Primeros Resultados según Matriz de Confusión

Medidas de desempeño de clasificadores.

Como es sabido, la *matriz de confusión* mide el rendimiento de un clasificador.
y se expresa en el caso de que solo existan dos clases como:

		Predicción	
		Positivo	***Negativo***
Real	***Positivo***	Positivo Verdadero (**TP**)	Falso Negativo (**FN**)
	Negativo	Falso Positivo (**FP**)	Negativo Verdadero (**TN**)

Las medidas más inmediatas de evaluación del modelo son:

- Exactitud: $\text{Acc} = \frac{TP+TN}{TP+TN+FP+FN}$
- Tasa de error: $Err = \frac{FP+FN}{TP+TN+FP+FN} = 1 - Acc$

Sin embargo, cuando existe un claro desequilibrio entre clases, como es el caso de los tuits de odio, se utilizarán los indicadores clásicos en clasificación binaria:

1. Precisión (*p*recision)
2. Exhaustividad (*r*ecall)

- **Precisión** (p) es el porcentaje de los *tuits* clasificados *correctamente* como de odio - *TP* -del total de los asignados a dicha clase por el clasificador - *TP+FP* - (% de aciertos).

$$p = \frac{TP}{TP + FP}$$

- **Exhaustividad** (r) es el porcentaje de los *tuits* de la clase odio existentes en el fichero que han sido clasificados correctamente.

$$r = \frac{TP}{TP + FN}$$

Proceso seguido.

Según lo que se deduce del conjunto inicial de muestra, el contenido de los tuits utilizados para etiquetado es el siguiente

Clase	Frecuencia	%
Odio	204	0,2%
Neutro	99.946	99,8%
Total	100.150	100%

Donde se aprecia claramente que estamos en un clarísimo desequilibrio de clases (0,2% frente a 99,8%).

El proceso que se ha seguido para la obtención del conjunto de entrenamiento parte del conjunto $\mathcal{S}$ (muestra) que se filtra mediante una lista de vocabulario obteniendo un nuevo conjunto $\mathcal{R}$ de mucha menor cardinalidad que se etiqueta manualmente, produciendo el conjunto de entrenamiento $\mathcal{T}$, a partir del cual se procederá a la estimación de los modelos de clasificación como muestra la Figura 1.

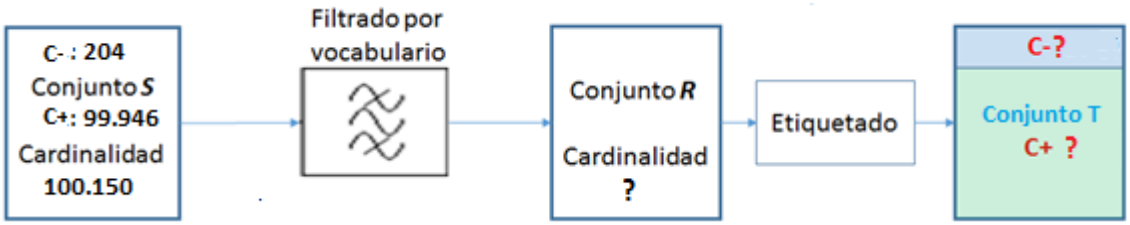


Figura 1

Seguidamente se procede a la prueba de los modelos (Bayes, K-NN, Redes Neuronales,…) a fin de seleccionar el más adecuado.

Resultados obtenidos.

En nuestro caso, los resultados son los siguientes:

1. **Matriz de confusión**:

		Predicción		
		Odio	Neutro	
Real	Odio	32	172	***204***
	Neutro	20	99.926	***99.946***
		52	***100.098***	**100.150**

		Predicción		
		Odio	Neutro	
Real	Odio	0,032%	0,172%	***0,204%***
	Neutro	0,020%	99,776%	***99,796%***
		0,052%	***99,948%***	**100%**

TP	32	0,032%
TN	99.926	99,776%
FP	20	0,020%
FN	172	0,172%
Total	100.150	100%

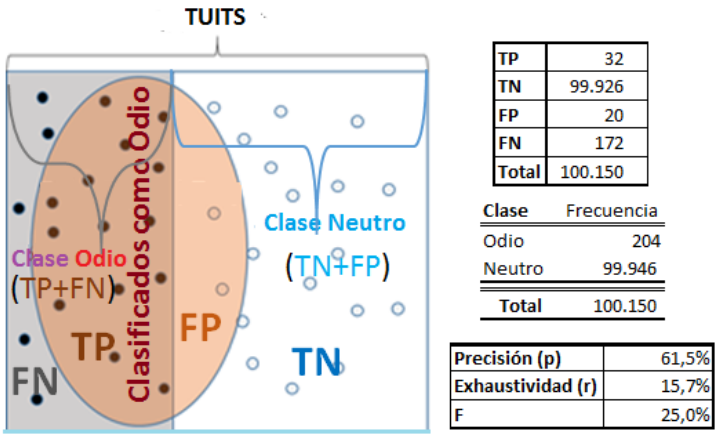
2. **Medidas de desempeño**:

Exactitud(Acc)	99,81%
Error(Err)	0,19%

Precisión (p)	61,5%
Exhaustividad (r)	15,7%
F	25,0%

Puede verse que, como era de esperar, el error es muy pequeño, sin embargo, el rendimiento del clasificador es bastante pobre sobre todo debido a su baja *exhaustividad*, lo que implica que ***el 84% de los tuits de odio originales no son detectados.***

La Figura 2 muestra gráficamente los diferentes conceptos.



TP	32
TN	99.926
FP	20
FN	172
Total	100.150

Clase	Frecuencia
Odio	204
Neutro	99.946
Total	100.150

Precisión (p)	61,5%
Exhaustividad (r)	15,7%
F	25,0%

Figura 2

Conclusión.

Si los resultados mostrados anteriormente son correctos, es claro que, como consecuencia del gigantesco desequilibrio existente entre las clases - 2: 1000 - , la *exhaustividad* de la clasificación es muy pequeña: solamente el 16% de los tuits de odio son clasificados correctamente y la precisión tampoco puede decirse que sea muy elevada.

Es preciso analizar más en detalle los resultados y ver dónde está el fallo, si en el filtro o en el clasificador.

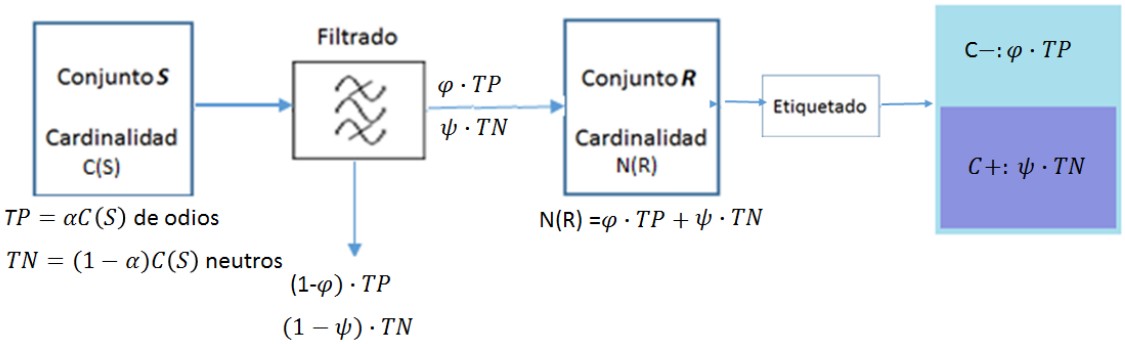


Figura 3: Filtrado de Tuits

Para ello es preciso ver

1º Cuántas instancias de odio (φ·*TP*) pasan el filtro y

2º Cuántos de los que pasan se clasifican erróneamente.

Todo ello en los dos posibles métodos de clasificación:

1. Sin filtrado previo:

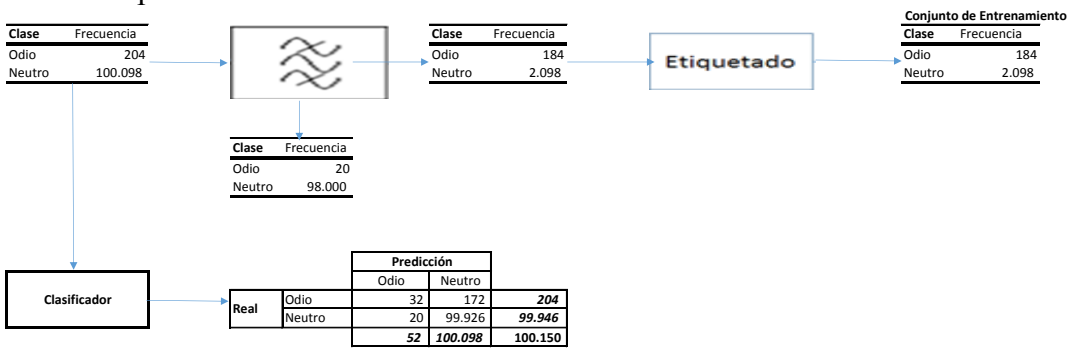


Figura 4: Clasificación sin Filtrado Previo

2. Con filtrado previo y posterior clasificación:

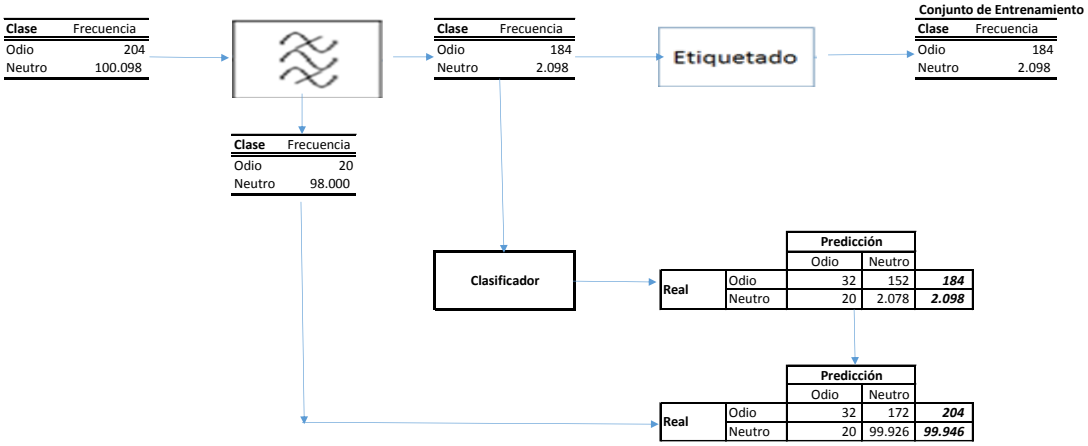


Figura 5: Clasificación con Filtrado Previo