

# Proyecto - Filtro de reseñas.

Estefany Carolina Segura Linares  
Brayan David Prieto Aya  
Alejandro Jaramillo Vallejo

2025-09-08

## Justificación de su importancia.

### 1. Definición del problema

#### 1.2 Justificación de la importancia.

Dentro del sistema de votación de internet, las reseñas son de extrema importancia, debido a que es la primera impresión que recibe cualquier internauta ante el producto, servicio o lugar. Las reseñas falsas pueden generar un impacto malintencionado sobre el producto, lo que impide que exista un sistema de votación transparente en internet, reduciendo gradualmente la confianza de cualquier usuario ante el producto, o en su defecto, el medio por donde lee la reseña al enaltecer o vituperar incorrectamente, pidiendo de sobremanera un filtro que pueda mitigar la problemática.

#### 1.3 Preguntas de investigación o hipótesis a responder.

Si las empresas consideran la presencia de sus servicios, para mejorar su rendimientos recurrirán a reseñas spam para aumentar sus ventas de prestación de servicios.

¿Existe una relación entre la edad y la cantidad de reseñas que publica una persona con respecto a un producto?.

Si la mayoría de usuarios del mundo están conectados a windows, será el sistema operativo que tenga más reseñas de carácter spam.

¿Existe algún tipo de relación de las reseñas de carácter spam con las aplicaciones más importantes?.

¿La dispersión es muy grande tanto para las reseñas verdaderas como de spam?.

### 2. Descripción del dataset

#### 2.2 Variables disponibles y su significado

- **review\_id**: Número identificativo de la reseña.
- **user\_id**: Número identificativo del usuario.
- **app\_name**: Nombre de la app desde donde se hizo la reseña.
- **app\_category**: Categoría de la app desde donde se hace la reseña.
- **review\_text**: Contenido de texto de la reseña.
- **review\_language**: Lenguaje de la reseña.

- **rating:** Puntuación de la reseña.
- **review\_date:** Fecha de publicación de la reseña.
- **verified\_purchase:** Verificación de validez de compra.
- **device\_type:** Tipo de dispositivo desde donde se hizo la reseña.
- **num\_helpful\_votes:** Número de votaciones de ayuda de la reseña por otros usuarios.
- **use\_age:** Edad de usuario que realizó la reseña.
- **user\_country:** País del usuario que hizo la reseña.
- **user\_gender:** Género del usuario que hizo la reseña.
- **app\_version:** Versión de la app desde donde se hizo la reseña.

## 2.3 Identificación de posibles problemas iniciales con los datos.

Se identifican posibles problemas iniciales en los datos, como la presencia de valores N/A (Not Available), los cuales deben ser tratados en el proceso de limpieza. Además, al ser un sistema basado en múltiples lenguajes, puede haber dificultades para establecer patrones consistentes que permitan detectar reseñas falsas, sin considerar aquellas generadas artificialmente mediante texto de relleno como Lorem Ipsum.

## 3. Análisis Exploratorio de Datos (EDA) Básico

### Importación de librerías.

```
library(ggplot2)
library(dplyr)
library(tidyverse)
```

### 3.1.1 Importación de dataset.

```
data_reviews <- read.csv("multilingual_mobile_app_reviews_2025.csv")
```

### 3.1.3 Identificar cantidad de filas y columnas.

```
dim(data_reviews)
```

```
## [1] 2514 15
```

El dataset presenta 2514 fila con 15 columnas.

### 3.2.2 Calcular la media, mediana y desviación estándar de las variables numéricas principales.

```
cat("=== Estadísticas de rating ===\n")
```

```
## === Estadísticas de rating ===
```

```
cat("Media: ", mean(data_reviews$rating, na.rm = TRUE), "\n")
```

```
## Media: 3.021034
```

```
cat("Mediana: ", median(data_reviews$rating, na.rm = TRUE), "\n")
```

```
## Mediana: 3
```

```
cat("Desviación estándar: ", sd(data_reviews$rating, na.rm = TRUE), "\n\n")
```

```
## Desviación estándar: 1.149955
```

```
cat("=== Estadísticas de user_age ===\n")
```

```
## === Estadísticas de user_age ===
```

```
cat("Media: ", mean(data_reviews$user_age, na.rm = TRUE), "\n")
```

```
## Media: 44.24781
```

```
cat("Mediana: ", median(data_reviews$user_age, na.rm = TRUE), "\n")
```

```
## Mediana: 44
```

```
cat("Desviación estándar: ", sd(data_reviews$user_age, na.rm = TRUE), "\n\n")
```

```
## Desviación estándar: 18.37229
```

```
cat("=== Estadísticas de num_helpful_votes ===\n")
```

```
## === Estadísticas de num_helpful_votes ===
```

```
cat("Media: ", mean(data_reviews$num_helpful_votes, na.rm = TRUE), "\n")
```

```
## Media: 616.7041
```

```
cat("Mediana: ", median(data_reviews$num_helpful_votes, na.rm = TRUE), "\n")
```

```
## Mediana: 620
```

```
cat("Desviación estándar: ", sd(data_reviews$num_helpful_votes, na.rm = TRUE), "\n")
```

```
## Desviación estándar: 363.7453
```

### 3.2.3 Obtener frecuencias de variables categóricas.

```

categoricas <- c("app_name", "app_category", "review_language",
                "verified_purchase", "device_type",
                "user_gender", "user_country")

for (var in categoricas) {
  cat("\n=== Frecuencias de", var, "===\n")

  freq <- data_reviews %>%
    count(.data[[var]]) %>%
    mutate(prop = n / sum(n))

  print(freq)
}

```

```

##
## === Frecuencias de app_name ===
##      app_name    n    prop
## 1  Adobe Photoshop 68 0.02704853
## 2      Airbnb 65 0.02585521
## 3      Amazon 54 0.02147971
## 4  Booking.com 62 0.02466189
## 5      Bumble 56 0.02227526
## 6      Canva 57 0.02267303
## 7    Coursera 59 0.02346858
## 8    Discord 44 0.01750199
## 9    Dropbox 70 0.02784407
## 10   Duolingo 57 0.02267303
## 11   Facebook 52 0.02068417
## 12  Google Drive 76 0.03023071
## 13  Google Maps 63 0.02505967
## 14  Grammarly 66 0.02625298
## 15   Instagram 72 0.02863962
## 16  Khan Academy 54 0.02147971
## 17   LinkedIn 56 0.02227526
## 18      Lyft 55 0.02187749
## 19    MX Player 72 0.02863962
## 20 Microsoft Office 68 0.02704853
## 21    Netflix 62 0.02466189
## 22   OneDrive 74 0.02943516
## 23    PayPal 61 0.02426412
## 24   Pinterest 80 0.03182180
## 25    Reddit 80 0.03182180
## 26    Signal 56 0.02227526
## 27   Snapchat 51 0.02028640
## 28    Spotify 51 0.02028640
## 29   Telegram 68 0.02704853
## 30    TikTok 63 0.02505967
## 31    Tinder 62 0.02466189
## 32   Twitter 59 0.02346858
## 33     Uber 64 0.02545744
## 34    Udemy 52 0.02068417
## 35     VLC 49 0.01949085
## 36    Venmo 50 0.01988862

```

```

## 37          Waze 64 0.02545744
## 38      WhatsApp 59 0.02346858
## 39          YouTube 66 0.02625298
## 40          Zoom 60 0.02386635
## 41          eBay 57 0.02267303
##
## === Frecuencias de app_category ===
##          app_category  n      prop
## 1          Business 150 0.05966587
## 2      Communication 137 0.05449483
## 3          Dating 140 0.05568815
## 4          Education 136 0.05409706
## 5      Entertainment 167 0.06642800
## 6          Finance 128 0.05091488
## 7          Games 117 0.04653938
## 8      Health & Fitness 155 0.06165473
## 9      Music & Audio 152 0.06046142
## 10      Navigation 161 0.06404137
## 11      News & Magazines 133 0.05290374
## 12      Photography 109 0.04335720
## 13      Productivity 140 0.05568815
## 14      Shopping 137 0.05449483
## 15      Social Networking 139 0.05529037
## 16      Travel & Local 159 0.06324582
## 17      Utilities 115 0.04574383
## 18 Video Players & Editors 139 0.05529037
##
## === Frecuencias de review_language ===
##      review_language  n      prop
## 1          ar 108 0.04295943
## 2          da 105 0.04176611
## 3          de 102 0.04057279
## 4          en 99 0.03937947
## 5          es 119 0.04733492
## 6          fi 111 0.04415274
## 7          fr 90 0.03579952
## 8          hi 92 0.03659507
## 9          id 111 0.04415274
## 10         it 100 0.03977725
## 11         ja 94 0.03739061
## 12         ko 114 0.04534606
## 13         ms 88 0.03500398
## 14         nl 116 0.04614161
## 15         no 103 0.04097056
## 16         pl 122 0.04852824
## 17         pt 93 0.03699284
## 18         ru 134 0.05330151
## 19         sv 102 0.04057279
## 20         th 97 0.03858393
## 21         tl 114 0.04534606
## 22         tr 102 0.04057279
## 23         vi 100 0.03977725
## 24         zh 98 0.03898170
##

```

```

## === Frecuencias de verified_purchase ===
##   verified_purchase    n      prop
## 1                False  575 0.2287192
## 2                 True 1939 0.7712808
##
## === Frecuencias de device_type ===
##   device_type    n      prop
## 1      Android  512 0.2036595
## 2 Android Tablet 509 0.2024662
## 3 Windows Phone 522 0.2076372
## 4         iOS   498 0.1980907
## 5        iPad   473 0.1881464
##
## === Frecuencias de user_gender ===
##   user_gender    n      prop
## 1                587 0.2334924
## 2             Female 495 0.1968974
## 3              Male 482 0.1917263
## 4        Non-binary 486 0.1933174
## 5 Prefer not to say 464 0.1845664
##
## === Frecuencias de user_country ===
##   user_country    n      prop
## 1                41 0.01630867
## 2      Australia 128 0.05091488
## 3    Bangladesh  95 0.03778839
## 4       Brazil   96 0.03818616
## 5       Canada   89 0.03540175
## 6        China  100 0.03977725
## 7       France   92 0.03659507
## 8       Germany 119 0.04733492
## 9        India  109 0.04335720
## 10    Indonesia 114 0.04534606
## 11       Italy  112 0.04455052
## 12       Japan  103 0.04097056
## 13    Malaysia 108 0.04295943
## 14       Mexico 111 0.04415274
## 15      Nigeria 109 0.04335720
## 16    Pakistan   94 0.03739061
## 17  Philippines  91 0.03619730
## 18       Russia  97 0.03858393
## 19   South Korea  94 0.03739061
## 20        Spain  99 0.03937947
## 21    Thailand   91 0.03619730
## 22       Turkey 115 0.04574383
## 23 United Kingdom 107 0.04256165
## 24  United States  89 0.03540175
## 25      Vietnam 111 0.04415274

```

### 3.3.1 Contar cuántos valores faltantes hay por variable (sin necesidad de imputarlos todavía).

```
null_data <- colSums(is.na(data_reviews))
print(null_data)
```

```
##      review_id      user_id      app_name      app_category
##           0           0           0           0
##      review_text  review_language      rating      review_date
##           0           0           37           0
## verified_purchase      device_type num_helpful_votes      user_age
##           0           0           0           0
##      user_country      user_gender      app_version
##           0           0           0
```

## 4.Revisión bibliográfica.

### 4.2 Métodos usados en investigaciones similares.

La literatura reciente (2022–2025) muestra una evolución clara en los enfoques utilizados para detectar reseñas falsas en múltiples idiomas.

De todos los métodos, destacan tres como los más recurrentes y efectivos:

---

#### 1. Transformadores multilingües (ej. XLM-R, mBERT, MuRIL)

- **Qué son:** modelos basados en *transformers* entrenados en decenas de lenguas, capaces de generar representaciones contextuales universales.
- **Cómo funcionan:** procesan el texto directamente en su idioma original, evitando pérdidas de señal al traducir.
- **Ventajas:**
  - Capturan semántica y estilo en paralelo.
  - Permiten *transfer learning* entre idiomas.
  - Suelen liderar los benchmarks multilingües.
- **Limitaciones:**
  - Requieren bastante cómputo.
  - Sesgos culturales en idiomas minoritarios.

## 2. Modelos híbridos (texto + metadatos de usuario/comportamiento)

- **Qué son:** arquitecturas que combinan análisis del texto con señales externas (frecuencia de reseñas, timestamps, historial del usuario, verified purchase).
  - **Cómo funcionan:** el texto se procesa con embeddings o transformers, y se fusiona con features tabulares en un clasificador (p. ej. redes neuronales, XGBoost).
  - **Ventajas:**
    - Más robustos ante reseñas generadas por IA.
    - Permiten detectar patrones de spam organizados.
  - **Limitaciones:**
    - No siempre hay metadatos disponibles.
    - Requieren integración de distintas fuentes de datos.
- 

## 3. Métodos basados en grafos (Graph Neural Networks)

- **Qué son:** técnicas que representan la relación entre usuarios, productos y reseñas como un grafo.
- **Cómo funcionan:** los nodos (usuarios, reseñas, productos) se conectan, y la red neuronal de grafos aprende a detectar patrones sospechosos de colusión.
- **Ventajas:**
  - Muy útiles para grandes plataformas (Amazon, Yelp, TripAdvisor).
  - Detectan redes organizadas de spam que otros modelos pasan por alto.
- **Limitaciones:**
  - Necesitan datos estructurados a gran escala.
  - Costosos de implementar en entornos pequeños.

## 5. Plan de trabajo.

### 5.1 Metodología general a seguir.

## Metodología Propuesta (Modelo Probabilístico Naive Bayes)

### 1. Diseño de la investigación

El proyecto se enmarca en un enfoque **experimental de clasificación supervisada**, cuyo propósito es identificar reseñas falsas en un contexto multilingüe. Se selecciona el modelo **Naive Bayes**, dado que es un clasificador probabilístico eficiente y ampliamente usado en el análisis de texto.



---

## 2. Conjunto de datos

### 2.1 Dataset entregado

- Se utiliza el conjunto de reseñas multilingües proporcionado por el curso/proyecto.
- Dicho dataset no cuenta con etiquetas claras de “falsa” o “real”, lo cual requiere una fase de categorización manual.

### 2.2 Categorización de reseñas

- **Criterios lingüísticos:** uso excesivo de adjetivos extremos, frases genéricas, repeticiones sospechosas.
- **Criterios contextuales:** incongruencias entre calificación y texto, alta frecuencia de reseñas en poco tiempo, comentarios poco específicos.
- **Proceso de anotación:**
  - El equipo investigador categorizará manualmente un subconjunto de reseñas.
  - Para mayor confiabilidad, se aplicará una anotación por pares y se medirá la concordancia inter-anotador.
  - El conjunto resultante servirá como base para el entrenamiento y prueba del modelo.

---

## 3. Preparación de los datos

- Normalización del texto (minúsculas, eliminación de caracteres especiales).
- Tokenización por idioma.
- Eliminación de *stopwords*.
- Representación de características mediante **Bolsa de Palabras (BoW)** o **TF-IDF**, según el escenario de prueba.

---

## 4. Modelo de clasificación

### 4.1 Algoritmo seleccionado

- Se emplea el **Naive Bayes Multinomial**, adecuado para tareas de clasificación de texto.
- Estima la probabilidad de que una reseña pertenezca a la clase *real* o *falsa* en función de la frecuencia de las palabras.

## 4.2 Parámetro de suavizado

- Se aplicará **suavizado de Laplace** ( $\alpha$ ) para manejar el problema de palabras no observadas en el entrenamiento.
- 

## 5. Estrategia de experimentación

- División de datos en 70% entrenamiento, 15% validación y 15% prueba.
  - Garantizar un balance razonable entre reseñas falsas y reales.
  - Escenarios de prueba:
    - **Monolingüe**: entrenamiento y prueba en un mismo idioma.
    - **Multilingüe combinado**: mezcla de todos los idiomas.
- 

## 6. Métricas de evaluación

- **Accuracy**: desempeño general del modelo.
  - **Precision y Recall**: capacidad para identificar correctamente reseñas falsas sin exceso de falsos positivos.
  - **F1-score**: equilibrio entre precisión y exhaustividad.
  - **Matriz de confusión**: análisis de errores.
  - **Cohen's Kappa**: medir confiabilidad entre anotadores en la fase de categorización.
- 

## 7. Análisis de resultados

- Comparar los resultados entre escenarios monolingüe y multilingüe.
  - Identificar patrones en los errores comunes.
  - Evaluar la suficiencia de los criterios manuales de categorización y su impacto en el rendimiento del modelo.
-

## 8. Aporte esperado

- Construcción de un dataset anotado de reseñas reales y falsas en múltiples idiomas.
- Validación de la aplicabilidad del modelo **Naive Bayes** en la detección de reseñas falsas multilingües.
- Discusión sobre la eficacia de un enfoque probabilístico clásico frente a problemas modernos de detección de fraude en reseñas.

## 5.3 Herramientas y lenguajes de programación.

### Lenguaje R

R como lenguaje cumple un factor decisivo para la realización del proyecto, sobre todo para el análisis de datos. Según su definición R es una sistema para la computación estadística y gráficas enfocado a un lenguaje de programación para generación de gráficas optimizadas y limpieza de errores[1].

Además de su diversidad de elementos que ofrece para el tratamiento de los datos según las necesidades del proyecto, consolidado por su vasto dominio de paquetes que tienen dominios como la estadística, aprendizaje automatizado, visualización de datos y análisis de series de tiempo definido bajo una red comprensiva de archivos (CRAN) con alrededor 18000 librerías desarrollada por la comunidad[2].

### RStudio

Posterior al entendimiento del lenguaje R, destacamos que Rstudio como editor de código abierto, que nos permite escribir citaciones, visualizar la generación de código y el comportamiento de las variables de datos involucradas[3].

Con características destacables como su consola, editor de scripts, un ambiente de panel que permite observar las variables afectadas en tiempo real, panel de historia que muestra los comando ejecutados y un espacio de proyectos para guardar el trabajo en conjunto a la solución de un proyecto[2].

### RMarkdown

Por último, para documentación del proceso de tratamiento de datos por la identificación de spam por medio de la ecuación Naive Bayes en reseñas, nos permite formatear las sustentaciones del código por medio de markdown para indicar la estética de los textos implícitos en los proceso escrito del proyecto, diferente a aplicaciones como word[4].

Facilitado por su portabilidad, y cambios de las aplicaciones de edición en código. RMarkdown es una herramienta de valor para el ámbito investigativo para la preservación de conocimiento en el tiempo con adaptación estandarizada que ofrece markdown para libros, tesis de universidades[4], etc.

## 6. Expectativas y retos.

### 6.1 Posibles dificultades y estrategias para resolverlas.

*Dificultad:* Poca claridad de palabras claves que nos permiten identificar un comportamiento de spam para una contraseña.

*Estrategia:* Investigación de proyectos previos que nos indiquen que palabras clases influyen en las reseñas podrían ser consideradas spam o no.

*Dificultad:* Presencia de datos nulos en la variable review\_text que no faciliten el procesamiento de clasificación de las reseñas, considerado como un dato atípico que puede entorpecer los cálculos de probabilidad o verosimilitud de spam.

*Estrategia:* Limpieza de datos de datos que sean nulos, para no permitir el entorpecimiento del cálculo de palabras claves para Naive Bayes.

*Dificultad:* Presencia de otras variables que tengan relación con el comportamiento de spam presentado en el dataset.

*Estrategia:* Identificación relación entre variables por medio de modelos estadísticos complementarios que nos permitan definir relaciones con el spam.

*Dificultad:* Definir porcentaje de éxito de clasificación spam por medio de la ecuación Naive Bayes.

*Estrategia:* Definición de KPI o parámetro evaluables que midan el estado del arte del modelo para identificar spam.

*Dificultad:* Presencia de valores atípicos diferentes review\_text que afecten la detección de spam.

*Estrategia:* Definición de procesos de análisis de relación (sea de cualitativo o cuantitativo) que definan su efecto en la generación de reseñas de spam.

## 7. Conclusión preliminar

### 7.1 Resumen del planteamiento del problema y hallazgos iniciales.

Aplicar la ecuación de Naive Bayes para la clasificación de spam en reseñas de las aplicaciones móviles principales en el mundo, por otra parte solucionar problemas con la claridad de los datos dentro del dataset para evitar la dispersión en la clasificación de reseñas y del mismo modo encontrar algún tipo de relación con las demás variables presentables dentro del dataset que puedan influir en el proceso de clasificación.

Establece el reconocimiento de la variables, además de la que compone a reseña como números identificativos del usuario, edad de los usuarios, sistema operativo, tipo de dispositivo, categoría de aplicación entre otros que pueden ser de ayuda para ver una relación entre la variables y su comportamiento con respecto a la reseñas.

Identificar un conjunto de dificultades como datos nulos, atípicos y medidas evaluativas de efectividad para proponer un serie de estrategias que nos permiten medir el alcance de funcionamiento del sistema para discernir el impacto de la reseña de carácter verdadero y de spam en el mundo real y su impacto en el rendimiento de los servicios aplicativos.

Comprender la sustentación técnica como lenguajes de programación con enfoque estadísticos, que nos permitan aplicar la ecuación Naive Bayes con gráficas estadísticas que para fácil visualización, además de un gran complejo de librerías de código abierto que facilitan el sistema de análisis de desarrollo.

## Referencias

- [1] GeeksforGeeks, “Pros and Cons of R Programming Language,” GeeksforGeeks, Jul. 23, 2025. <https://www.geeksforgeeks.org/r-language/pros-and-cons-of-r-programming-language/>
- [2] R Core Team, “R Language Definition.” <https://cran.r-project.org/doc/manuals/r-release/R-lang.html>
- [3] GeeksforGeeks, “RStudio Tutorial,” GeeksforGeeks, Jul. 23, 2025. <https://www.geeksforgeeks.org/r-language/rstudio-tutorial/>
- [4] “Getting started | Markdown Guide.” <https://www.markdownguide.org/getting-started/>
- [5] M. I. Ragab, E. Hussein Mohamed, and W. Medhat, “Multilingual Propaganda Detection: Exploring Transformer-Based Models mBERT, XLM-RoBERTa, and mT5,” ACL Anthology, 2025. <https://aclanthology.org/2025.nakbanlp-1.9>

- [6] Anonymous, “A deep learning approach for detecting fake reviewers: Exploiting reviewing behavior and textual information,” *Decision Support Systems*, vol. 166, p. 113911, 2023. <https://www.sciencedirect.com/science/article/abs/pii/S0167923622001828>
- [7] L.-C. Cheng, Y. T. Wu, C.-T. Chao, and J.-H. Wang, “Detecting fake reviewers from the social context with a graph neural network method,” *Decision Support Systems*, 2024. <https://ouci.dntb.gov.ua/en/works/4yXaNdp7>
- [8] Anonymous, “Detecting review fraud using metaheuristic graph neural networks,” *International Journal of Information Technology*, 2024. <https://link.springer.com/article/10.1007/s41870-024-01896-w>