UNIVERSIDAD DE CONCEPCIÓN

FACULTAD DE INGENIERÍA

DEPARTAMENTO DE INGENIERÍA INFORMÁTICA Y CIENCIAS DE LA COMPUTACIÓN

Análisis De Datos

Proyecto semestral:

ChatSentinel

Integrantes:

Javier Cadagán Parra
Gabriel Huerta Torres
Diego Oyarzo Navia

Fecha: 23 de Junio, 2025

Índice

1 Resumen Ejecutivo

ChatSentinel es un proyecto orientado a la detección automática de conversaciones de grooming online mediante técnicas de procesamiento de lenguaje natural y machine learning. El proyecto ha evolucionado desde un análisis exploratorio inicial hasta la implementación de un modelo de clasificación basado en 3 modelos para identificar a los participantes como 'predador' o 'victima' dentro de una conversación. Finalmente se eligió el modelo Regresión Logística por tener el mejor equilibrio entre la correcta identificación y la minimizan de falsas alarmas.

2 Introducción y Contexto

El grooming online representa una amenaza creciente en el entorno digital actual, donde individuos malintencionados establecen relaciones de confianza con menores de edad para fines de explotación. La detección temprana de estas interacciones es crucial para la protección infantil en espacios digitales.

2.1 Objetivos del Proyecto

Objetivo General: Desarrollar un sistema de detección automática de conversaciones de grooming mediante análisis de patrones lingüísticos y comportamentales de los participantes.

Objetivos Específicos:

- Analizar las características distintivas del lenguaje utilizado en conversaciones de grooming.
- Implementar un modelo de clasificación que distinga entre los roles de victima o groomer, basándose en su lenguaje.
- Evaluar la efectividad del modelo propuesto e identificar sus limitaciones.
- Proponer mejoras y líneas futuras de investigación.

2.2 Justificación

La automatización de la detección de grooming es esencial dado el volumen masivo de comunicaciones online y la limitación de recursos humanos para monitoreo manual. Este proyecto contribuye al desarrollo de herramientas de protección infantil basadas en machine learning.

3 Metodología y Datos

3.1 Descripción de los Conjuntos de Datos

3.1.1 Dataset Principal - ChatCoder 2.0

- **Fuente:** ChatCoder HomePage (obtenido mediante Wayback Machine ChatCoder Dataset, 2020)

- **Formato:** 56 archivos XML
- **Estructura:**
  - `user_id`: Identificador único del usuario
  - `datetime`: Fecha del mensaje.
  - `message`: Contenido textual del mensaje
  - `role`: Clasificación del usuario (PREDATOR/VICTIM)

3.1.2 Dataset Complementario - DailyDialog

- **Fuente:** Hugging Face platform
- **Propósito:** Debido a que necesitamos identificar que chats son considerados "grooming" y "no grooming", optamos por recurrir a un segundo data publico (DailyDialog) Li et al. (2017) set que contiene conversaciones cotidianas, calificando sus chats como"no grooming"), teniendo así un dataset con diálogos naturales sin contenido de grooming.

3.2 Procesamiento y Limpieza de Datos

3.2.1 Extracción de Datos para Dataset Principal

Se implementó un sistema de procesamiento XML utilizando la librería `xml.etree`, con la siguiente lógica:

```
def parse_chatlog(xml):
    # Lectura y parseo del contenido XML
    # Identificacion de usuarios PREDATOR y VICTIM
    # Extraccion de posts individuales
    # Asignacion de roles segun usuario
    # Construccion de DataFrame estructurado
```

3.2.2 Extracción de Datos para Dataset Secundario

Para este dataset de "no grooming"solamente se extrajeron las conversaciones, nos dimos cuenta que eran conversaciones de longitud muy pequeña pero de mucha cantidad con respecto al dataset principal, por lo que se tomó la decisión de concatenar varias conversaciones de 40 en 40, de manera que el promedio de caracteres por dato "full chat"se mantuviera similar [14.000 vs 17000 caracteres aproximadamente] y que la cantidad de full chats fuera similar en ambos dataset (47 vs 56), seleccionando una muestra aleatoria del 20 % para cumplir esta cuota. A continuación se muestra la lógica que se siguió para esta concatenación y así lograr un dataset parecido al principal en cuanto a longitud, con la siguiente lógica:

```
from datasets import load_dataset

# Lectura del dataset
# Normalizacion del texto
# Eliminacion de columnas que no fueron utilizadas
# Asignacion de columna label = 0 (No grooming)
```

```
# Concatenacion para lograr media y desviacion estandar parecidas
    a dataset principal
```

3.2.3 Limpieza y Normalización

Problemas Identificados y Soluciones:

1. **Caracteres Inválidos en XML:** Resolución mediante expresiones regulares para reemplazo de caracteres problemáticos en los mensajes
2. **Inconsistencias Temporales:** Estandarización al formato `%Y-%m-%d %H:%M:%S`
3. **Normalización Lingüística:** Se eliminaron mensajes vacíos y nulos para evitar afectar al modelo, como también se desarrollo la normalización de espacios y saltos de línea, la implementación de diccionario personalizado para slangs y abreviaciones, y como un apoyo a esto ultimo, se utilizo de la librería `contractions` complementada con expresiones regulares propias.

3.3 Enfoque Inicial de ML: Clasificación de Chats

Los mensajes de cada archivo XML del dataset ChatCoder fueron concatenados para formar un único documento por conversación, y ambos datasets fueron etiquetados como:

- **Clase 1 (Grooming):** Conversaciones del dataset ChatCoder que contienen grooming.
- **Clase 0 (No Grooming):** Diálogos del dataset DailyDialog.

Se realizó un análisis de longitudes para eliminar outliers extremos, de manera que se borraron los chats con mas de 40.000 caracteres, asegurando así la consistencia en el formato final.

3.3.1 Limitaciones del enfoque inicial

Tras la implementación y evaluación de este modelo (cuyos resultados se detallan en la sección 6), se identificaron fallos metodológicos que invalidaban su utilidad practica.

Los dos datasets, a pesar de normalizar su longitud, tenían estilos de conversación muy distintos en contenido, jerga y estructura. Por lo que el modelo, en vez de aprender a identificar detalles del grooming, tan solo aprendió a identificar características superficiales de las dos fuentes de datos, osea un modelo con métricas perfectas pero con mala capacidad de generalización. Estas limitaciones hicieron necesario un rediseño completo del problema.

3.4 Enfoque Final: Clasificación de Roles

Se redefinió el problema para centrarse en un único dataset consistente; Clasificar el rol de un usuario ('predator' o 'victim') basándose en su lenguaje. Todos los mensajes de un mismo usuario (`username`) dentro de una misma conversación (`file`) fueron concatenados en un único texto, cada uno de estos textos se convirtió en una muestra de datos independientes. La entrada (X) es el texto completo del usuario, y la etiqueta (Y) es su rol en la conversación.

- **Clase 1 (Predator)**
- **Clase 0 (Victim)**

Con esta metodología se elimina el problema de la disparidad entre los datasets y hay más equilibrio entre las clases, con la desventaja de tener una menor cantidad de muestras que utilizar.

4 Análisis Exploratorio Consolidado

4.1 Análisis Temporal

El análisis de patrones temporales reveló patrones significativos sobre el comportamiento de grooming:

Hallazgos Principales: Se puede apreciar una actividad mínima durante horas de madrugada (3-9h), como también un incremento gradual en la cantidad de mensajes desde las 10hrs, alcanzando un pico máximo a las 22hrs (>11,000 mensajes) y finalmente una baja en esta actividad nocturna a las 1am.

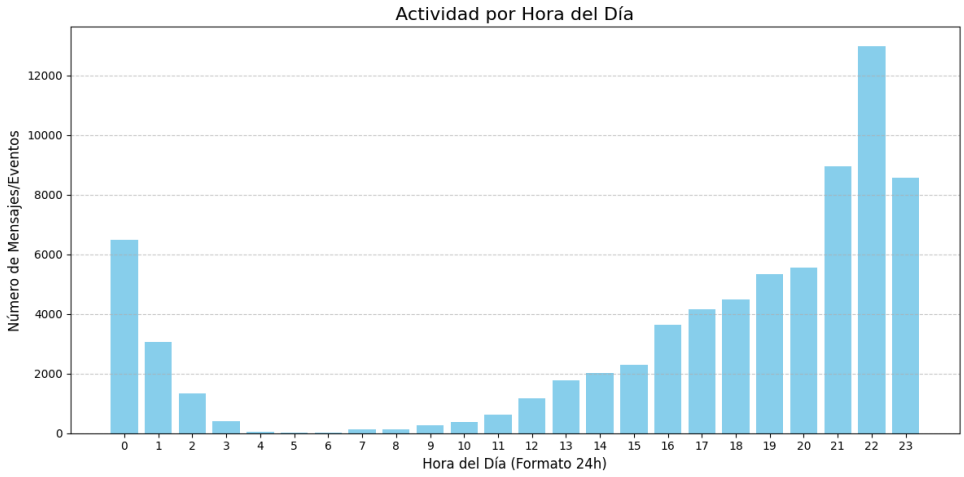

Figura 1: Actividad por hora del día del dataset principal

Interpretación: Los patrones coinciden con períodos de menor supervisión parental y mayor disponibilidad de menores online como se puede ver en la figura 1, sugiriendo una correlación entre horarios de vulnerabilidad y actividad de grooming.

4.2 Análisis Lingüístico

4.2.1 Longitud de Mensajes

Observaciones Clave:

- **Víctimas:** Tendencia a mensajes más cortos y frecuentes
- **Predadores:** Mayor variabilidad en longitud con abundantes outliers

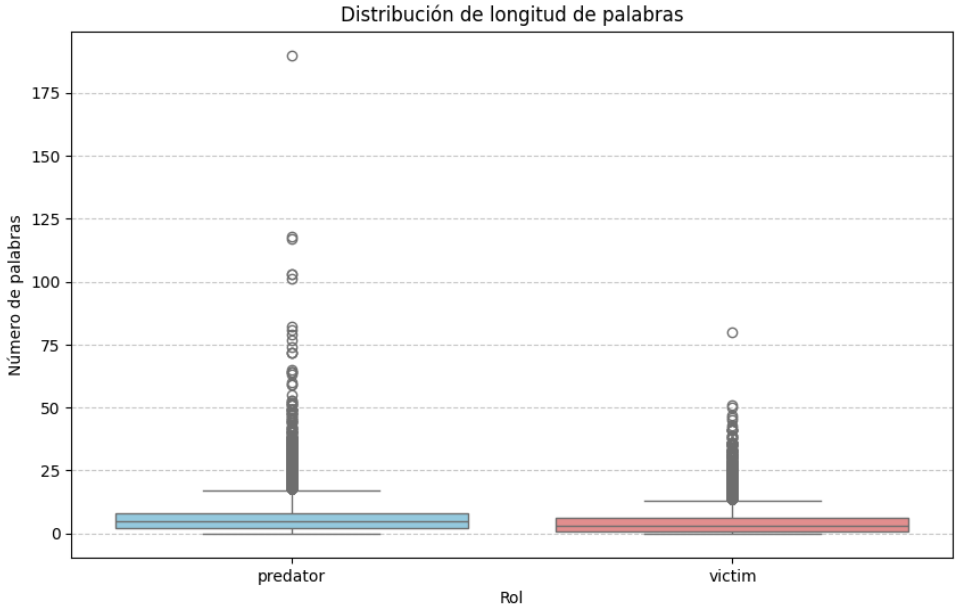

Figura 2: Actividad por hora del día del dataset principal

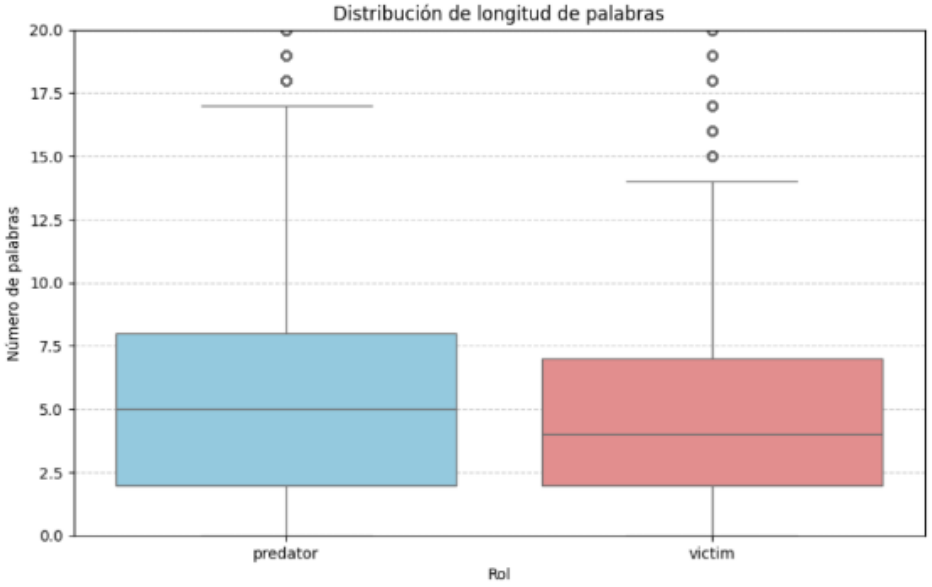

Figura 3: Actividad por hora del día del dataset principal sin outliers

Interpretación: Los predadores utilizan mensajes más largos, como podemos ver en 2 con la finalidad de manipular y obtener confianza.

4.2.2 Análisis Léxico (WordClouds)

La visualización mediante nubes de palabras mostró diferencias significativas en el vocabulario utilizado por cada rol como se puede ver en , validando la hipótesis de que existen patrones lingüísticos distintivos detectables computacionalmente.

Figura 4: Wordcloud de los mensajes de los groomers

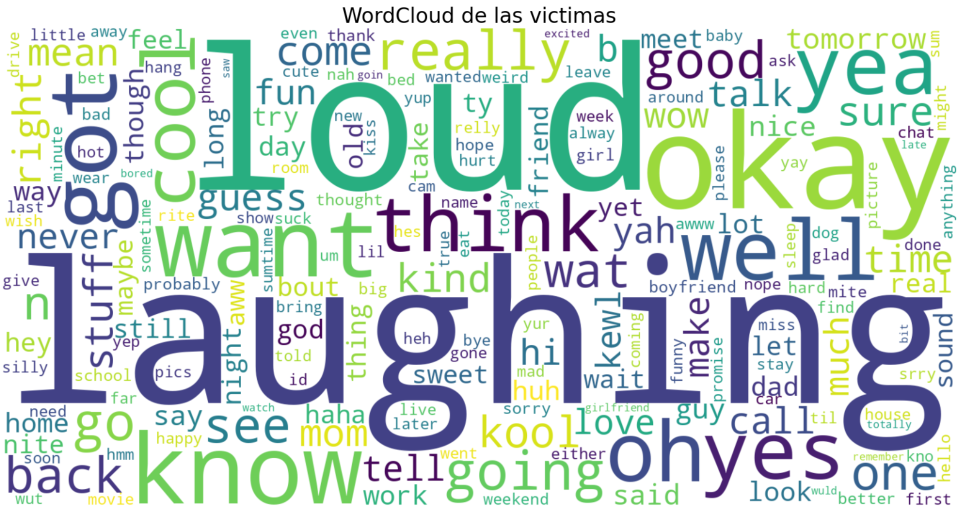

Figura 5: Wordcloud de los mensajes de las victimas

5 Modelo de Clasificación

Tras identificar las limitaciones del enfoque inicial, el proyecto se centró en el desarrollo de un modelo para clasificar el rol de un participante basandose en el conjunto de todos sus mensajes.

5.1 Selección del Modelo

Modelos Evaluados: Regresión Logística, Support Vector Machine y Random Forest.
Justificación técnica:

- **Regresión Logística:** Se eligió por su eficiencia, interpretabilidad y simpleza, además con sus coeficientes podemos analizar fácilmente los patrones lingüísticos de cada rol.

- **Support Vector Machine:** Se incluyeron para explorar modelos con mayor capacidad de capturar relaciones no lineales y comparar opciones.

5.2 Ingeniería de Características

Para transformar el texto crudo a un formato numérico que pueda ser procesado por los modelos, se construyó un `Pipeline` de Scikit-learn. Esto hace que siempre se apliquen pasos de preprocesamiento tanto en el entrenamiento como en la prueba.

1. **Vectorización TF-IDF:** El texto de cada participante se convirtió en un vector numérico utilizando `TF-IDF Vectorizer`, que pondera la importancia de las palabras por su frecuencia y rareza. Algunos de sus parámetros clave son:
   - **stop_words='english':** Elimina palabras comunes sin mucho valor.
   - **ngram_range=(1, 2):** Para capturar palabras individuales (unigramas) y pares de palabras adyacentes (bigramas).
   - **max_features=10000:** Limita el vocabulario a las 10.000 características mas relevantes para controlar la dimensionalidad.
2. **Clasificador:** Segundo paso del pipeline, aplica el modelo elegido.

5.3 Configuración de Entrenamiento

Se implemento una estrategia de validación para evitar la fuga de datos:

- **Estrategia de División:** Se utilizó `GroupShuffleSplit` de Scikit-learn para asegurarse que los datos de una misma conversación (`file`) estuvieran agrupados en el mismo conjunto.
- **Ratio de División:** Se asignó un 65 % de las conversaciones al conjunto de prueba y el 35 % restante al de entrenamiento.
- **Justificación:** Esta metodología asegura que todos los participantes (predadores y víctimas) de una misma conversación terminen en el mismo conjunto (entrenamiento o prueba). Esto previene que el modelo .ªprenda"sobre una conversación durante el entrenamiento y luego sea probado en otro participante de esa misma conversación, lo que inflaría las métricas. El modelo se evalúa sobre conversaciones completamente invisibles, simulando un escenario real.

6 Resultados y Evaluación Crítica

6.1 Modelo Inicial (Clasificación de chats)

El modelo inicial, entrenado para diferenciar entre los datasets ChatCoder y DailyDialog, arrojo los siguientes resultados en el conjunto de prueba: **Resultados Obtenidos:**

Métrica	Clase 0 (No Grooming)	Clase 1 (Grooming)
Precision	1.00	1.00
Recall	1.00	1.00
F1-Score	1.00	1.00

Cuadro 1: Métricas de desempeño por clase

- **Accuracy General:** 1.00 (100 %)

Distribución del Conjunto de Prueba:

- Clase 0 (No Grooming): 11 ejemplos
- Clase 1 (Grooming): 10 ejemplos

6.1.1 Análisis de Resultados

Aunque aparentemente se ven perfectos, esto indica que el modelo es defectuoso. La perfección de los resultados se debe a un sobreajuste severo a la fuente de los datos, el modelo no aprendió a detectar grooming, más bien aprendió a distinguir el estilo de conversación entre un chat de internet de 2006 de gran longitud, con conversaciones cotidianas y cortas de 2017. Por esto se determino en abandonar el enfoque de mezclar datasets y redefinir el problema para operar dentro del dataset original.

6.2 Modelo Final (Clasificación de Roles)

El segundo y definitivo enfoque del proyecto realiza la clasificación respecto al rol del participante ('predator' o 'victim'). Así elimina los problemas de sesgo y permite la evaluación de patrones lingüísticos relevantes.

Modelo	Precision (Predator)	Recall (Predator)	F1-Score (Predator)
Regresión Logística	0.84	**0.90**	**0.87**
Support Vector Machine	**0.91**	0.75	0.82
Random Forest	0.67	0.92	0.78

Cuadro 2: Métricas comparativas de los modelos para la clasificación de roles. Se destaca el rendimiento en la clase 'Predator'.

6.2.1 Análisis de Resultados:

- **Regresión Logística (LR):** Se posiciona como el modelo más balanceado y efectivo, logrando el F1-Score más alto (0.87). Su alto recall (0.90) minimiza el riesgo de no detectar a los predadores.
- **Support Vector Machine (SVM):** Ofrece la mayor precisión (0.91), pero a costa de un recall bajo (0.75). Este modelo es demasiado conservador y deja sin identificar a un 25 % de los predadores (falsos negativos).
- **Random Forest (RF):** Alcanza el recall más alto, pero con una precisión muy pobre, por lo que tiene una alta tasa de falsas alarmas.

6.3 Matrices de Confusión

Las matrices de confusión visualizan la distribución de los errores. El error más crítico es el Falso Negativo (esquina inferior izquierda), que representa un groomer no detectado.

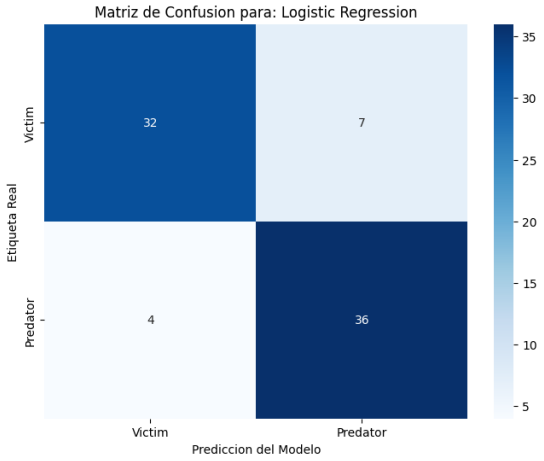

Figura 6: Matriz de Confusión para Regresión Logística. Muestra el mejor equilibrio, con solo 4 Falsos Negativos.

6.4 Selección del Modelo Ganador

Basado en los resultados, la **Regresión Logística** es seleccionado como el modelo final para ChatSentinel. Al tener alto recall y buena precisión se logra maximizar la detección de amenazas (recall) sin generar una cantidad excesiva de falsas alarmas (precision).

7 Propuesta de Mejora

El modelo actual ha mostrado buenos resultados y tiene buen rendimiento en la clasificación de roles. Sin embargo, nuestra población de chats es muy pequeña para realizar mejor experimentación, las siguientes lineas de trabajo se proponen para mejorar algunas limitaciones actuales:

7.1 Mejoras Inmediatas sobre el Modelo Actual

Estas mejoras se pueden implementar directamente sobre el pipeline existente para optimizar su rendimiento.

7.1.1 Ajuste del Umbral de Decisión

El modelo actual utiliza un umbral de probabilidad por defecto de 0.5 para clasificar a un usuario. Dado que el coste de un falso negativo (un predador no detectado) es extremadamente alto, se podría ajustar este umbral.

- **Estrategia:** Analizar la curva Precision-Recall para seleccionar un umbral que maximice el recall hasta un nivel aceptable de precisión. Se podría implementar una métrica como **F2-Score**, que pondera el recall el doble que la precisión, alineándose mejor con los objetivos del proyecto.

7.1.2 Optimización de Hiperparámetros

Los modelos se entrenaron con parámetros mayormente por defecto. Una búsqueda sistemática de hiperparámetros podría mejorar los resultados.

- **Estrategia:** Utilizar `GridSearchCV` o `RandomizedSearchCV` para encontrar la combinación óptima de parámetros para el `TfidfVectorizer` y el clasificador de Regresión Logística (ej. el parámetro de regularización 'C').

7.2 Mejoras Avanzadas y Futuras Direcciones

Estas propuestas implican cambios más profundos en la metodología para capturar patrones más complejos.

7.2.1 Ingeniería de Características Contextuales y Semánticas

Para que el modelo tenga un mayor entendimiento de palabras, se pueden incorporar características y técnicas de NLP (Natural Language Processing):

1. **Características Semánticas:** Reemplazar o complementar TF-IDF con *word embeddings* (ej. Word2Vec, GloVe). Estos modelos representan palabras como vectores en un espacio donde las palabras con significados similares están más cerca, logrando que el modelo generalice mejor.

2. **Análisis de Sentimientos:** Incorporar una puntuación de sentimiento por mensaje o por usuario para modelar el tono emocional de la conversación.

7.2.2 Adopción de Modelos de Lenguaje Profundo (Deep Learning)

Un paso mucho más avanzado, sería implementar modelos de lenguaje pre-entrenados que son el estado del arte en NLP.

- **Modelos Basados en Transformers (ej. BERT):** A diferencia de los métodos anteriores, estos modelos procesan el texto completo y entienden el contexto de cada palabra en relación con las demás. Un *fine-tuning* de un modelo como BERT sobre los datos de ChatSentinel podría capturar detalles del lenguaje de grooming que el modelo actual no puede, mejorando drásticamente la detección de casos difíciles (Falsos Negativos).

7.2.3 Hacia un Sistema de Detección en Tiempo Real

Los temas de este proyectop podrían aplciarse un sistema de producción.

- **Validación Continua:** Desplegar el modelo en un entorno controlado para evaluar su rendimiento con datos nuevos y medir su latencia.

- **Explicabilidad (XAI):** Integrar herramientas como LIME o SHAP para que, cuando el sistema emita una alerta, pueda proporcionar una explicación de qué palabras o frases específicas activaron la alarma. Esto es crucial para la confianza y usabilidad por parte de los moderadores humanos.

8 Conclusiones

8.1 Logros del Proyecto

ChatSentinel ha demostrado la viabilidad técnica de la detección automática de grooming mediante análisis de texto, estableciendo una base para futuros avances. Nuestro principales logros incluyen:

1. **Iteración Metodológica Exitosa:** El proyecto nos llevo a aprender a identificar fallas en un enfoque inicial, diagnosticar la causa y arreglarlo exitosamente hacia una solución superior.

2. **Desarrollo de un Modelo de Clasificación Robusto:** El modelo final de Regresión Logística es capaz de identificar a los predadores con un **recall del 90 %** y un F1-Score de 0.87.

3. **Validación de Patrones Lingüísticos Relevantes:** A través del análisis de características del modelo final, se confirmó cuantitativamente que existen patrones lingüísticos distintivos para predadores y víctimas, validando la hipótesis del proyecto.

8.2 Lecciones Aprendidas

1. **Metodología VS Métricas Superficiales:** La lección más importante fue que una precisión del 100 % en el modelo inicial no indicaba éxito, sino un error metodológico. Esto indica que la comprensión del problema nos ayudo a no ser engañados por las métricas aisladas.

2. **Validación Robusta:** La implementación de `GroupShuffleSplit` fue importante para evitar la fuga de datos y obtener una estimación realista del rendimiento del modelo.

3. **Contexto del Problema:** Se determinó que para un problema de seguridad como este, el **recall** para la clase 'Predator' era la métrica más importante, priorizando la minimización de falsos negativos sobre otras medidas de rendimiento.

8.3 Reflexión Final

ChatSentinel es un intento de avance hacia la automatización fiable de la detección de grooming. El modelo final, con su alto recall, demuestra una base sólida y prometedora para futuras herramientas de seguridad. Ya no se trata de si la detección es posible, sino de cómo refinarla y hacerla más robusta.

Aunque el modelo es efectivo, no es infalible. Sus limitaciones en la detección de grooming temprano y sutil nos recuerdan que el camino hacia un sistema completamente autónomo requiere una combinación de avances técnicos (como los modelos de lenguaje profundo propuestos), consideraciones éticas (privacidad y explicabilidad) y validación en el mundo real.

La protección de menores en el entorno digital es una responsabilidad colectiva. ChatSentinel demuestra que, con una metodología rigurosa, la ciencia de datos nos ayuda a crear herramientas potentes y confiables para construir un entorno digital más seguro.

Referencias

ChatCoder Dataset. (2020). ChatCoder 2.0 Dataset (archivado vía Wayback Machine) [Accedido en junio de 2025. URL no disponible públicamente.].

Li, Y., Su, H., Shen, X., Li, W., Cao, Z., & Niu, S. (2017). DailyDialog: A Manually Labelled Multi-turn Dialogue Dataset. *Proceedings of the 8th International Joint Conference on Natural Language Processing (IJCNLP)*, 986-995. https://aclanthology.org/I17-1099