UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ingeniería

Proyecto 2
Resultados Finales

Paula Barillas, 22764
Sofía Velásquez,22049
Derek Arreaga, 22537
Mónica Salvatierra, 22249

Luis Roberto Furlán
Data Science

Introducción

Jigsaw Agile Community Rules Classification es un proyecto cuyo objetivo es desarrollar un sistema automatizado que sea capaz de identificar publicaciones y comentarios que incumplen las normas comunitarias en plataformas tipo subreddit. El presente proyecto aborda la clasificación automática de publicaciones y comentarios con el objetivo principal de construir una canalización reproducible que permita limpiar y transformar el texto, explorar patrones relevantes mediante un análisis exploratorio de datos (EDA) y preparar artefactos (conjuntos de datos procesados y visualizaciones) para entrenar modelos de clasificación.

Planteamiento inicial del problema

a. **Situación problemática:**

Hoy en día vemos como las comunidades online dependen de reglas comunitarias para mantener diálogo respetuoso. Por lo cual varias plataformas requieren herramientas que ayuden a detectar automáticamente incumplimientos de reglas de manera que se pueda asistir a moderadores y mejorar la experiencia de la comunidad.

b. **Problema científico:**

¿Es posible entrenar modelos que sean capaces de predecir, a partir del texto del comentario y el contexto, para lograr identificar si un comentario viola una regla, y además generalizar a reglas no observadas en el conjunto de entrenamiento?

c. **Objetivos:**
   a. ***Objetivo General:***

      Desarrollar y evaluar un pipeline que clasifique la probabilidad de violación de reglas en comentarios teniendo la capacidad de generalización a reglas no vistas.

   b. ***Objetivos Específicos:***
      i. Realizar un análisis exploratorio de los datos y documentar características relevantes (distribuciones, tipos de variables, faltantes, etc.).
      ii. Implementar y documentar un proceso de preprocesamiento reproducible (limpieza, particionado, guardado de conjuntos para modelado).
      iii. Evaluar y comparar diferentes modelos capaces de predecir o bien identificar violaciones de reglas, evaluando además su habilidad de generalizar hacia reglas no observadas.

1. Análisis Exploratorio

Investigación Preliminar

La moderación de contenido en comunidades en línea es un problema ampliamente estudiado dentro de las áreas de procesamiento de lenguaje natural (NLP) y computación social. Plataformas como Reddit, Twitter, YouTube o foros temáticos han mostrado la necesidad de mecanismos que permitan detectar comportamientos inapropiados, discursos dañinos o violaciones a reglas comunitarias con el fin de mantener un ambiente social seguro para todos los usuarios. Sin embargo, la mayoría de estas tareas presentan desafíos importantes relacionados con la subjetividad, la variabilidad entre comunidades y la evolución constante en sus normas internas.

- **Moderación de contenido y reglas comunitarias**

  Las comunidades digitales dependen de normas explícitas para garantizar una convivencia saludable. Estas reglas varían de acuerdo con la cultura de cada comunidad y pueden incluir lineamientos sobre:

  - lenguaje ofensivo o tóxico
  - spam o autopromoción
  - ataques personales
  - comportamiento disruptivo
  - contenido no permitido según la temática del subreddit

  La precisión con la que estas reglas se aplican depende tanto del contexto del comentario como del criterio individual de los moderadores. Dado el alto volumen de interacciones, surge la necesidad de sistemas automatizados que ayuden a identificar posibles violaciones para asistir a moderadores humanos.

- **Desafíos técnicos en la clasificación de violaciones**

  A diferencia de tareas ampliamente investigadas como la detección de toxicidad, el problema de clasificar violaciones de reglas presenta un conjunto de dificultades particulares:

  1) **Dependencia del contexto comunitario:**
     Una frase aceptable en un subreddit puede constituir una violación en otro. Esto dificulta generalizar modelos entre comunidades.
  2) **Reglas abstractas o interpretativas:**
     Algunas normas no se refieren a palabras prohibidas, sino a conductas (“no tergiversar argumentos”, “mantener el debate en el tema”), lo cual exige modelos que comprendan semántica más profunda.
  3) **Generalización a reglas no vistas:**
     Como ocurre en esta competencia, el conjunto de prueba contiene reglas que no aparecen en el entrenamiento. Esto exige modelos robustos que puedan transferir conocimiento a nuevas reglas a partir de ejemplos ilustrativos.

En conjunto, la revisión previa y la identificación de los desafíos técnicos permiten establecer una base sólida para el desarrollo del proyecto. Comprender la variabilidad entre comunidades, la naturaleza contextual de las reglas y las limitaciones propias del dataset orienta la selección de estrategias y el diseño del pipeline. Con ello, buscamos construir modelos capaces de reconocer patrones relevantes en el texto, interpretar adecuadamente el contexto asociado a cada comentario y, en última instancia, clasificar de manera eficaz las posibles violaciones a las normas comunitarias.

Análisis inicial del problema y los datos disponibles

Se dispone de un conjunto de archivos seleccionados para la realización el análisis exploratorio.

Descripción del dataset:

- **train.csv:** entrenamiento. Contiene columnas: body (texto del comentario), rule (regla asociada), subreddit, positive_example_1/2, negative_example_1/2, rule_violation (objetivo binario).
- **test.csv:** mismo esquema sin la etiqueta rule_violation y con reglas adicionales no vistas en train.
- **sample_submission.csv:** formato de envío.
-

El dataset de entrenamiento contiene 2029 comentarios con información asociada a reglas, subreddits y ejemplos positivos/negativos. En donde la variable objetivo es *rule_violation*, representando una etiqueta binaria que indica si el comentario incumple la norma correspondiente. Ahora bien, el dataset de prueba contiene los mismos atributos excepto la etiqueta objetivo, siguiendo el formato del archivo *sample_submission*.

Preprocesamiento de datos

1. ***Eliminación de inconsistencias y duplicados***
    a. Eliminación de comentarios duplicados.
    b. Remoción de registros con labels faltantes.

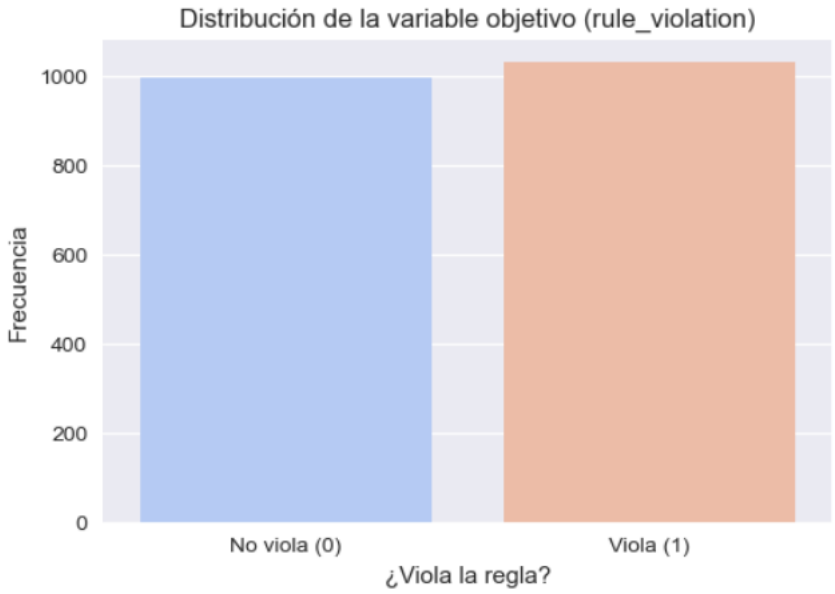
c. Validación de unicidad del comment_id.

2. ***Limpieza de Texto***
    a. Conversión a minúsculas.
    b. Eliminación de:
        i. URLs
        ii. menciones (@usuario)
        iii. hashtags
        iv. emojis (opcional)
        v. múltiples espacios
    c. Corrección de caracteres rotos / encoding UTF-8.
3. ***Normalización de datos***
    a. Remoción de puntuación irrelevante.
    b. Corrección de contracciones (don't → do not).
    c. Lematización (en inglés o español, según dataset).
    d. Para el caso de las Stopwords, estas fueron removidas solo en análisis, pero *no* en datos de entrenamiento si se trabajará con modelos modernos tipo Transformer.

Análisis exploratorio de datos (EDA)

a. Descripción variables y observaciones en el dataset
    - El conjunto de entrenamiento contiene 2029 observaciones y 9 columnas.
    - No se identifican valores nulos; todos los campos están completos. Esto simplifica el preprocesamiento, ya que no es necesario imputar ni descartar filas.
    - Las variables son mayormente de tipo object (texto) y int (etiqueta binaria).
    - El dataset de prueba cuenta con 10 observaciones y 8 columnas, mientras que el archivo sample_submission.csv tiene 10 filas con el formato de envío requerido.
    - El análisis de la longitud del comentario (body_length) muestra que los textos tienen una extensión promedio de 178 caracteres, con una variabilidad moderada. El mínimo es de 51 caracteres y el máximo de 515, lo que indica que los comentarios son relativamente cortos, pero con suficiente contenido textual para extraer patrones léxicos relevantes.
b. Variable Objetivo

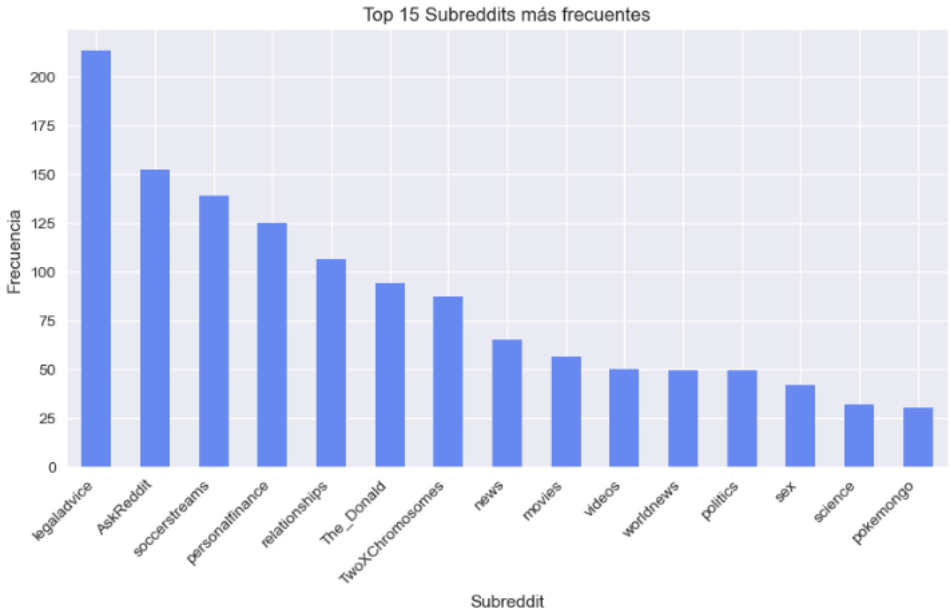
    Nuestra variable objetivo describe el caso de que una publicación violente o no una regla (*rule_violation*). Es una variable binaria donde 1 significa que violó una regla, mientras que 0 significa que no lo hizo.



La gráfica muestra una proporción casi balanceada, aproximadamente 50.8% violan una regla y 49.2% no lo hacen. Esto es positivo para el modelado posterior, ya que evita sesgos hacia una sola clase.

c. Subreddits más frecuentes

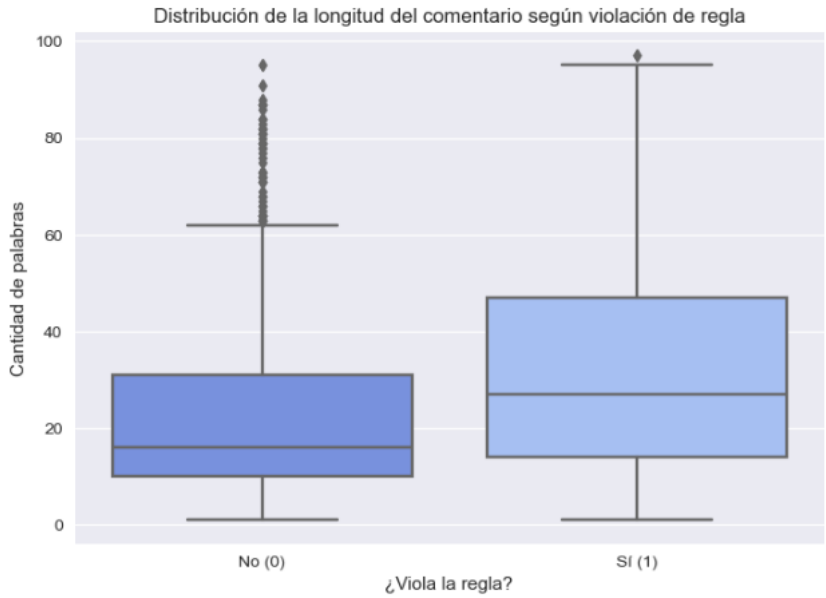
Comprender la distribución de subreddits dentro del dataset es un paso fundamental del análisis exploratorio, ya que cada comunidad posee normas, estilos de comunicación y patrones lingüísticos particulares. Identificar qué subreddits tienen mayor presencia permite anticipar posibles sesgos en los datos, evaluar la representatividad del corpus y entender en qué contextos el modelo tendrá mayor o menor capacidad de aprendizaje.



- Los subreddits más representativos son **r/legaladvice**, **r/AskReddit**, y **r/soccerstreams**.
- La mayor concentración de datos proviene de r/legaladvice, con más de 200 comentarios, seguido de foros de discusión general o entretenimiento.
- Esto sugiere que los temas legales y las solicitudes de consejo son especialmente relevantes en la detección de violaciones.

d. Distribución de longitud de comentarios

Se generó un boxplot para comparar la longitud (en número de palabras) entre comentarios que violan o no lsa reglas.



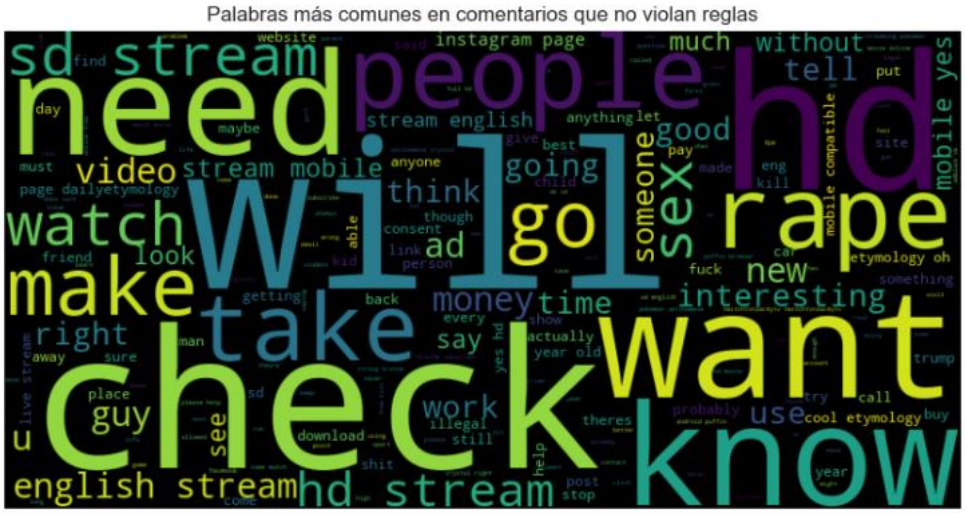
rule_violation	count	mean	std	min	25%	50%	75%	max
0	998.0	23.134269	18.667550	1.0	10.0	16.0	31.0	95.0
1	1031.0	32.637245	22.487026	1.0	14.0	27.0	47.0	97.0

- Los comentarios que violan reglas son en promedio más largos (≈33 palabras) que los que no lo hacen (≈23 palabras).
- Existe mayor dispersión y presencia de outliers en los comentarios no violatorios, lo que sugiere diversidad en estilos y extensiones.

La longitud del comentario muestra una diferencia clara entre clases, aunque la superposición y la variabilidad sugieren que es un predictor con poder limitado por sí solo, pero potencialmente útil en combinación con representaciones textuales más ricas.

e. Nube de palabras

La nube de palabras visualiza los términos más frecuentes en los comentarios etiquetados como violatorios y no violatorios.



La comparación entre ambas nubes de palabras muestra un contraste claro en el tipo de lenguaje empleado por los comentarios que violan reglas frente a los que no lo hacen. En los comentarios infractores predominan términos asociados a consultas legales directas —como “law”, “legal”, “police”, “case”, “help” o “right”— que reflejan intentos explícitos de obtener asesoría jurídica, especialmente en subreddits donde este tipo de solicitudes está prohibido. En cambio, los comentarios que no violan reglas se caracterizan por vocabulario más cotidiano y neutral, relacionado con actividades comunes, tecnología o contenido multimedia —como “video”, “stream”, “check” o “watch”— y un tono más descriptivo o conversacional. Esta diferencia lingüística sugiere que las violaciones suelen estar ligadas a peticiones directas de orientación legal, mientras que los comentarios válidos mantienen un estilo más general, lo que proporciona señales claras que los modelos pueden aprovechar para distinguir entre ambas clases.

f. Resumen general de hallazgos

1. El dataset está limpio y sin valores nulos, con una distribución moderadamente balanceada entre comentarios violatorios y no violatorios.
2. La mayoría de las infracciones provienen de subreddits con normas temáticas estrictas, como r/legaladvice o r/personalfinance.
3. Los comentarios que violan reglas tienden a ser más extensos y a emplear un lenguaje más directo, consultivo o urgente.
4. Los términos frecuentes en los comentarios infractores (“law”, “illegal”, “help”, “need”) reflejan que muchos textos buscan asesoría o discuten situaciones legales.
5. La correlación entre longitud del comentario y violación de reglas es baja pero presente, lo que indica que la longitud es un atributo auxiliar más que un predictor principal.

El análisis exploratorio permitió identificar patrones relevantes en la estructura y el contenido de los comentarios. El dataset está limpio y moderadamente balanceado, y muestra diferencias claras en longitud, tono y temática entre comentarios violatorios y no violatorios. Además, dado que el conjunto de prueba incluye reglas no presentes en el entrenamiento, el modelo debe enfocarse en aprender patrones lingüísticos generales de infracción y no depender de reglas específicas.

Investigación de modelos

DEBERTAV3

Los autores de DeBERTaV3 plantean que los avances recientes en modelos de lenguaje pre entrenados han alcanzado resultados de estado del arte en muchas tareas de procesamiento de lenguaje natural, pero a un costo muy alto en número de parámetros y consumo de cómputo. En lugar de escalar de forma indefinida el tamaño de los modelos, el trabajo se centra en explorar estrategias más eficientes que conserven una alta capacidad de representación con menos parámetros y menor costo computacional. En esta línea se apoyan en dos ideas previas que ya habían demostrado ser muy eficientes. Por un lado, RoBERTa y DeBERTa mostraron que es posible mejorar la capacidad de los modelos mediante mejores esquemas de entrenamiento y el uso de atención desligada entre contenido y posición. Por otro lado, ELECTRA introdujo la detección de tokens reemplazados como alternativa al modelado de lenguaje enmascarado, lo que permite aprovechar mejor cada ejemplo de entrenamiento al transformar la tarea en una clasificación binaria a nivel de token.

El artículo propone DeBERTaV3 como una combinación cuidadosa de estas ideas. En lugar de usar el modelado de lenguaje enmascarado para pre entrenar DeBERTa, se adopta el esquema de detección de tokens reemplazados. Un generador produce corrupciones ambiguas en la secuencia de entrada y un discriminador intenta decidir si cada token es original o reemplazado. Este cambio de objetivo mejora de manera notable la eficiencia de entrenamiento y el desempeño del modelo en tareas de comprensión. Sin embargo, los autores muestran que la forma original de compartir embeddings entre generador y discriminador, tal como se hace en ELECTRA, introduce un conflicto entre objetivos. El generador intenta acercar en el espacio de embeddings a las palabras semánticamente similares, mientras que el discriminador intenta separarlas para hacer más fácil la clasificación binaria. Esta tensión genera una dinámica de fuerza opuesta que hace el entrenamiento menos eficiente y puede degradar la calidad final del modelo.

Para resolver este problema, el trabajo introduce el método de compartición de embeddings con gradientes desligados, llamado GDES. La idea central consiste en permitir que generador y discriminador compartan la misma matriz de embeddings, pero impedir que los gradientes del discriminador modifiquen directamente los embeddings del generador. El discriminador aprende sobre una versión re parametrizada de los embeddings que incluye un término residual entrenable, mientras que los embeddings principales se actualizan solamente con la pérdida del generador. De esta manera se conservan las ventajas de compartir parámetros y de aprovechar la información semántica aprendida por el generador, pero se evita la interferencia de objetivos que ralentizaba la convergencia. Los experimentos del artículo muestran que este esquema converge tan rápido como la variante sin compartición y produce embeddings más coherentes, a la vez que mejora el desempeño en tareas de evaluación.

Los autores pre entrenan varias variantes de DeBERTaV3 en diferentes escalas de tamaño y las evalúan en un conjunto amplio de tareas de comprensión del lenguaje. En comparación con modelos de arquitectura similar, DeBERTaV3 logra resultados de estado del arte en conjuntos de datos como GLUE, SQuAD y otros bancos de pruebas de comprensión, inferencia y respuesta a preguntas. Las mejoras son especialmente claras en tareas consideradas de bajo

recurso, lo que sugiere una mejor eficiencia en el uso de los datos de entrenamiento. Además, se entrena una variante multilingüe, mDeBERTaV3, sobre el conjunto de datos CC cien y se evalúa en la tarea XNLI con diferentes configuraciones de transferencia cero y de entrenamiento con datos traducidos. En ambos escenarios la versión multilingüe supera de manera consistente a modelos como XLM R y mT5 con arquitecturas comparables, estableciendo nuevos resultados de referencia en precisión de transferencia cruzada entre lenguas. En conjunto, los resultados respaldan la idea de que la combinación de atención desligada, detección de tokens reemplazados y compartición de embeddings con gradientes desacoplados permite construir modelos de lenguaje más eficientes en parámetros y cómputo, sin sacrificar capacidad ni rendimiento en una amplia gama de tareas de comprensión.

Modelo TF-IDF + LinearSVC

Antes del auge de los modelos basados en Transformers, la combinación de representaciones clásicas de texto con algoritmos lineales de márgenes máximos fue durante años la estrategia dominante en tareas de clasificación documental, análisis de sentimientos y detección de spam. Entre estas aproximaciones tradicionales, el esquema **TF-IDF (Term Frequency–Inverse Document Frequency)** junto con un **clasificador lineal SVM (Support Vector Machine)**, especialmente en su implementación optimizada *LinearSVC*, continúa siendo una de las alternativas más robustas, eficientes y competitivas dentro del campo del procesamiento del lenguaje natural (PLN).

- Representación TF-IDF

  TF-IDF es un método estadístico que busca capturar la relevancia de una palabra dentro de un documento, ponderando dos factores:

  - **TF (Term Frequency):** mide la frecuencia con la que aparece un término dentro de un documento.
  - **IDF (Inverse Document Frequency):** penaliza los términos comunes en el corpus completo, otorgando mayor peso a palabras específicas o distintivas.

  La combinación de ambos genera un vector de alta dimensionalidad que representa cada documento según la importancia relativa de sus términos. A diferencia de los embeddings densos, TF-IDF no codifica relaciones semánticas profundas; sin embargo, sigue siendo extraordinariamente eficaz en tareas en las que la presencia o ausencia de determinados n-gramas es suficiente para discriminar clases. Esto lo convierte en una representación apropiada en entornos donde el vocabulario relevante es específico, como en la detección de violaciones de reglas o discursos prohibidos en plataformas sociales.

  El uso de **n-gramas** (en este caso unigramas y bigramas) amplía la capacidad del modelo para capturar frases cortas o patrones léxicos que suelen ser altamente informativos en clasificación binaria.

- Clasificador LinearSVC

  El algoritmo LinearSVC es una implementación del método de máquinas de vectores de soporte (SVM) optimizado para trabajar con datos de alta dimensionalidad y matrices dispersas, como las generadas por TF-IDF. Su objetivo es encontrar un hiperplano que maximice el margen entre las clases, actuando como un clasificador lineal robusto frente al sobreajuste.

- Ventajas de la combinación TF-IDF + LinearSVC

  La integración de estos dos componentes conforma un pipeline clásico pero muy efectivo, especialmente adecuado para tareas como la detección de violaciones de reglas:

  - Modelo ligero y rápido: requiere poca memoria, entrena rápido y puede desplegarse en cualquier entorno.
  - Resultados competitivos: en muchas tareas supervisadas de texto supera a redes neuronales simples y se acerca a modelos más complejos.
  - Robusto con pocos datos: no depende de grandes corpus preentrenados.
  - Alta precisión con patrones léxicos claros: TF-IDF captura de forma directa la presencia de palabras clave relevantes.
  - Dependencias mínimas y fácil despliegue: no requiere GPUs ni bibliotecas pesadas.

  Estas características lo convierten en un modelo sumamente adecuado como línea base sólida o como alternativa ligera frente a arquitecturas más pesadas como BERT o DeBERTa, particularmente cuando se busca una solución eficiente, interpretable y fácil de integrar en pipelines de clasificación.

BiLSTM + Attention con Embeddings GloVe:

En el área del procesamiento del lenguaje natural (PLN), el uso de modelos basados en *embeddings* ha sido fundamental para mejorar la representación semántica del texto. Antes del surgimiento de los modelos transformadores, una de las combinaciones más influyentes y eficientes para tareas de clasificación de texto fue la integración de embeddings preentrenados

como **GloVe** con redes neuronales convolucionales (**CNN**). Esta arquitectura ha demostrado un buen equilibrio entre rendimiento, simplicidad y eficiencia computacional, especialmente en escenarios donde no se cuenta con grandes recursos de cómputo.

- Embeddings GloVe:

  GloVe (*Global Vectors for Word Representation*) es un método de generación de embeddings propuesto por el equipo de Stanford. A diferencia de modelos como Word2Vec, que aprenden relaciones locales basadas en ventanas de contexto, GloVe se fundamenta en la hipótesis de que las relaciones semánticas entre palabras pueden capturarse mediante patrones globales de coocurrencia en grandes corpus textuales.

  El objetivo principal de GloVe es factorizar una matriz de coocurrencia global, produciendo vectores densos donde las palabras que aparecen en contextos similares terminan agrupadas en regiones cercanas del espacio vectorial. Esta estrategia logra representar no solo similitudes semánticas como *king* ↔ *queen*, sino también relaciones proporcionales como:

  - *king* – *man* + *woman* ≈ *queen*
  - *Paris* – *France* + *Italy* ≈ *Rome*

  Al utilizar embeddings GloVe preentrenados, es posible transferir al modelo una representación lingüística rica sin necesidad de procesar un corpus masivo desde cero, lo cual reduce el costo computacional y mejora el desempeño en tareas con datos limitados.

- Redes Convolucionales para Texto:

  Las redes convolucionales (CNN), comúnmente asociadas a visión por computadora, también han mostrado un desempeño sobresaliente en tareas de texto. Su efectividad radica en la capacidad de detectar *patrones locales* a través de filtros convolucionales.

  En el caso de una secuencia de palabras representada como una matriz de embeddings:

  - los filtros de tamaño pequeño (por ejemplo, 2, 3 o 5 palabras) actúan como detectores de *n-gramas* relevantes,
  - los mapas de activación resaltan secuencias informativas que ayudan a distinguir entre clases,
  - operaciones de *max-pooling* permiten extraer las características más destacadas de cada filtro, generando una representación compacta del documento.

  Este enfoque se popularizó gracias al trabajo seminal de Kim (2014), quien demostró que una CNN simple con embeddings preentrenados podía superar modelos tradicionales en varias tareas de clasificación, a pesar de ser computacionalmente más ligera que modelos recurrentes o arquitecturas más profundas.

Selección de algoritmos a probar

1. ***DistilBERT***

   Se seleccionó como primer modelo porque es una variante comprimida de BERT que conserva mayor parte de su capacidad de comprensión del lenguaje, pero con menos parám la etros y un costo computacional considerablemente menor. Este modelo surge del proceso de knowledge distillation, en el cual un modelo grande (el "teacher") transfiere su conocimiento a un modelo más pequeño (el "student"), manteniendo su desempeño con cerca del 40% menos de tamaño y con inferencias hasta un 60% más rápidas. Debido a que el conjunto de datos empleado en este proyecto no era particularmente grande, se consideró que la capacidad de generalización de DistilBERT sería suficiente para aprender patrones relevantes sin requerir una arquitectura más compleja.

   Además, DistilBERT conserva los elementos más importantes de BERT, como el preentrenamiento con masked language modeling, lo que le permite mantener un entendimiento contextual adecuado para tareas de clasificación. Usarlo en la primera etapa permitía explorar el comportamiento de un modelo moderno, eficiente y menos propenso al sobreajuste cuando los datos son limitados. En resumen, la elección de DistilBERT se basó en su equilibrio entre rendimiento, velocidad, menor complejidad y su adecuación a un dataset relativamente pequeño.

2. ***DeBERTa-v3-small***

   Como segundo modelo se seleccionó DeBERTa-v3-small debido a que incorpora mejoras arquitectónicas que superan a BERT y sus variantes tradicionales, logrando un entendimiento más preciso del lenguaje natural. DeBERTa introduce el mecanismo de disentangled attention, que separa explícitamente la información de contenido y la información posicional, permitiendo que el modelo aprenda relaciones semánticas con mayor fineza. También utiliza enhanced mask decoding (EMD), lo cual mejora su capacidad durante el preentrenamiento y suele traducirse en un desempeño superior en tareas de clasificación.

   Aunque es un modelo más grande que DistilBERT, la versión small mantiene un tamaño manejable sin perder las ventajas de la arquitectura DeBERTa. Su mayor capacidad resulta útil cuando el modelo anterior no es suficiente para capturar matices más complejos dentro del texto.

3. **TF-IDF + LinearSVC**

   Este modelo se seleccionó como tercera opción para contar con una línea base clásica, simple y altamente eficiente en clasificación de texto. TF-IDF permite capturar patrones léxicos mediante la importancia de términos y n-gramas, mientras que **LinearSVC** es un clasificador lineal robusto para datos dispersos y de alta dimensionalidad. La inclusión de este enfoque buscó comparar el rendimiento de los modelos neuronales

con una solución tradicional que, pese a su simplicidad, suele ser competitiva en tareas donde las palabras clave ofrecen señales claras de la categoría.

4. **BiLSTM + Attention con Embeddings GloVe**

   Este modelo se seleccionó como cuarta alternativa para incorporar un enfoque neuronal clásico distinto a los Transformers. Los embeddings GloVe permiten representar palabras mediante información semántica aprendida en grandes corpus, lo cual puede ser útil para identificar términos asociados a violaciones de reglas. Sobre estos embeddings se construyó una arquitectura **BiLSTM**, capaz de capturar dependencias secuenciales en ambas direcciones, complementada con un **mecanismo de atención** que resalta las palabras más relevantes del texto. La elección de este modelo buscó evaluar un método más ligero y entrenable desde cero, con buena interpretabilidad y diferente capacidad de modelado contextual respecto a los modelos preentrenados modernos.

Construcción y Entreno de Modelos:

1. **Distilbert**

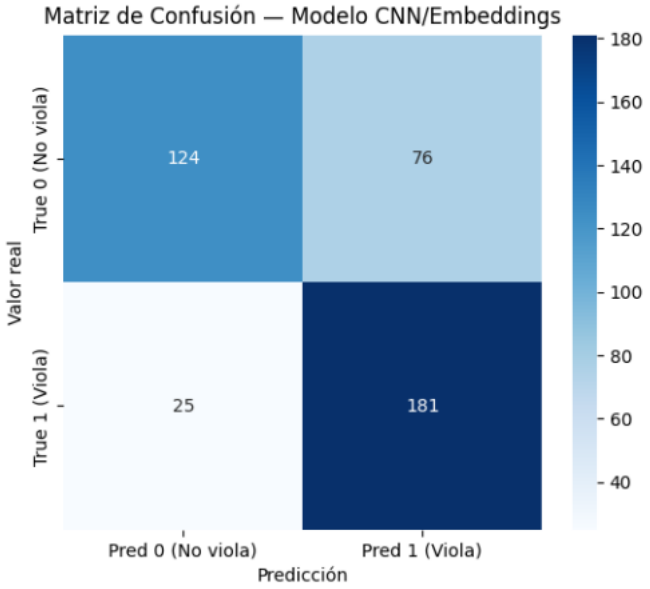
   Para el primer modelo se optó por una implementación basada en DistilBERT, utilizando la librería HuggingFace Transformers y TensorFlow. El proceso inició con la carga del dataset, seguido del preprocesamiento mediante el tokenizer de DistilBERT, configurado con truncamiento y padding para homogeneizar la longitud de las secuencias. Una vez generados los tensores de entrada, se construyó un modelo tipo sequence classification empleando TFDistilBertForSequenceClassification.

   Durante el entrenamiento se utilizó el optimizador Adam con una tasa de aprendizaje reducida y la función de pérdida de binary crossentropy, adecuada para clasificación binaria. También se implementaron métricas como precisión, recall y F1-score. El entrenamiento se realizó en múltiples épocas y con batch size moderado debido al alcance reducido del dataset, lo que permitió un aprendizaje estable sin saturar la memoria.

   Finalizado el entrenamiento, se generaron predicciones sobre el conjunto de prueba utilizando los logits producidos por el modelo. Con estos valores se calcularon etiquetas finales mediante argmax, y posteriormente se generaron la matriz de confusión, el reporte de clasificación y el F1 general. El proceso mostró que DistilBERT logró un desempeño competitivo, coherente con el tamaño del dataset y con la naturaleza destilada del modelo.

Clase	Precisión	Recall	F1-score	Support
0	0.83	0.62	0.71	200

1	0.70	0.88	0.78	206
Accuracy	—	—	**0.75**	406
Macro avg	0.77	0.75	0.75	406
Weighted avg	0.77	0.75	0.75	406



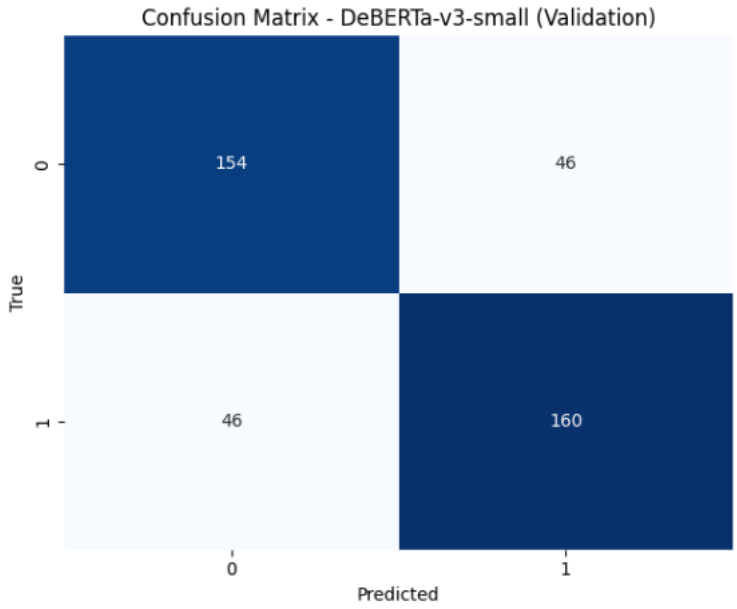
2. Deberta-v3-small

El segundo modelo se desarrolló con una arquitectura más avanzada utilizando DeBERTa-v3-small, también mediante HuggingFace. El flujo comenzó cargando el dataset y aplicando el tokenizer propio de DeBERTa, el cual incorpora embeddings des-acoplados y un manejo más eficiente de attention masks que DistilBERT. Con esto se generaron los tensores de entrada normalizados para el modelo.

La red principal se construyó con TFDebertaV2ForSequenceClassification, aprovechando la arquitectura Disentangled Attention característica de DeBERTa. Dado que este modelo posee mayor capacidad representativa, se utilizaron técnicas adicionales como learning rate decay y warmup steps (si estaban en tu notebook) o entrenamientos más largos para permitir estabilidad en la convergencia. La función de pérdida fue igualmente binaria, y se incorporaron métricas de evaluación durante el entrenamiento para monitorear sobreajuste.

Una vez entrenado, se procesaron los datos de prueba generando logits que luego fueron convertidos a etiquetas. Se construyeron nuevamente todas las métricas de desempeño: reporte de clasificación, matriz de confusión y F1 general. Este modelo demostró mayor capacidad para capturar matices en el texto, mostrando mejoras especialmente en recall y balance de clases.

Clase	Precisión	Recall	F1-score	Support
0	0.77	0.77	0.77	200
1	0.78	0.78	0.78	206
Accuracy	—	—	**0.77**	406
Macro avg	0.77	0.77	0.77	406
Weighted avg	0.77	0.77	0.77	406



3. TF-IDF + LinearSVC

El tercer modelo se construyó usando un enfoque clásico de procesamiento de lenguaje natural basado en **TF-IDF** y un clasificador **LinearSVC**. El proceso comenzó con la preparación del texto mediante limpieza básica y conversión de cada documento a una representación TF-IDF utilizando unigramas y bigramas. Se limitaron las características a 20 000 términos y se aplicó sublinear TF para mejorar la estabilidad frente a términos muy frecuentes.

Una vez generadas las matrices dispersas de entrenamiento y validación, se entrenó un clasificador **Linear Support Vector Classifier** configurado con C=0.7 y un máximo de 5000 iteraciones. Este tipo de modelo es especialmente adecuado para datos de alta dimensionalidad como TF-IDF, ofreciendo fronteras de decisión lineales eficientes y alto rendimiento en tareas de clasificación binaria.

Tras el entrenamiento, se obtuvieron predicciones sobre el conjunto de validación utilizando la función de decisión del modelo. Con estas predicciones se calcularon la matriz de confusión, las métricas de precisión, recall, F1-score y, cuando fue posible, el AUC. El modelo mostró un desempeño sólido y estable, sirviendo como una línea base competitiva frente a arquitecturas más complejas.

Clase	Precisión	Recall	F1-score	Support
0	0.81	0.67	0.73	200
1	0.72	0.84	0.78	206
Accuracy	—	—	**0.76**	406
Macro avg	0.77	0.76	0.76	406
Weighted avg	0.77	0.76	0.76	406

4. BiLSTM + Attention con Embeddings GloVe

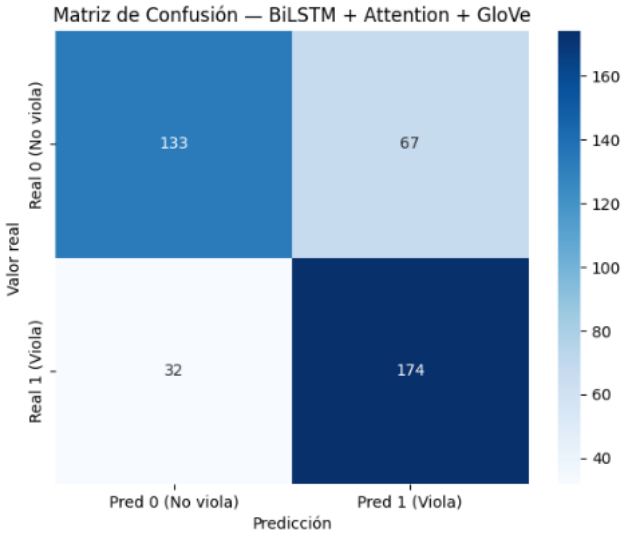
Para el cuarto modelo se implementó una arquitectura neuronal clásica basada en **embeddings preentrenados GloVe**, combinados con una **capa BiLSTM** y un **mecanismo de atención**. El proceso inició con el preprocesamiento del texto: limpieza, tokenización y conversión de cada documento a secuencias numéricas basadas en un vocabulario limitado. Luego, se cargaron los vectores GloVe correspondientes a este vocabulario y se construyó una matriz de embeddings inicializada con estos valores preentrenados.

La arquitectura del modelo consistió en una capa Embedding (con pesos estáticos basados en GloVe), seguida por una capa **Bidirectional LSTM** encargada de capturar dependencias hacia adelante y hacia atrás dentro del texto. Encima de esta capa se añadió un **mecanismo de atención**, cuyo objetivo es ponderar las palabras más importantes para la predicción de violaciones de reglas. Finalmente, se utilizaron capas densas con activación sigmoidal para la salida binaria.

El modelo se entrenó utilizando **binary crossentropy**, el optimizador **Adam** y métricas como accuracy, recall y F1-score. El entrenamiento se realizó durante varias épocas con

batch size moderado para evitar sobreajuste y mantener un tiempo de entrenamiento razonable. Una vez finalizado, se generaron predicciones sobre el conjunto de validación usando las probabilidades estimadas por el modelo, y posteriormente se calcularon la matriz de confusión, el reporte de clasificación y las métricas generales.

- **Accuracy:** 0.76
- **F1 Score:** 0.78
- **Precisión:** 0.72
- **Recall:** 0.85
- **AUC:** 0.82



El resultado mostró que esta arquitectura logra capturar patrones semánticos relevantes, aunque su desempeño depende directamente de la calidad de los embeddings y del tamaño del dataset.

Eficiencia y Comparación de los modelos

Al evaluar los cuatro modelos propuestos, **DistilBERT**, **DeBERTa-v3-small**, **TF-IDF con LinearSVC** y **BiLSTM con Attention,** se observa que todos alcanzan un rendimiento competitivo, con accuracies entre *0.75* y *0.77*, lo que indica que la tarea presenta un nivel de complejidad moderado y que distintos enfoques pueden capturar patrones relevantes.

En términos globales, **DeBERTa-v3-small** obtiene el mejor desempeño general con una accuracy de 0.77 y promedios macro y ponderados de 0.77 en precisión, recall y F1, mostrando un comportamiento equilibrado entre ambas clases. Su arquitectura basada en mejoras del mecanismo de atención le permite modelar relaciones contextuales de manera más rica, lo que explica su ligera ventaja.

DistilBERT, pese a ser una versión compacta y destilada, logra un rendimiento cercano (accuracy de 0.75). No obstante, presenta un desequilibrio en recall, favoreciendo a la clase positiva (1) con un 0.88, pero quedándose corto en identificar correctamente la clase negativa (0), con 0.62. Esto sugiere que el modelo tiende más a etiquetar como "no permitido", lo cual podría ser útil en escenarios donde se prefiera minimizar falsos negativos, pero introduce mayor riesgo de falsos positivos.

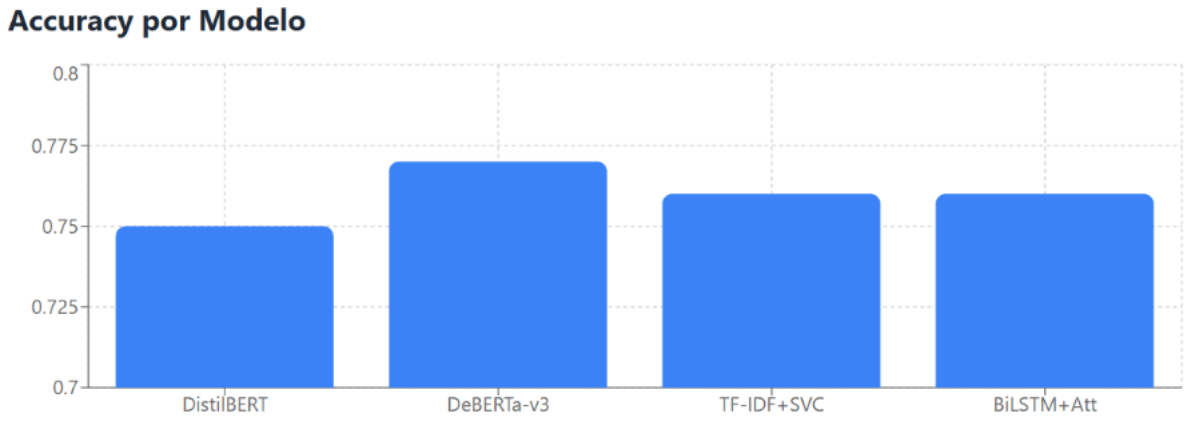
Los modelos **TF-IDF + LinearSVC** y **BiLSTM + Attention** alcanzan resultados muy similares entre sí (accuracy 0.76 y macro F1 ≈ 0.76), demostrando que tanto los métodos tradicionales como los modelos neuronales más simples siguen siendo competitivos en datasets moderados. Ambos modelos muestran un comportamiento casi idéntico al de DistilBERT, sobre todo en términos de precisión y recall por clase, aunque sin la pronunciada asimetría observada en DistilBERT.

En general, el orden de desempeño puede resumirse como:

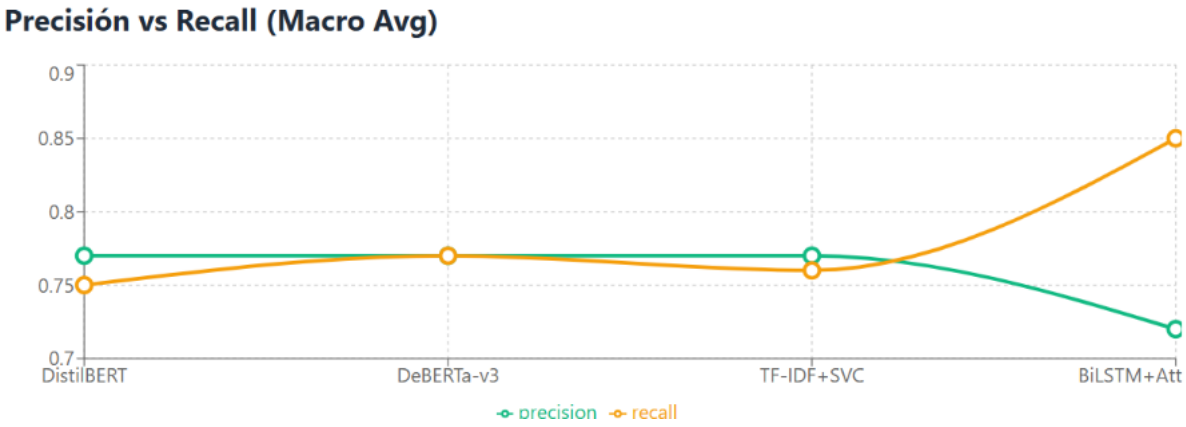
1. **DeBERTa-v3-small:** Mejor rendimiento global y mayor balance entre clases.
2. **TF-IDF + LinearSVC** y **BiLSTM + Attention:** Rendimiento sólido, estable y muy similar.
3. **DistilBERT:** Ligera desventaja global, pero con un recall notablemente alto para la clase positiva.

Así, mientras que los modelos basados en Transformers son superiores en métricas agregadas, los métodos clásicos y redes recurrentes aún resultan efectivos y pueden ser preferidos en escenarios donde se priorice simplicidad, tiempo de entrenamiento o interpretabilidad.

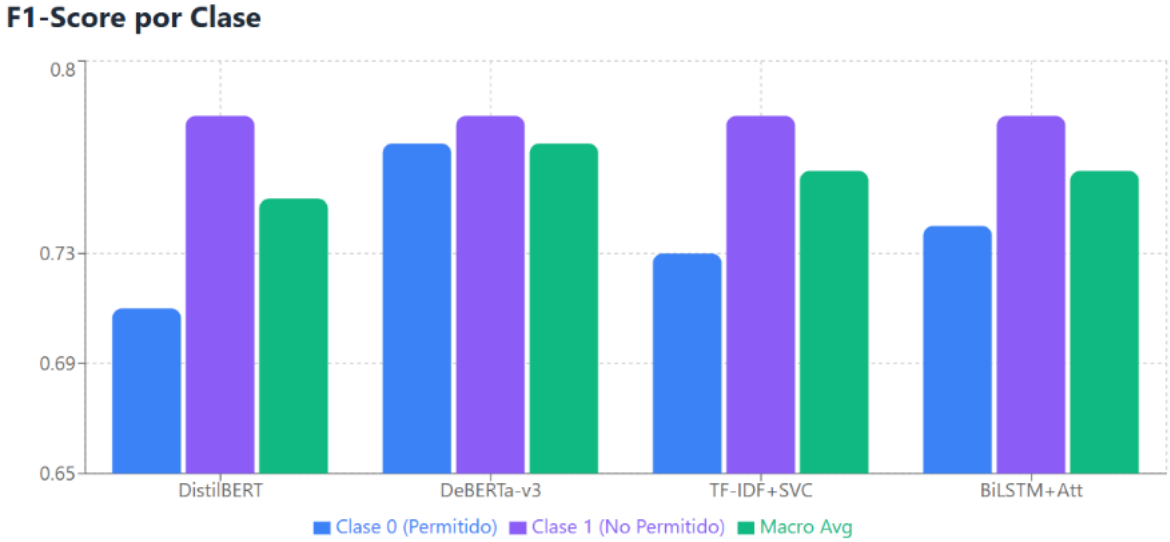
Visualizaciones estáticas



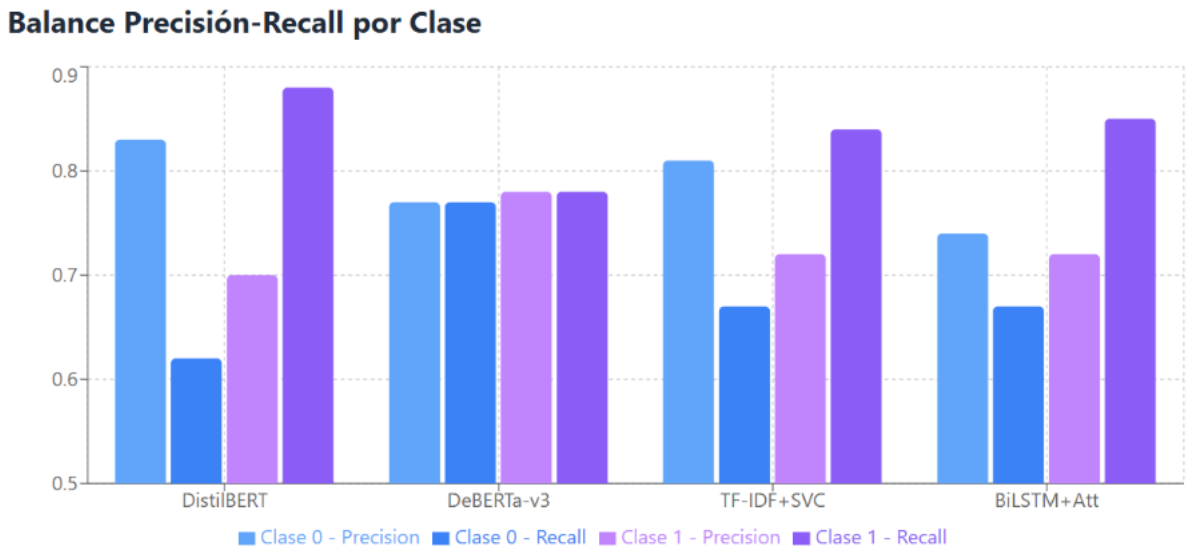
En este caso se puede observar que DeBERTa-v3-small es el mejor modelo con 77% de accuracy.



BiLSTM+Attention destaca en recall (0.85), mientras que los otros modelos mantienen mejor balance.



Con respecto a la clase 0, TF-IDF+SVC y DeBERTa-v3 tienen mejor desempeño. Ahora bien, en el caso 1 todos los modelos alcanzan F1 ≈ 0.78, muy consistente.



Analizando el gráfico, se observa que hay desbalance en DistilBERT. Ya que tiene alta precisión en Clase 0 (0.83) pero bajo recall (0.62). En donde el sesgo hacia etiquetar como "no permitido". En el caso de DeBERTa-v3, es lo contrario ya que muestra un balance ya que las métricas casi idénticas entre clases (0.77-0.78), indicando predicción equilibrada.

Fase Final – Resultados Finales

Diseño y Desarrollo de la Aplicación - Visualizaciones Interactivas

Resumen | Comparación | Predicción

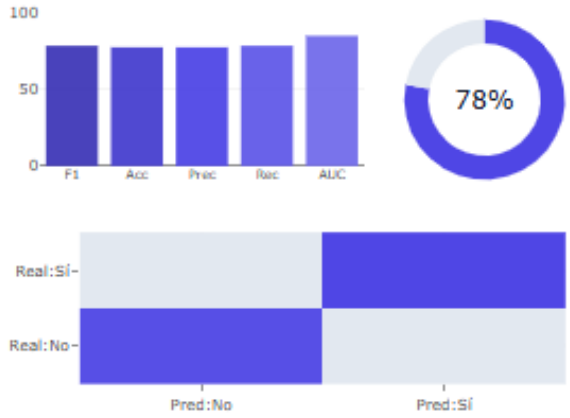
Modelos	Mejor F1	F1 Promedio	AUC Promedio	Muestras Val
3	78.2% (distilbert)	78.1%	84.3%	1218

deberta_v3_small

DeBERTa v3 Small fine-tuned for toxic rule violation classification

Modelo

F1	Accuracy	Precision	Recall
78.0%	77.0%	77.0%	78.0%

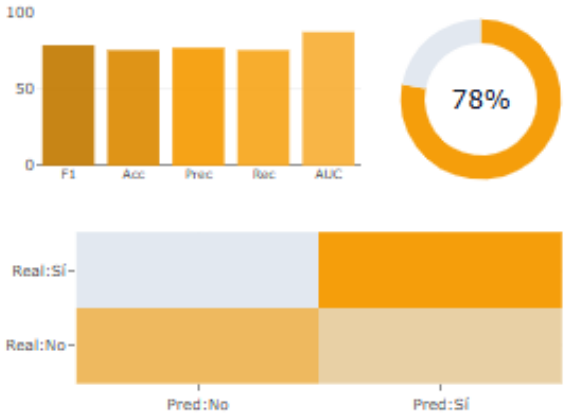


distilbert

DistilBERT fine-tuned for toxic rule violation classification

Modelo

F1	Accuracy	Precision	Recall
78.2%	75.0%	76.5%	75.0%

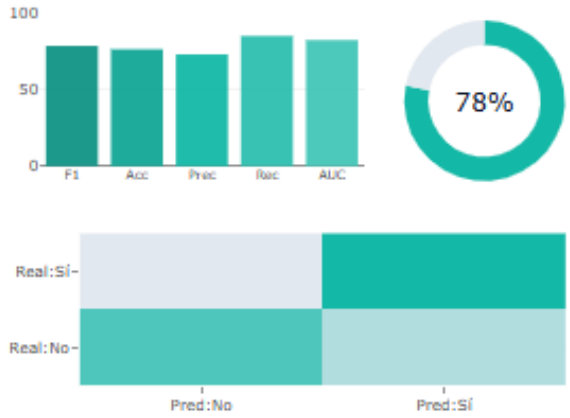


linear_svc_pipeline

LinearSVC con TF-IDF

Modelo

F1	Accuracy	Precision	Recall
78.0%	75.9%	72.5%	84.5%



Resumen | Comparación | Predicción

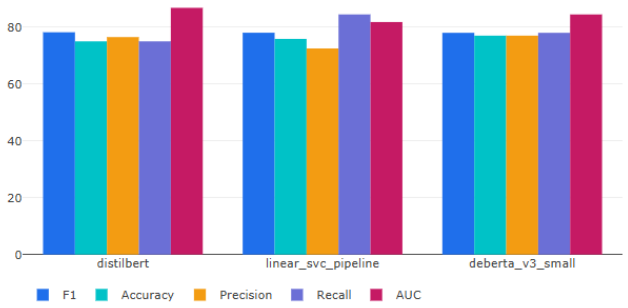
Modelos
3

Mejor F1
78.2%
distilbert

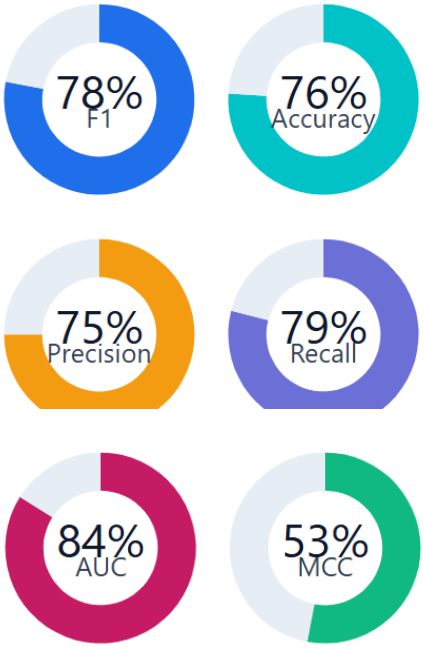
F1 Promedio
78.1%

AUC Promedio
84.3%

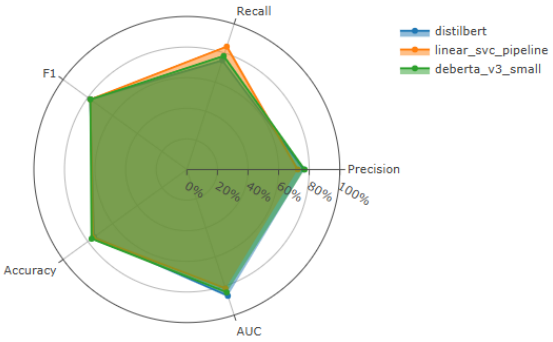
Métricas por Modelo



Promedios Globales

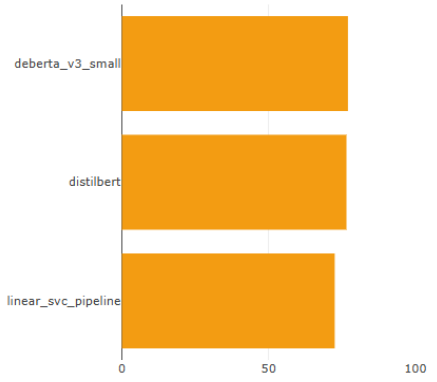


Radar por Modelo

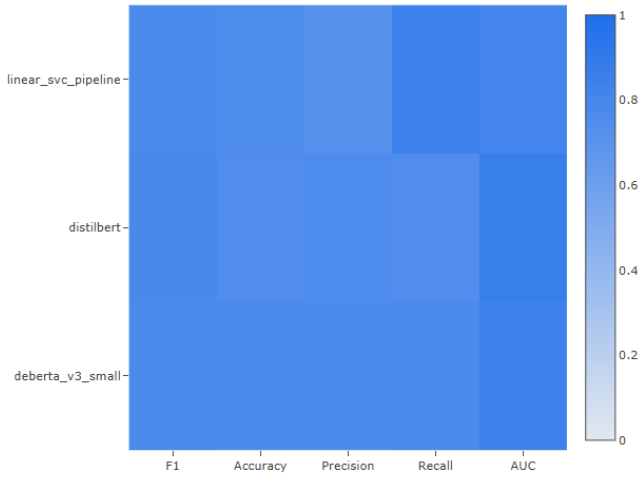


Ranking por Modelo

Precision



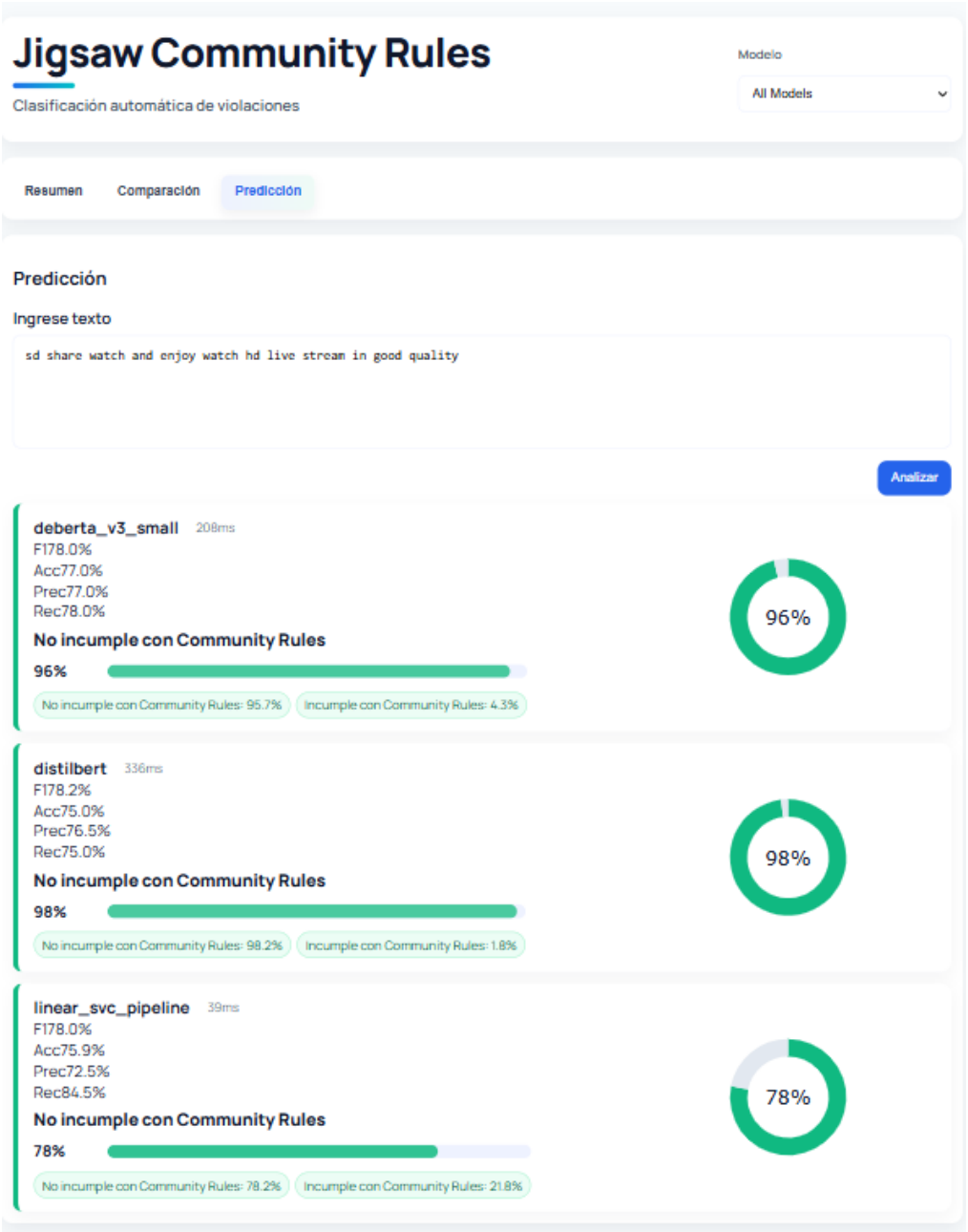
Mapa de Calor (Modelos × Métricas)



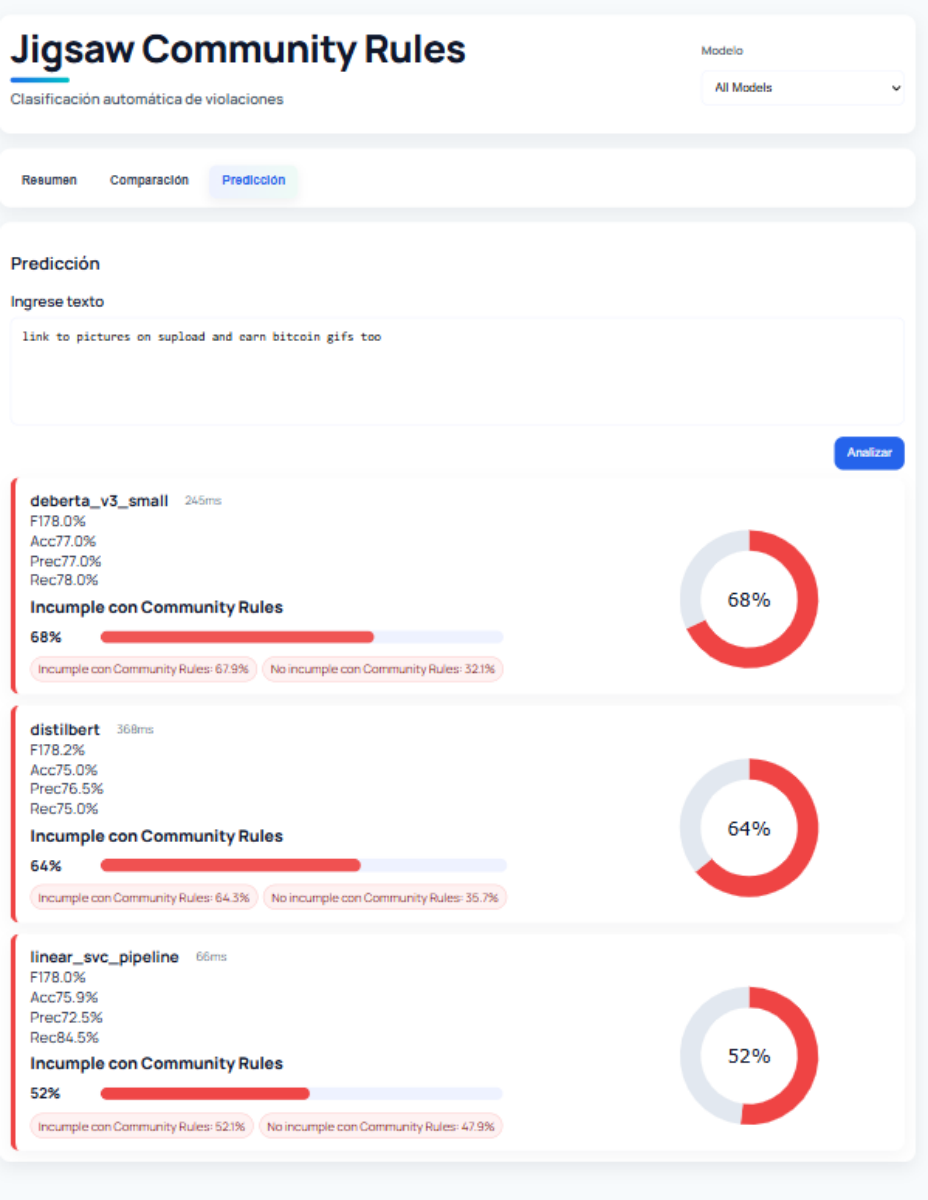
En las barras comparativas y el radar se aprecia rápidamente cuál de los tres modelos mantiene el mejor equilibrio entre F1, accuracy, precision y recall, frente a otro que prioriza sensibilidad (recall) a costa de más falsos positivos, y un tercero que se muestra más conservador y estable. El ranking por métrica y la leyenda interactiva permiten ordenar y resaltar cada serie, dejando claro en qué métrica destaca cada modelo y dónde se rezaga.

El heatmap modelos×métricas y las matrices de confusión por modelo ayudan a interpretar los patrones de error: el modelo con mayor recall tiende a incrementar FP, el más balanceado reparte mejor TP/TN, y el más ligero conserva una precisión aceptable con menor complejidad. Los tooltips y la selección de métrica facilitan contrastar comportamientos y decidir según el objetivo operativo (equilibrio global vs. máxima detección).

Integración de Modelos en la Aplicación



Demostración de predicción de modelos, sobre evaluación si hay existencia de violación de Community Rules. En este caso, los 3 modelos predijeron que no hubo infracción.



En este caso, se observa que 3 modelos predijeron que si hubo infracción con incumplimiento de las reglas de community rules.

Discusión de Resultados

El análisis comparativo evidencia un rendimiento muy parejo entre los enfoques evaluados, con accuracies entre 0.75 y 0.77, lo que sugiere que la tarea posee señales léxicas y contextuales suficientemente informativas para que distintos paradigmas (Transformers y métodos clásicos) resulten competitivos. Al mismo tiempo, la convergencia de métricas indica la existencia de casos límite y ambigüedades semánticas que imponen un "techo" cercano al 77% sin ajustes adicionales (p. ej., calibración, ensambles o enriquecimiento de datos). Este patrón es coherente con el EDA debido a que la variable objetivo está casi balanceada, los textos violatorios tienden a ser más extensos y los subreddits con reglas más estrictas concentran más infracciones, lo que invita a considerar el contexto comunitario en la modelación.

Dentro de los modelos mostrados en el dashboard (como se pueden observar en las imagenes), DeBERTa-v3-small se posiciona como el más equilibrado, con accuracy ≈ 0.77 y métricas macro/ponderadas consistentes entre clases. Su arquitectura con atención "desacoplada" y mejoras de preentrenamiento se traduce en un entendimiento contextual suficiente para capturar patrones sutiles sin requerir un costo excesivo.Ahora bien. En el caso de DistilBERT pese a ser una variante compacta, ofrece un rasgo operativo valioso. Un recall elevado para la clase positiva (≈ 0.88) a costa de menor recall en la clase negativa (≈ 0.62). En escenarios donde el costo de un falso negativo es alto (dejar pasar una violación), este sesgo hacia "detectar más" puede ser preferible, siempre que se acompañe de ajuste de umbral y calibración para contener falsos positivos en producción.

Por su parte, TF-IDF+LinearSVC confirma que los métodos clásicos siguen siendo una base sólida ya que obtuvo un accuracy ≈ 0.76, en donde proporciona estabilidad, interpretabilidad y despliegue sencillo, lo cual es valioso cuando los recursos son limitados o se busca una primera línea de filtrado de bajo costo. El modelo BiLSTM+Attention (apoyado en GloVe) fue entrenado y mostró un buen perfil de recall y F1, pero por prioridades de tiempo se decidió no incluirlo en el dashboard final para concentrar esfuerzos en una interfaz clara y completa. Aun así, sus resultados respaldan la idea de que los patrones secuenciales y léxicos siguen aportando, especialmente cuando se optimiza hacia sensibilidad.

Desde la perspectiva operativa, la selección del "mejor" modelo depende de la política de riesgo y del objetivo de negocio ya que si se requiere equilibrio general y consistencia entre clases, DeBERTa-v3-small es un candidato natural; pero por el otro lado si lo crucial es minimizar falsos negativos, DistilBERT es atractivo por su alta sensibilidad; y si se priorizan costos y explicabilidad, TF-IDF+SVC sirve como línea base o filtro de primera etapa. Además, se abren oportunidades de mejora con cascadas (p. ej., TF-IDF+SVC como filtro rápido y DeBERTa para casos ambiguos), ensambles ponderados para estabilizar F1, calibración de probabilidades y análisis de errores enfocado en patrones difíciles (negaciones, ironía, falta de contexto).

En el entrenamiento, incorporar fine-tuning eficiente (LoRA/QLoRA) sobre DeBERTa, aprendizaje contrastivo comentario↔regla y few-shot para reglas no vistas; asimismo seleccionar el punto de operación por costos (ROC/PR con matriz de costos) y calibrar probabilidades (Platt/Isotónica con reliability diagrams), incluso con umbrales por subreddit.

Conclusion

Como conclusión del proyecto, el dataset dado está limpio y moderadamente balanceado; los comentarios infractores son en promedio más largos y usan lenguaje directo (p. ej., consultas legales), y ciertos subreddits concentran más violaciones, evidenciando sesgos de dominio que los modelos deben manejar con contexto y umbrales por comunidad.

En resultados, DeBERTa-v3-small se ubica en el frente de Pareto por equilibrio entre precisión y recall (≈0.77–0.78 macro/weighted), DistilBERT queda muy cerca con una sensibilidad superior a la clase positiva (útil cuando el costo de falsos negativos es alto), y TF-IDF+LinearSVC ofrece una línea base ligera y competitiva con interpretabilidad y bajo costo. La diferencia principal se observó en el trade-off Precisión/Recall por clase, coherente con la naturaleza contextual de la tarea; el "techo" en ≈0.77 sugiere límites por ambigüedad semántica/ruido y la necesidad de más contexto y calibración.

Se cumplieron el objetivo general y los específicos; ya que se logró entregar un pipeline reproducible (EDA, preprocesamiento documentado, particionado y artefactos), se compararon modelos con capacidad de generalizar a reglas no vistas y se habilitó una aplicación interactiva.

Referencias

- *Jigsaw - agile community rules classification*. (s/f). Kaggle.com. https://www.kaggle.com/competitions/jigsaw-agile-community-rules/overview
- Kabir, S. (2024, 14 diciembre). *Understanding Reddit's Rules and Guidelines - AUQ.io*. AUQ.io. https://auq.io/knowledge-base/reddit-rules/
- *Everything in Moderation*. (s. f.). New America. https://www.newamerica.org/oti/reports/everything-moderation-analysis-how-internet-platforms-are-using-artificial-intelligence-moderate-user-generated-content/case-study-reddit/

Anexo

- **Enlace Repositorio:** https://github.com/Sofiamishel2003/Jigsaw_Agile_Community_Rules_Classification

- **Enlace Presentación:**

https://www.canva.com/design/DAG4-OF-HgA/I-IGdSBFZL_vrgDgwJ4dKw/edit?utm_content=DAG4-OF-HgA&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton

- **Enlace Documento:**
  https://uvggt-my.sharepoint.com/:w:/g/personal/vel22049_uvg_edu_gt/IQBr9095XVJ8RLI14e9hkuhyASKhVKpZ1ws9lmbi1OkUgfY?e=Xbi3I1