c_j)$ para cada palabra w_k en el vocabulario V :

$$P(w_k | c_j) = \frac{n_k + \alpha}{n + \alpha \cdot |\text{Vocabulary}|}$$

donde n_k representa el número de ocurrencias de la palabra w_k en el documento concatenado $Text_j$.

- Calculamos la probabilidad a priori $P(c_j)$:

$$P(c_j) = \frac{|docs_j|}{|\text{total number of documents}|}$$

- Para calcular $P(w_k | c_j)$, necesitamos extraer el vocabulario V del corpus de entrenamiento.

0.3.10 Palabras desconocidas

Tratamiento de palabras desconocidas en los datos de prueba:

- Cuando encontramos palabras desconocidas en los datos de prueba que no aparecen en los datos de entrenamiento o en el vocabulario, las ignoramos.
- Eliminamos estas palabras desconocidas del documento de prueba como si no estuvieran presentes en absoluto.
- No asignamos ninguna probabilidad a estas palabras desconocidas en el proceso de clasificación.

Esto es una visión general del modelo Naive Bayes multinomial y su aplicación a la clasificación de texto. Cabe destacar que existen variantes y extensiones más sofisticadas de Naive Bayes que se adaptan a diferentes requisitos y características de los datos.

0.4 Ejemplo

Datos de Entrenamiento:

Categoría	Texto
Negative	Just plain boring, entirely predictable and lacks energy.
Negative	No surprises and very few laughs.
Positive	Very powerful.
Positive	The most fun film of the summer.

Test:

Categoría	Texto
?	Predictable with no fun.

	Cat	Documents
Training	-	just plain boring entirely predictable and lacks energy no surprises and very few laughs + very powerful + the most fun film of the summer
Test	?	predictable with no fun

1. Prior from training:

$$\hat{P}(c_j) = \frac{N_{c_j}}{N_{total}} \quad P(-) = 3/5 \quad P(+) = 2/5$$

2. Drop "with"

3. Likelihoods from training:

$$p(w_i|c) = \frac{\text{count}(w_i, c) + 1}{(\sum_{w \in V} \text{count}(w, c)) + |V|}$$

$$P(\text{"predictable"}|-) = \frac{1+1}{14+20} \quad P(\text{"predictable"}|+) = \frac{0+1}{9+20}$$

$$P(\text{"no"}|-) = \frac{1+1}{14+20} \quad P(\text{"no"}|+) = \frac{0+1}{9+20}$$

$$P(\text{"fun"}|-) = \frac{0+1}{14+20} \quad P(\text{"fun"}|+) = \frac{1+1}{9+20}$$

4. Scoring the test set:

$$P(-)P(S|-) = \frac{3}{5} \times \frac{2 \times 2 \times 1}{34^3} = 6.1 \times 10^{-5}$$

$$P(+)P(S|+) = \frac{2}{5} \times \frac{1 \times 1 \times 2}{29^3} = 3.2 \times 10^{-5}$$

0.5 Naive Bayes como modelo de lenguaje

Cuando utilizamos características de palabras individuales y consideramos todas las palabras en el texto, el naive Bayes tiene una similitud importante con la modelización del lenguaje.

Específicamente, un modelo naive Bayes se puede ver como un conjunto de modelos de lenguaje de unigramas específicos de cada clase, en el que el modelo para cada clase instancia un modelo de lenguaje de unigrama.

Las características de verosimilitud del modelo naive Bayes asignan una probabilidad a cada palabra $P(\text{word}|c)$, y el modelo también asigna una probabilidad a cada oración:

$$P(s|c) = \prod_{i \in \text{positions}} P(w_i|c)$$

Consideremos un modelo naive Bayes con las clases positiva (+) y negativa (-) y los siguientes parámetros del modelo:

w	$P(w +)$	$P(w -)$
I	0.1	0.2
love	0.1	0.001
this	0.01	0.01
fun	0.05	0.005
film	0.1	0.1
...

Cada una de las dos columnas anteriores instancian un modelo de lenguaje que puede asignar una probabilidad a la oración "I love this fun film":

$$P(\text{"I love this fun film"}|+) = 0,1 \times 0,1 \times 0,01 \times 0,05 \times 0,1 = 0,0000005$$

$$P(\text{"I love this fun film"}|-) = 0,2 \times 0,001 \times 0,01 \times 0,005 \times 0,1 = 0,000000010$$

Como sucede, el modelo positivo asigna una probabilidad más alta a la oración:

$$P(s|\text{pos}) > P(s|\text{neg})$$

Cabe destacar que esto es solo la parte de verosimilitud del modelo naive Bayes; una vez que multiplicamos por la probabilidad a priori, un modelo naive Bayes completo podría tomar una decisión de clasificación diferente.

0.6 Evaluación

- Consideremos solo tareas de clasificación de texto binario.
- Imagina que eres el CEO de Delicious Pie Company.
- Quieres saber lo que la gente está diciendo sobre tus pasteles.
- Por lo tanto, construyes un detector de tweets de "Delicious Pie" con las siguientes clases:
 - Clase positiva: tweets sobre Delicious Pie Co.
 - Clase negativa: todos los demás tweets.

	Sistema Positivo	Sistema Negativo
Oro Positivo	Verdadero Positivo (VP)	Falso Negativo (FN)
Oro Negativo	Falso Positivo (FP)	Verdadero Negativo (VN)

0.6.1 La Matriz de Confusión 2x2

Recall (también conocido como **Sensibilidad** o **Tasa de Verdaderos Positivos**):

$$\text{Recall} = \frac{VP}{VP + FN}$$

Precisión:

$$\text{Precisión} = \frac{VP}{VP + FP}$$

Exactitud:

$$\text{Exactitud} = \frac{VP + VN}{VP + FP + VN + FN}$$

0.6.2 Evaluación: Exactitud

¿Por qué no usamos la exactitud como nuestra métrica?

Imagina que vimos 1 millón de tweets:

- 100 de ellos hablaban sobre Delicious Pie Co.
- 999,900 hablaban de otra cosa.

Podríamos construir un clasificador tonto que simplemente etiquete todos los tweets como "no sobre pasteles":

- ¡¡¡Obtendría una exactitud del 99.99 %!!! ¡¡¡Wow!!!
- ¡Pero sería inútil! ¡No devuelve los comentarios que estamos buscando!

Por eso usamos precisión y recall en su lugar.

0.6.3 Evaluación: Precisión y Recall

Precisión mide el porcentaje de elementos que el sistema detectó (es decir, los elementos que el sistema etiquetó como positivos) que son realmente positivos (según las etiquetas de oro humanas).

$$\text{Precisión} = \frac{\text{Verdaderos Positivos}}{\text{Verdaderos Positivos} + \text{Falsos Positivos}}$$

Recall mide el porcentaje de elementos que el sistema identificó correctamente de todos los elementos que deberían haber sido identificados.

$$\text{Recall} = \frac{\text{Verdaderos Positivos}}{\text{Verdaderos Positivos} + \text{Falsos Negativos}}$$

0.6.4 ¿Por qué Precisión y Recall?

Considera nuestro clasificador tonto de pasteles que simplemente etiqueta nada como "sobre pasteles".

- Exactitud = 99.99 % (etiqueta correctamente la mayoría de los tweets como no relacionados con pasteles)
 - Recall = 0 (no detecta ninguno de los 100 tweets relacionados con pasteles)
- La precisión y el recall, a diferencia de la exactitud, enfatizan los verdaderos positivos:
- Se centran en encontrar las cosas que se supone que debemos buscar.

0.6.5 Una Medida Combinada: Medida F

La medida F es un número único que combina la precisión (P) y el recall (R), definida como:

$$F_{\beta} = \frac{(\beta^2 + 1)PR}{\beta^2 P + R}$$

La medida F, definida con el parámetro β , pondera diferencialmente la importancia del recall y la precisión.

- $\beta > 1$ favorece al recall
- $\beta < 1$ favorece a la precisión

Cuando $\beta = 1$, la precisión y el recall son iguales, y tenemos la medida F_1 equilibrada:

$$F_1 = \frac{2PR}{P + R}$$

0.6.6 Conjuntos de Prueba de Desarrollo ("Devsets")

- Para evitar el sobreajuste y proporcionar una estimación más conservadora del rendimiento, comúnmente utilizamos un enfoque de tres conjuntos: conjunto de entrenamiento, conjunto de desarrollo y conjunto de prueba.

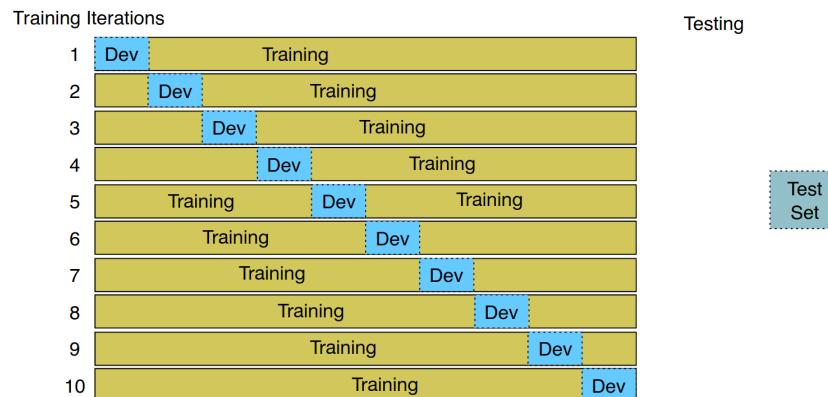


- **Conjunto de entrenamiento:** Se utiliza para entrenar el modelo.
- **Conjunto de desarrollo:** Se utiliza para ajustar el modelo y seleccionar los mejores hiperparámetros.
- **Conjunto de prueba:** Se utiliza para informar el rendimiento final del modelo.
- Este enfoque garantiza que el modelo no esté ajustado específicamente al conjunto de prueba, evitando el sobreajuste.
- Sin embargo, crea una paradoja: queremos la mayor cantidad de datos posible para el entrenamiento, pero también para el conjunto de desarrollo.
- ¿Cómo dividimos los datos?

0.6.7 Validación Cruzada: Múltiples Divisiones

- La validación cruzada nos permite utilizar todos nuestros datos para el entrenamiento y la prueba sin tener un conjunto de entrenamiento, conjunto de desarrollo y conjunto de prueba fijos.

- Elegimos un número k y dividimos nuestros datos en k subconjuntos disjuntos llamados pliegues.
- En cada iteración, uno de los pliegues se selecciona como conjunto de prueba mientras que los $k - 1$ pliegues restantes se utilizan para entrenar el clasificador.
- Calculamos la tasa de error en el conjunto de prueba y repetimos este proceso k veces.
- Finalmente, promediamos las tasas de error de estas k ejecuciones para obtener una tasa de error promedio.
- Por ejemplo, la validación cruzada de 10 pliegues implica entrenar 10 modelos con el 90 % de los datos y probar cada modelo por separado.
- Las tasas de error resultantes se promedian para obtener la estimación final del rendimiento.
- Sin embargo, la validación cruzada requiere que todo el corpus sea ciego, lo que impide examinar los datos para sugerir características o comprender el comportamiento del sistema.
- Para abordar esto, se crea un conjunto de entrenamiento y un conjunto de prueba fijos, y se realiza la validación cruzada de 10 pliegues dentro del conjunto de entrenamiento.
- La tasa de error se calcula convencionalmente en el conjunto de prueba.



0.6.8 Matriz de Confusión para clasificación de 3 clases

		gold labels			
		urgent	normal	spam	
system output	urgent	8	10	1	$\text{precision}_u = \frac{8}{8+10+1}$
	normal	5	60	50	$\text{precision}_n = \frac{60}{5+60+50}$
	spam	3	30	200	$\text{precision}_s = \frac{200}{3+30+200}$
		$\text{recall}_u = \frac{8}{8+5+3}$	$\text{recall}_n = \frac{60}{10+60+30}$	$\text{recall}_s = \frac{200}{1+50+200}$	

Cómo combinar métricas binarias (Precisión, Recall, F_1) de más de 2 clases para obtener una métrica única:

- Macro-promedio:
 - Calcular las métricas de rendimiento (Precisión, Recall, F_1) para cada clase individualmente.

- Promediar las métricas en todas las clases.
- Micro-promedio:
 - Recopilar las decisiones para todas las clases en una matriz de confusión.
 - Calcular la Precisión y el Recall a partir de la matriz de confusión.

		Class 1: Urgent		Class 2: Normal		Class 3: Spam		Pooled	
		true	true	true	true	true	true	true	true
		urgent	not	normal	not	spam	not	yes	no
system	urgent	8	11	system	60	55	system	200	33
	not	8	340		40	212		51	83

$\text{precision} = \frac{8}{8+11} = .42$

 $\text{precision} = \frac{60}{60+55} = .52$

 $\text{precision} = \frac{200}{200+33} = .86$

 $\text{microaverage precision} = \frac{268}{268+99} = .73$

$\text{macroaverage precision} = \frac{.42+.52+.86}{3} = .60$