Proyecto - Filtro de reseñas.

Estefany Carolina Segura Linares
Brayan David Prieto Aya
Alejandro Jaramillo Vallejo

2025-10-28

1. Introducción

- Breve presentación del proyecto.
- Contextualización del problema y su relevancia.

La influencia del internet ha cambiado la dinamicas de la relevancia de un servicio, las personas denominados en un rol de usuarios buscan la buena imagen a la hora descargar algun tipo de aplicación por medio de las reseñas que validan la experiencia del acceso a este mismo para tomar una decisión, en este caso las reseñas positivas pueden influir en prestigio asociado al negocio.

Por eso, se pueden dar sponsors de "opinion spamming" donde se patrocinan reseñas fraudulentas para promover aplicaciones o en caso de compotencia desleal, para desvaluar una aplicación de otro negocio por el área de un servicio.

De esta manera, aplicar sistemas de mineria de datos para evaluar no veridicas reseñas positivas o en orden de probar un competencia justa dentro del negocio de las aplicaciones que se quieren dañar su reputación, por esto mismo genera procesos de datos que permiten dar una resolución real a los usuarios o identificar que tipo de estructuras se usan en ese tipo de practicas.

2. Justificación

La importancia del estudio de los datos en este problema radica en comprender a fondo qué variables son realmente relevantes para el sistema de clasificación. No todas las características presentes en el conjunto de datos aportan información útil, y un análisis preliminar permite identificar aquellas que influyen directamente en la detección y clasificación de las reseñas. Este proceso no solo evita el ruido dentro del modelo, sino que también mejora la precisión y eficiencia del algoritmo, al centrarse en los patrones lingüísticos y semánticos más representativos. Además, el estudio de los datos permite detectar sesgos, inconsistencias o distribuciones desbalanceadas, factores que pueden comprometer la capacidad del sistema para generalizar correctamente ante nuevas reseñas. En conjunto, este análisis es un paso fundamental para garantizar la calidad, coherencia y validez del modelo de clasificación.

- Valor agregado del análisis realizado.

3. Objetivos

- **General**: Enunciar el objetivo principal del proyecto.

Generar un sistema de mineria de datos, que aplica el algoritmo de Naive Bayes para determinar y/o de reseñas spam multilenguaje para aplicaciones.

- **Específicos**: Al menos tres objetivos que detallen las metas técnicas del análisis.

Identificar las palabras o términos con mayor peso predictivo dentro de las reseñas SPAM y no-SPAM, aportando interpretabilidad al modelo.

Analizar los patrones de comportamiento de los usuarios que generan reseñas SPAM, identificando tendencias según el sistema operativo y otras variables demográficas o técnicas relevantes.

Diseñar un dashboard interactivo que permita visualizar la tendencia de reseñas en función de la app seleccionada.

4. Fases del Proceso KDD (documentadas según su aplicación)

4.1 Dominio del problema

- Describir el contexto del fenómeno o situación a analizar.

Dentro de la competición digital que se presenta hoy dentro de los negocios digitales, las reseñas se han vuelto una forma bastante importante para acceder a un servicio digital(app), siendo un actividad tan importante para la posición de marca de un negocio que opta por este medio las reseñas determinan su puesto frente a la competenecia y el mercado.

Debido a esto se puede presentar situaciones donde la reseñas son patrocinadas en una inflación por spam para posicionar la marca de app o utilizar la tecnica para desprestigiar a los competidores genera una duda de la veracidad de los servicios digitales.

El modelo presente busca evaluar y predecir que tipo de contenido en reseña multilenguaje determina que una reseña en una aplicación es de tipo spam, por medio del algoritmo de Naive Bayes con su especialización en clasificar nos permite discernir en el comportamiento de competenecia desleal en linea.

- Formular preguntas de investigación o hipótesis que orienten la minería de datos.
- Identificar la relevancia del problema y su impacto en la toma de decisiones.

La relevancia del problema se centra en la necesidad de asegurar la autenticidad de las reseñas en plataformas digitales. La presencia de reseñas falsas o SPAM afecta la credibilidad de los sistemas de valoración y puede distorsionar la percepción que los usuarios tienen sobre un producto o servicio.

Detectar y clasificar adecuadamente este tipo de reseñas permite mantener la confianza del usuario y proteger la reputación de las aplicaciones. Además, contribuye a generar entornos digitales más transparentes y fiables para la toma de decisiones.

El impacto en la toma de decisiones es significativo, ya que los desarrolladores y administradores pueden identificar patrones anómalos, optimizar estrategias de marketing y mejorar la calidad de sus servicios. En conjunto, el análisis de reseñas se convierte en una herramienta clave para fortalecer la gestión y credibilidad de las plataformas.

4.2 Selección de Datos

- Describir la fuente de los datos (base utilizada, variables disponibles).
- Seleccionar las variables relevantes y justificar su elección.

Para la alimentación y elección del dataset se tomaran las variables - **review_text** : El contenido de texto de la reseña asociada a la aplicación que contiene palabras o mensajes que indiquen spam.

- **rating** : Puntuación de la aplicación para valoración si presenta algun tipo de inflación.
- **verified_purchase**: Variable que verifica si una reseña ha sido verificada por validez, tal vez las cantidades esten infladas para el posicionamiento de la aplicación.
- Indicar las variables eliminadas (por redundancia, irrelevancia o datos faltantes excesivos).

Las variables eliminadas fueron:

- **review_id**: Identificador único de cada reseña; no aporta información útil al modelo, ya que su función es únicamente de indexación.
- **user_id**: Identificador único del usuario; no contribuye a la clasificación ni al análisis, y su inclusión puede generar ruido o riesgo de sesgo por usuario.
- **num_helpful_votes**: Número de votos útiles que recibió una reseña; es un valor externo al contenido y puede estar influenciado por la visibilidad o popularidad, no por la autenticidad del texto.
- **user_gender**: Género del usuario; presenta una proporción significativa de datos faltantes y no tiene un impacto directo sobre el contenido o tipo de reseña.
- **app_version**: Versión de la aplicación reseñada; no afecta el análisis semántico ni la clasificación de reseñas, por lo que no es relevante para el modelo.

4.3 Limpieza de Datos

- Valores faltantes: especificar la estrategia usada (eliminación, imputación, estimación).

Imputación de **User_country** Para este caso, se decidió imputar el país en función del idioma de la reseña, utilizando una relación directa entre review_language y el país donde dicho idioma es predominante. Este método de imputación basada en reglas permite mantener la coherencia contextual de los datos, evitando introducir ruido aleatorio o sesgos significativos. Solo se aplicó la imputación cuando la asociación entre idioma y país era clara y dominante (por ejemplo, japonés con Japón, portugués con Brasil). En idiomas de uso global, como inglés o español, se evitó imputar para no sobrerrepresentar regiones.

Además de eso, se eliminaron los valores restantes no clasificados, que representaron alrededor de 20 campos, un valor ínfimo dentro del total del dataset.

Eliminación de valores faltantes en **user_rating** Los registros sin valoración fueron eliminados porque no aportan una medida cuantificable para el análisis o entrenamiento del modelo. Dado que representan un porcentaje muy bajo (alrededor del 1.2% de los datos totales), su eliminación no afecta de forma significativa la representatividad del conjunto.

Imputar estos valores habría introducido varianza artificial y podría distorsionar la distribución del objetivo principal del modelo

Eliminación de valores faltantes en **review_text** Las reseñas sin texto fueron eliminadas ya que no contienen información semántica para el análisis de texto o la clasificación basada en lenguaje natural. Aunque algunas aplicaciones permiten dejar una valoración sin comentario, en este contexto analítico el texto es esencial para los modelos de clasificación y detección de spam. El número de eliminaciones fue mínimo (aproximadamente otro 1.2%), por lo que esta decisión mejora la calidad del dataset sin comprometer su tamaño.

- Errores e inconsistencias: documentar correcciones, duplicados o errores tipográficos.

El conjunto de datos presentaba desde su origen un alto nivel de calidad y consistencia, sin requerir procesos de depuración significativos. Las variables se encontraban correctamente estructuradas, con tipos de datos apropiados, sin presencia de valores atípicos evidentes, duplicados ni errores de codificación.

En consecuencia, no fue necesario aplicar procedimientos adicionales de limpieza, imputación o transformación más allá de las comprobaciones básicas de integridad. La información se encontraba organizada y libre de anomalías, lo cual facilitó directamente su uso para las etapas de análisis exploratorio y modelado.

Este escenario permitió conservar la estructura original del dataset, garantizando la trazabilidad y autenticidad de los datos, manteniendo así la fidelidad respecto a su fuente y su interpretación analítica.

- Outliers: describir el método de detección (boxplot, z-score, etc.) y las decisiones tomadas.

4.4 Transformación de Datos

- Normalización o estandarización de variables numéricas.

En este caso no se aplicó ningún proceso de normalización o estandarización debido a que la única variable numérica relevante **rating** ya se encuentra expresada en una escala limitada y uniforme (0 a 5). Esta escala es inherentemente comparable y está diseñada para representar de forma directa el nivel de satisfacción del usuario, por lo que no requiere ajustes adicionales.

Además, el modelo seleccionado para el análisis, Naive Bayes, no depende de magnitudes o distancias entre variables, sino de distribuciones de probabilidad. Por tanto, modificar la escala de rating no aportaría mejoras en la capacidad predictiva del modelo ni en la consistencia de los resultados.

- Codificación de variables categóricas (one-hot o label encoding).
- Creación de nuevas variables derivadas relevantes para el modelo.
- Reducción de dimensionalidad, si aplica (PCA u otras técnicas).
- Mostrar fragmentos de código en R y comparaciones “antes y después” de las tablas.

4.5 Minería de Datos

- Seleccionar y justificar el algoritmo o técnica empleada (clasificación, regresión, clustering, etc.).
- Describir la división de datos (entrenamiento y prueba).
- Presentar las métricas de evaluación (Accuracy, F1-Score, MAE, etc.).
- Incluir visualizaciones que respalden los resultados del modelo.

4.6 Interpretación y Evaluación

- Analizar críticamente los resultados obtenidos.
- Validar el conocimiento descubierto frente a las hipótesis y objetivos planteados.
- Evaluar el valor del conocimiento extraído para el contexto del problema.

5. Conclusiones

- Sintetizar los principales hallazgos.
- Reflexionar sobre el proceso completo y sus limitaciones.
- Proponer posibles trabajos futuros o mejoras.

6. Anexos

- Gráficos, tablas, fragmentos de código, resultados adicionales que complementen el análisis.

```
library(readr)
library(dplyr)
```

```
##
## Adjuntando el paquete: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
multilingual_mobile_app_reviews_2025 <- read_csv("multilingual_mobile_app_reviews_2025.csv")
```

```
## Rows: 2514 Columns: 15

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (8): app_name, app_category, review_text, review_language, device_type,...
## dbl  (5): review_id, user_id, rating, num_helpful_votes, user_age
## lgl  (1): verified_purchase
## dttm (1): review_date
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
View(multilingual_mobile_app_reviews_2025)
```

```
# Contar valores vacíos en todo el dataset
colSums(is.na(multilingual_mobile_app_reviews_2025))
```

```
##         review_id           user_id          app_name      app_category
##                 0                 0                 0                 0
##       review_text   review_language            rating       review_date
##                59                 0                37                 0
## verified_purchase       device_type num_helpful_votes          user_age
##                 0                 0                 0                 0
##      user_country       user_gender       app_version
##                41               587                30
```

```
Copia_Datos_Limpios <- multilingual_mobile_app_reviews_2025
Copia_Datos_Limpios <- Copia_Datos_Limpios %>%
  select(-review_id, -num_helpful_votes, -user_id, -user_gender, -app_version)
colSums(is.na(Copia_Datos_Limpios))
```

```
##          app_name      app_category       review_text   review_language
##                 0                 0                59                 0
##            rating       review_date verified_purchase       device_type
##                37                 0                 0                 0
##          user_age      user_country
##                 0                41
```

```
# Filtro general para cualquier variable
filtro_user_country <- Copia_Datos_Limpios %>% filter(is.na(user_country))
filtro_review_text  <- Copia_Datos_Limpios %>% filter(is.na(review_text))
filtro_rating       <- Copia_Datos_Limpios %>% filter(is.na(rating))
```

```
Copia_Datos_Limpios <- Copia_Datos_Limpios %>%
  mutate(
    user_country = case_when(
      review_language == "es" ~ "Spain",          # Spanish
      review_language == "pt" ~ "Brazil",         # Portuguese
      review_language == "ja" ~ "Japan",          # Japanese
      review_language == "hi" ~ "India",          # Hindi
      review_language == "ko" ~ "South Korea",    # Korean
      review_language == "zh" ~ "China",          # Chinese
      review_language == "de" ~ "Germany",        # German
      review_language == "fr" ~ "France",         # French
      review_language == "it" ~ "Italy",          # Italian
      review_language == "ru" ~ "Russia",         # Russian
      TRUE ~ user_country                         # Keep original if no match
    )
  )

Copia_Datos_Limpios <- Copia_Datos_Limpios %>%
  filter(!is.na(rating))

Copia_Datos_Limpios <- Copia_Datos_Limpios %>%
  filter(!is.na(review_text))

Copia_Datos_Limpios <- Copia_Datos_Limpios %>%
  filter(!is.na(user_country))

colSums(is.na(Copia_Datos_Limpios)[, c("user_country", "review_text", "rating")])
```

```
## user_country  review_text       rating
##            0            0            0
```

```
sapply(Copia_Datos_Limpios,class)
```

```
## $app_name
## [1] "character"
##
## $app_category
## [1] "character"
##
## $review_text
```

```
## [1] "character"
##
## $review_language
## [1] "character"
##
## $rating
## [1] "numeric"
##
## $review_date
## [1] "POSIXct" "POSIXt"
##
## $verified_purchase
## [1] "logical"
##
## $device_type
## [1] "character"
##
## $user_age
## [1] "numeric"
##
## $user_country
## [1] "character"
```

```
# languajes <- Copia_Datos_Limpios %>% unique(review_language)
# colnames(Copia_Datos_Limpios)
```