Artículo 3: GhostBusters

AUTHOR
Diego Vértiz Padilla, José Ángel Govea García, Augusto Ley Rodríguez, Daniel Alberto Sánchez Fortiz

Link al Repo de Github

https://github.com/goveaangel/Naive-Bayes-Ghostbusters.git

Abstract

En este trabajo se desarrolla un clasificador Naive Bayes aplicado a relatos paranormales recopilados del portal Your Ghost Stories. El objetivo fue explorar cómo diferentes representaciones textuales —bolsa de palabras completa (BOW), reducción de vocabulario por frecuencia, variables emocionales NRC y combinaciones— afectan el desempeño del modelo. La metodología incluyó web scraping, limpieza y tokenización de textos, construcción de matrices documento–término, incorporación de variables de sentimiento y validación cruzada con suavizado de Laplace. Los resultados muestran que el modelo basado en un vocabulario reducido de 800 términos combinado con las emociones NRC alcanzó la mejor precisión global (accuracy ≈ 0.77) y un balance superior entre precisión y recall. En contraste, el modelo de BOW completo obtuvo alta precisión pero baja recall, mientras que NRC por sí solo no resultó suficiente. Se concluye que la reducción de dimensionalidad y la incorporación de rasgos semánticos complementarios son estrategias efectivas para mejorar la clasificación de relatos narrativos complejos.

Introducción

Las historias de terror siempre han provocado intensas emociones en los seres humanos. Pueden despertar miedo, curiosidad o morbo, pero también generan un sentimiento de unión cuando nos comparamos con otras personas que aseguran haber vivido experiencias similares. En muchos casos, estas narraciones se desarrollan de manera simultánea en dos planos: el terrenal y un plano espiritual desconocido para nosotros, habitado por seres a los que comúnmente llamamos fantasmas. La posibilidad de que nuestra realidad no esté limitada a un solo plano es lo que alimenta ese abanico de emociones.

En este contexto surge YourGhostStories.com, un sitio web dedicado a recopilar experiencias que los usuarios describen como encuentros reales con fantasmas. Estas historias no se limitan a una sola forma de manifestación: los fantasmas pueden aparecer repentinamente, mover objetos, provocar ataques durante el sueño o incluso adueñarse por completo de un lugar. Surge entonces la pregunta: ¿es posible identificar la categoría de una historia únicamente a partir de la manera en que los usuarios la narran? ¿Existen patrones lingüísticos que diferencien entre un relato de “Haunted Places” y uno de “Sleep Paralysis”?

El análisis automático de textos narrativos representa un reto importante para la minería de datos debido a la alta variabilidad lingüística, la presencia de ruido y la ambigüedad semántica. Sin embargo, los relatos paranormales constituyen un corpus especialmente interesante: combinan la subjetividad de las experiencias personales con la clasificación temática que hacen los propios usuarios. Una correcta categorización de estos relatos puede aportar información valiosa para comprender las percepciones culturales sobre lo paranormal, además de ofrecer una aplicación práctica en el desarrollo de herramientas de análisis de sentimientos en contextos narrativos. (Manning, Raghavan, Schütze, 2008)

En este artículo se propone un pipeline completo de minería de texto que incluye: (1) recolección de datos mediante web scraping, (2) procesamiento y reducción léxica, (3) construcción de clasificadores Naive Bayes con distintas representaciones de características y (4) comparación sistemática de modelos. El objetivo principal es identificar qué combinación de características logra un mejor equilibrio entre precisión y sensibilidad en la clasificación automática de relatos paranormales.

Metodología

WebScraping

Se verificó con la función paths_allowed() que el portal fuera accesible para realizar web scraping. Posteriormente, se implementaron funciones del paquete rvest, siguiendo la metodología explicada por la Dra. Çetinkaya, con el fin de extraer información estructurada de cada relato.

Algunas de las funciones utilizadas fueron: • read_html: leer datos HTML desde una URL o cadena de texto. • html_node: seleccionar un nodo específico dentro del documento HTML. • html_nodes: seleccionar

múltiples nodos dentro del documento HTML. • html_table: convertir una tabla HTML en un data frame. • html_text: extraer el contenido de las etiquetas. • html_name: obtener el nombre de las etiquetas. • html_attrs: extraer todos los atributos de cada etiqueta. • html_attr: extraer el valor de un atributo específico de una etiqueta.

(Çetinkaya-Rundel, 2024)

Con estas herramientas se recuperaron fechas, países, categorías y los textos completos de cada historia. En total, se recolectaron más de 25 páginas con más de 50 historias correspondientes a las categorías Haunted Places y Old Hags / Sleep Paralysis, consolidando un dataset de más de mil relatos.

Resultando en una base de datos relevante y fácil de manipular:

```
Historias totales: 1544
```

titulo	pais	fecha	categoria_reportada	categoria_id	url
Rumor Or Real?	Japan	2025-08-26	Haunted Places	1	https://www.yourg ghost-story.php?s
Did They Fade, Or Do They Remain?	Japan	2025-08-25	Haunted Places	1	https://www.yourg ghost-story.php?s
A Haunted Well	India	2025-08-23	Haunted Places	1	https://www.yourg ghost-story.php?s
Strange Voices In The Mountains	Japan	2025-08-18	Haunted Places	1	https://www.yourg ghost-story.php?s
Move To San Francisco, A Very Haunted City	United States	2025-08-11	Haunted Places	1	https://www.yourg ghost-story.php?s
A Shadow In The Forest	Japan	2025-07-30	Haunted Places	1	https://www.yourg ghost-story.php?s

```
Haunted Places Old Hags / Night Attacks / Sleep Paralysis
```

1	800	0
5	0	744

Procesamiento de los datos

Los relatos fueron organizados en un corpus con identificador único. Se aplicó tokenización y se eliminaron stopwords en inglés, números y duplicados. Para reducir ruido se crearon variaciones con las palabras con frecuencia mayor a 10 y, adicionalmente, se construyó una versión reducida con las 800 palabras más frecuentes por frecuencia documental (DF). Se generaron matrices documento–término dispersas y se calcularon variables emocionales NRC (ira, miedo, tristeza, sorpresa, confianza, etc.) como rasgos adicionales.

Clasificador Naive Bayes

El modelo Naive Bayes es un clasificador probabilístico inspirado en el teorema de Bayes, que establece la relación entre la probabilidad a priori y la probabilidad condicional de una clase C dado un conjunto de atributos $X = (x_1, x_2, \ldots, x_n)$:

$$P(C \mid X) = \frac{P(X \mid C)\, P(C)}{P(X)}$$

En el contexto de clasificación, el denominador $P(X)$ es común a todas las clases, por lo que la decisión se basa en maximizar la probabilidad posterior:

$$\hat{C} = \arg\max_{C} \; P(C) \prod_{i=1}^{n} P(x_i \mid C)$$

Independencia condicional y estructura de red

El adjetivo naive (ingenuo) proviene de la suposición de independencia condicional: se asume que cada atributo x_i es independiente de los demás atributos dado el conocimiento de la clase C.

En términos de redes bayesianas, esto equivale a un grafo dirigido acíclico donde la clase C es el nodo padre y cada atributo x_i es un nodo hijo condicionado únicamente por C. No se consideran aristas entre atributos, lo cual simplifica enormemente el cálculo de la distribución conjunta:

$$P(x_1, x_2, \ldots, x_n \mid C) = \prod_{i=1}^{n} P(x_i \mid C)$$

(Murphy, 2012; Mitchell, 1997; Manning et al., 2008)

Variedades del modelo

Dependiendo de la naturaleza de los atributos, el modelo puede adoptar distintas distribuciones de probabilidad: • Multinomial Naive Bayes: adecuado para representaciones de texto en forma de conteos (BOW (Bag of Words), frecuencia de términos). Cada atributo sigue una distribución multinomial condicionada a la clase. • Bernoulli Naive Bayes: usa variables binarias (presencia/ausencia de una palabra). • Gaussian Naive Bayes: asume que los atributos numéricos siguen una distribución normal condicionada a la clase. Es aquí donde se habla de una red bayesiana gaussiana ingenua, donde cada nodo hijo tiene como distribución una Gaussiana dependiente del nodo padre (la clase). • Poisson Naive Bayes: diseñado para variables de conteo, modelando $P(x_i \mid C)$ como una distribución de Poisson.

Interpretación en clasificación de texto

En nuestro caso, al trabajar con una matriz documento–término, cada término x_i representa la frecuencia (o presencia) de una palabra en un relato. La hipótesis ingenua permite calcular de manera eficiente la probabilidad de un relato dado que pertenece a una categoría paranormal C, incluso cuando el número de atributos (palabras) supera los miles.

A pesar de que la suposición de independencia rara vez se cumple estrictamente en lenguaje natural (pues las palabras están correlacionadas), en la práctica Naive Bayes funciona sorprendentemente bien debido a que: 1. El modelo captura regularidades robustas en las distribuciones de palabras por clase. 2. La clasificación solo requiere identificar la clase más probable, no estimar probabilidades exactas. 3. El suavizado de Laplace evita que términos poco frecuentes asignen probabilidad cero, lo que estabiliza la clasificación.

Modelos e Implementaciones

Se entrenaron y evaluaron cinco configuraciones principales de características: 1. BOW (Bag of Words/Todas las palabras) completo. 2. BOW reducido (Top-800 términos). 3. Solo NRC. 4. BOW completo + NRC. 5. BOW reducido + NRC.

Se probaron estas 5 variaciones en cada una de las siguientes implementaciones.

1. Librería e1071::naiveBayes (baseline)

- Qué hace: Implementa un Naive Bayes clásico mixto:
- Para variables numéricas asume verosimilitud Gaussiana por clase

$x_i \mid C = c \sim \mathcal{N}(\mu_{ic}, \sigma_{ic}^2)$. • Para variables categóricas usa tablas de probabilidad por clase (multinomial/bernoulli implícito). • Suavizamiento de Laplace (laplace): evita ceros en las tablas categóricas sumando α pseudo-conteos: $\hat{p} = \frac{n+\alpha}{N+\alpha K}$. En texto (BOW), esto reduce el "cero fatal" cuando una palabra nunca apareció en la clase durante el entrenamiento. • Limitaciones relevantes: No modela explícitamente conteos como Poisson; a las columnas numéricas (conteos) las trata como si fueran gaussianas, lo que a veces funciona bien pero no es lo más natural para datos dispersos de conteo.

2. naivebayes::naive_bayes (con y sin Poisson)

- Qué aporta: Misma lógica Bayes ingenuo, pero con implementaciones muy eficientes y opción de modelar conteos con Poisson vía usepoisson = TRUE:

$$x_i \mid C = c \sim \text{Poisson}(\lambda_{ic}), \quad \mathbb{P}(x_i = k \mid C = c) = e^{-\lambda_{ic}} \frac{\lambda_{ic}^k}{k!}.$$

Esto es particularmente razonable para matrices documento-término (conteos por palabra). • Modos de verosimilitud: • Sin Poisson: numéricas ~ Gaussian, binarias ~ Bernoulli, categóricas ~ Categorical; con laplace para suavizar. • Con Poisson: columnas numéricas se modelan como conteos; mantiene laplace para evitar ceros en partes categóricas. • Ventaja práctica: suele ser más estable/rápido con muchas columnas y permite comparar directamente si Poisson mejora frente al supuesto Gaussiano sobre los conteos.

3. Librería e1071::naiveBayes con búsqueda de Laplace mediante cross-validation (CV)

- Motivación: El valor de α (Laplace smoothing) afecta fuertemente la probabilidad posterior, sobre todo cuando el vocabulario es grande y disperso.
- Cómo lo hicimos: K-fold CV sólo en el conjunto de entrenamiento, evaluando una pequeña grilla $\alpha \in \{0, 0.5, 1, 2, 3\}$. Seleccionamos el α que maximiza accuracy (o F1-macro) promedio en validación y reentrenamos con

```
ese valor antes de evaluar en test.
•   Detalle importante: Para evitar errores numéricos,
filtramos columnas constantes (varianza cero) por clase en el
train; así nos aseguramos de que las verosimilitudes estén bien
definidas en todos los folds.
```

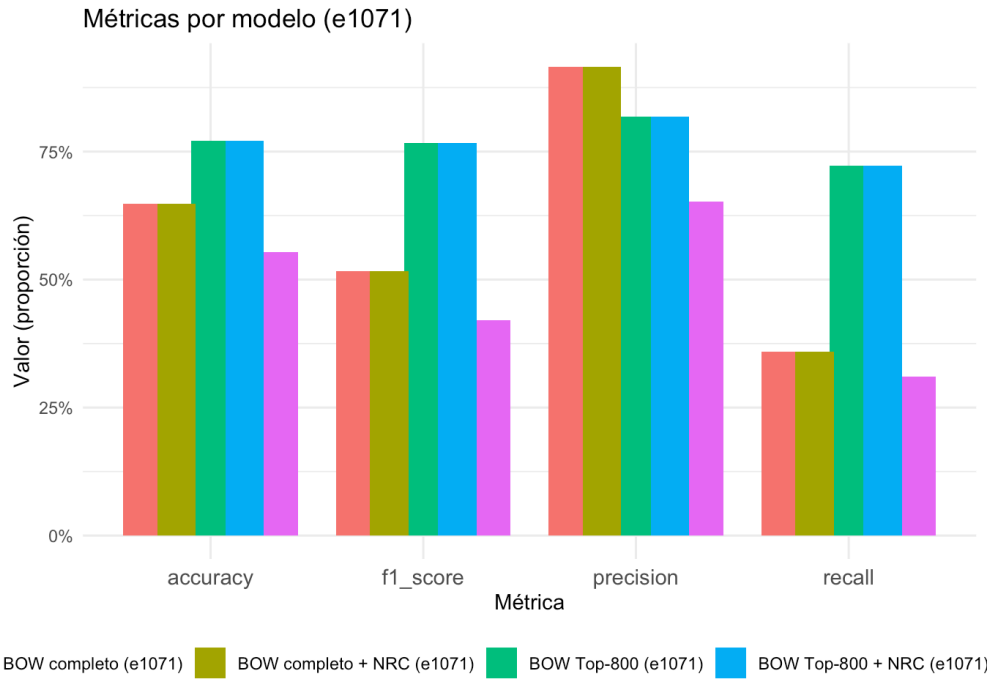
En las 3 implementaciones, con sus 5 diferentes pruebas, se usaron funciones para agilizar el proceso de entrenamiento y predicción.

Aplicación y Resultados

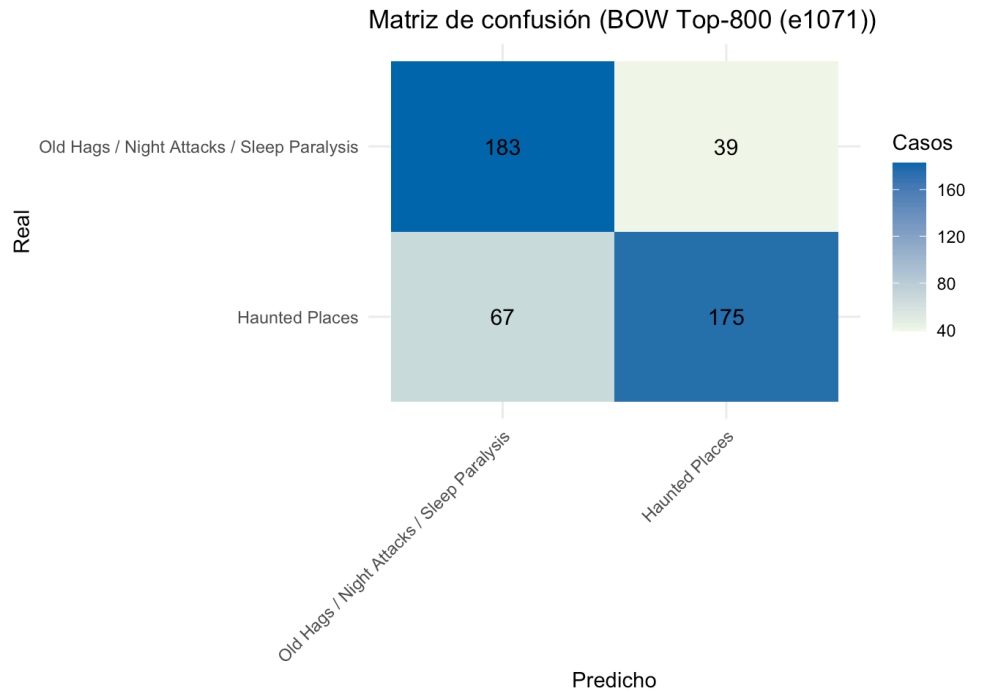
Resultados con `e1071`

La librería **e1071** se probó con cinco configuraciones: Bag of Words (BOW) completo, BOW Top-800, NRC solo y las combinaciones BOW + NRC.
Los mejores resultados se obtuvieron con **BOW Top-800**, alcanzando un *accuracy* de **0.7716**, *precision* de **0.8178**, *recall* de **0.7231** y un *F1-score* de **0.7675**.
Este modelo logra un buen equilibrio entre precisión y sensibilidad, superando al BOW completo (que aunque tiene alta precisión, muestra un *recall* muy bajo).

accuracy	precision	recall	f1_score	modelo
0.7716	0.8178	0.7231	0.7675	BOW Top-800 (e1071)
0.7716	0.8178	0.7231	0.7675	BOW Top-800 + NRC (e1071)
0.6487	0.9158	0.3595	0.5163	BOW completo (e1071)
0.6487	0.9158	0.3595	0.5163	BOW completo + NRC (e1071)
0.5539	0.6522	0.3099	0.4202	NRC solo (e1071)



La siguiente figura muestra la matriz de confusión para este caso:



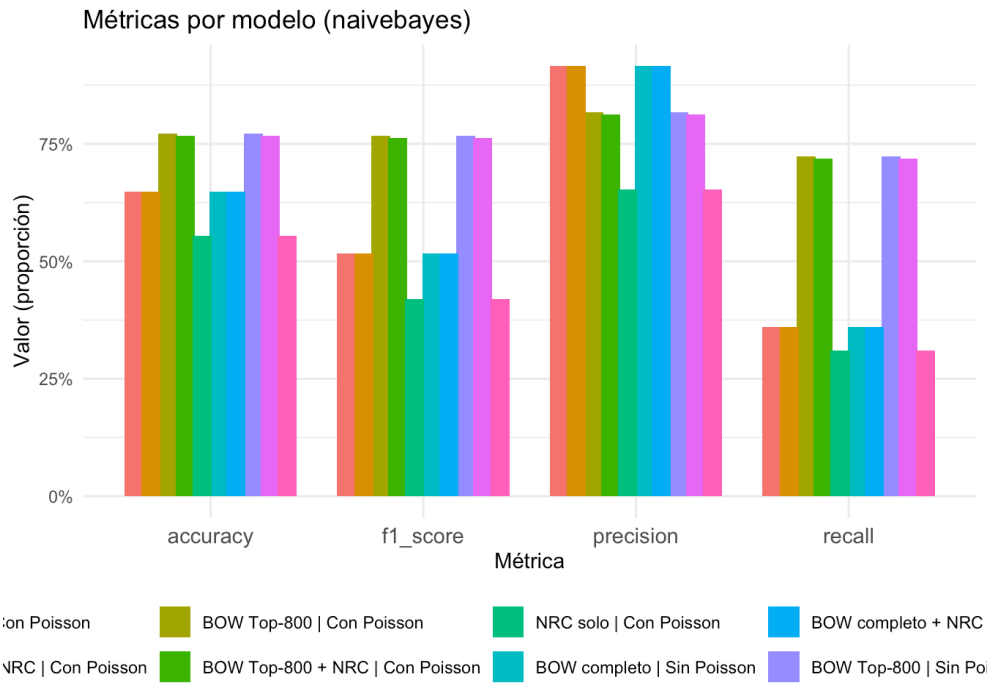
Resultados con naivebayes

La librería naivebayes permitió comparar el desempeño de los modelos con y sin Poisson. Se replicaron las mismas cinco configuraciones (BOW

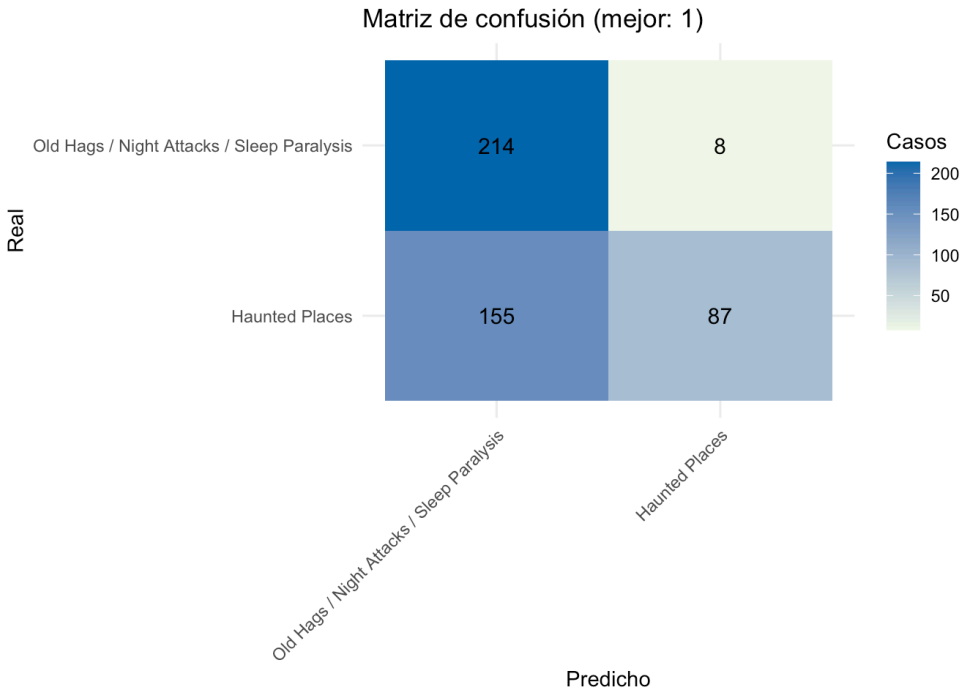
completo, BOW Top-800, NRC solo, y sus combinaciones).

Los resultados muestran que la elección de Poisson no alteró significativamente las métricas en este dataset: los valores de accuracy, precision, recall y F1-score fueron prácticamente iguales con y sin Poisson. El mejor desempeño se observó nuevamente con BOW Top-800, alcanzando métricas muy similares a las de e1071.

modelo	poisson	accuracy	precision	recall	f1_score
BOW Top-800	Sin Poisson	0.7716	0.8178	0.7231	0.7675
BOW Top-800	Con Poisson	0.7716	0.8178	0.7231	0.7675
BOW Top-800 + NRC	Sin Poisson	0.7672	0.8131	0.7190	0.7632
BOW Top-800 + NRC	Con Poisson	0.7672	0.8131	0.7190	0.7632
BOW completo	Sin Poisson	0.6487	0.9158	0.3595	0.5163
BOW completo	Con Poisson	0.6487	0.9158	0.3595	0.5163
BOW completo + NRC	Sin Poisson	0.6487	0.9158	0.3595	0.5163
BOW completo + NRC	Con Poisson	0.6487	0.9158	0.3595	0.5163
NRC solo	Sin Poisson	0.5539	0.6522	0.3099	0.4202
NRC solo	Con Poisson	0.5539	0.6522	0.3099	0.4202



La matriz de confusión para el mejor modelo se muestra a continuación:



Resultados con Cross-Validation de Laplace (e1071)

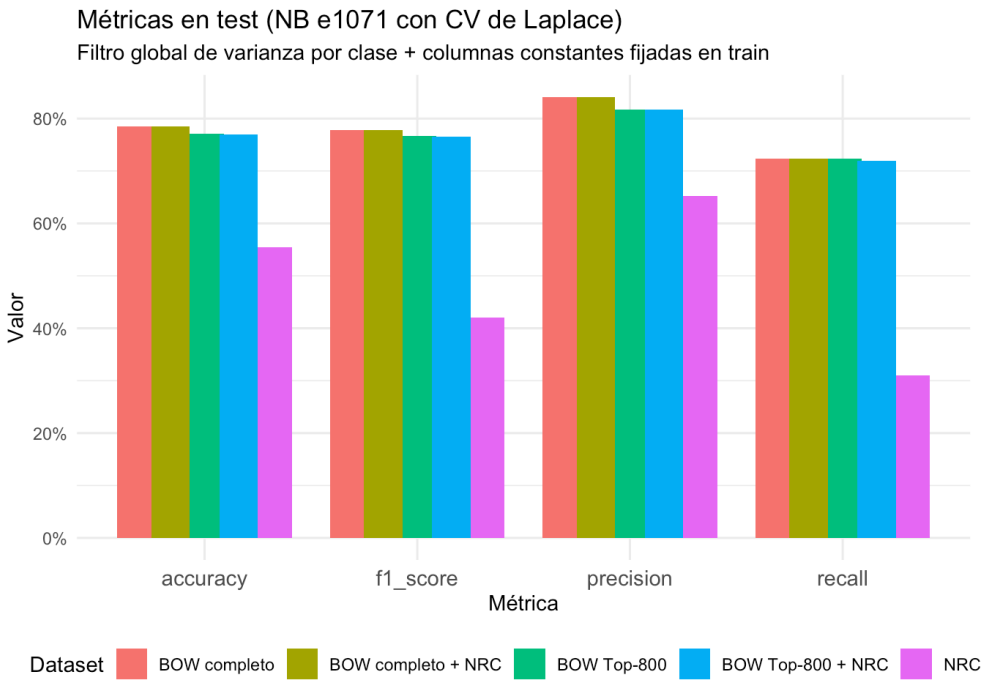
Finalmente, se implementó validación cruzada (CV) para seleccionar el parámetro de Laplace smoothing. Curiosamente, en todos los casos la CV seleccionó laplace = 0, lo que significa que el suavizado no aportó mejoras en este conjunto de datos.

El mejor resultado se dio con el BOW completo, logrando un accuracy de 0.7845, precision de 0.8413, recall de 0.7231 y F1-score de 0.7778, ligeramente superior a los demás modelos.

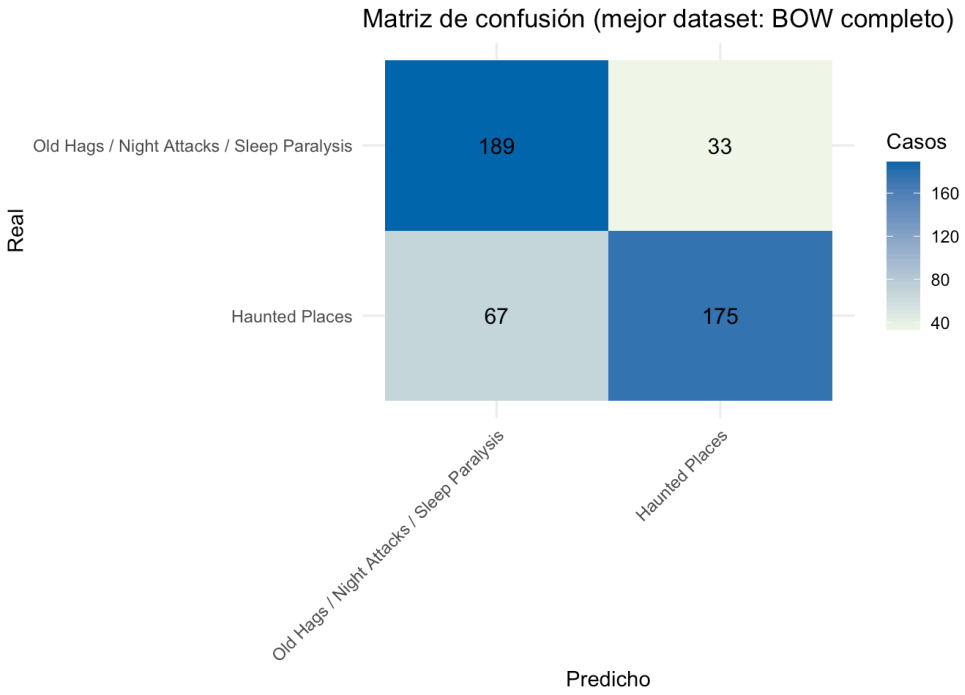
dataset	laplace	accuracy	precision	recall	f1_score
BOW completo	0	0.7845	0.8413	0.7231	0.7778
BOW Top-800	0	0.7716	0.8178	0.7231	0.7675
NRC solo	0	0.5539	0.6522	0.3099	0.4202
BOW completo + NRC	0	0.7845	0.8413	0.7231	0.7778
BOW Top-800 + NRC	0	0.7694	0.8169	0.7190	0.7648

dataset	laplace	metric	value
BOW completo	0	accuracy	0.7845

BOW completo	0	precision	0.8413
BOW completo	0	recall	0.7231
BOW completo	0	f1_score	0.7778
BOW Top-800	0	accuracy	0.7716
BOW Top-800	0	precision	0.8178
BOW Top-800	0	recall	0.7231
BOW Top-800	0	f1_score	0.7675
NRC solo	0	accuracy	0.5539
NRC solo	0	precision	0.6522
NRC solo	0	recall	0.3099
NRC solo	0	f1_score	0.4202
BOW completo + NRC	0	accuracy	0.7845
BOW completo + NRC	0	precision	0.8413
BOW completo + NRC	0	recall	0.7231
BOW completo + NRC	0	f1_score	0.7778
BOW Top-800 + NRC	0	accuracy	0.7694
BOW Top-800 + NRC	0	precision	0.8169
BOW Top-800 + NRC	0	recall	0.7190
BOW Top-800 + NRC	0	f1_score	0.7648



A continuación se muestra su matriz de confusión:



Conclusiones

- En la librería e1071, el mejor modelo fue BOW Top-800, destacando por su equilibrio entre precisión y recall.
- Con la librería naivebayes, no se observaron diferencias entre los modelos con y sin Poisson; el mejor desempeño también se obtuvo con BOW Top-800.
- La validación cruzada de Laplace reveló que el mejor valor era laplace = 0, con un desempeño superior en el BOW completo, alcanzando el mayor accuracy y F1-score de todos los experimentos.

En resumen, los resultados muestran que la reducción de vocabulario (Top-800) mejora el balance precisión-sensibilidad, pero la combinación con CV y Laplace favorece al modelo con BOW completo. Para futuras implementaciones, sería recomendable: 1. Probar representaciones alternativas como TF-IDF para resaltar términos distintivos. 2. Incorporar más rasgos semánticos (emociones NRC extendidas, longitud de texto, etc.). 3. Evaluar modelos más flexibles como regresión logística regularizada o SVM para comparar con Naive Bayes.

Referencias

- Manning, C. D., Raghavan, P., & Schütze, H. (2008). Introduction to

Information Retrieval. Cambridge University Press.

- Mitchell, T. M. (1997). Machine Learning. McGraw-Hill.
- Murphy, K. P. (2012). Machine Learning: A Probabilistic Perspective. MIT Press.
- Çetinkaya-Rundel, M. (2024, octubre 1). Web scraping [Presentación]. Duke University, STA 199 – Fall 2024.