



Proyecto Aplicado I

Grupo Nro 2: Aplicación sobre Detección de Fake News

Grupo 2

Integrantes:

1. Christian Camilo Quintana Muñoz
2. Jose Pablo Gonzalez Campos
3. Marcos Irving Mera Sanchez
4. Pablo Andrés Colinas Espina
5. Robinson Elías Maqui Canaviri

Sección 1: Motivación y Justificación General

1.1 Problema a Resolver:

- **Desafío de las Fake News:** En la era digital, la propagación rápida de noticias falsas representa un desafío significativo para la sociedad, afectando la opinión pública y la integridad de los procesos democráticos. La capacidad de identificar y filtrar noticias falsas con eficacia es crucial para mantener informados a los ciudadanos con datos verídicos.

1.2 Acciones Previas:

- **Investigaciones Anteriores:** Se han implementado diversas estrategias para combatir las noticias falsas, desde la verificación manual hasta el empleo de sistemas automáticos basados en el procesamiento de lenguaje natural (NLP) y análisis de imágenes.
- **Limitaciones:** Estas soluciones han enfrentado limitaciones en cuanto a escalabilidad, precisión y la habilidad de mantenerse al día con las técnicas sofisticadas empleadas para crear fake news.

1.3 Solución Propuesta:

Proponemos soluciones completas y múltiples, desde la solución clásica (NLP) para solo texto, para solo imágenes y un enfoque híbrido mezclando la relación entre texto e imagen.

- **Solo Texto:** Proponemos diferentes arquitecturas de redes neuronales recurrentes, como LSTM, GRU y RNN simple.



Para cada uno de estas opciones el preprocesamiento de Texto utiliza un tokenizador para convertir texto en secuencias de tokens, manejando tokens fuera del vocabulario con un token especial (<OOV>), se fija un tamaño máximo de vocabulario y una longitud máxima de secuencia de texto. Se realiza un padding de las secuencias para estandarizar su longitud. Para la creación de Embeddings, se opta por utilizar embeddings preentrenados de GloVe, aunque el código permite la extensión a otros tipos como FastText o embeddings one-hot. Se prepara una matriz de embeddings basada en el índice de palabras del tokenizador y los vectores GloVe.

Para la construcción de los modelos, se define y se entrena una serie de modelos secuenciales con diferentes capas de recurrentes (LSTM, GRU, RNN) utilizando las capas de embeddings creadas. Se incluyen capas de Dropout para regularizar y prevenir el sobreajuste. Se define una capa de salida con una activación sigmoidea para la clasificación binaria.

Para el entrenamiento y evaluación, los modelos se compilan y se entrenan con datos de entrenamiento y validación. Se evaluará el rendimiento utilizando métricas como precisión, recall y F1-score. Se guardará el modelo para nuevas mejoras.

- **Enfoque orientado a la Imagen:** Proponemos una Red Neuronal Convolutiva (CNN) con Keras para clasificar imágenes, utilizando capas convolucionales, de agrupamiento, y densas, junto con técnicas como el dropout para evitar el sobreajuste. La arquitectura CNN se justifica teóricamente por su habilidad para detectar patrones jerárquicos en datos visuales y se valida prácticamente a través de una división del conjunto de datos en entrenamiento y validación, usando la precisión y la matriz de confusión como métricas de rendimiento.
- **Enfoque Híbrido:** Proponemos un enfoque híbrido aprovecha el modelo preentrenado CLIP de OpenAI, conocido por su capacidad de entender conjuntamente imágenes y texto. Se añade una capa bilineal para combinar las incrustaciones de texto e imagen, seguida de capas ReLU y lineales. El modelo se congela para mantener los pesos preentrenados en las capas de CLIP y se entrena solamente las capas adicionales. Se utiliza un conjunto de entrenamiento y otro de validación, con la precisión y el AUC-ROC como métricas de evaluación, y se implementa un programador de tasa de aprendizaje para ajustar dicha tasa durante el entrenamiento basándose en el rendimiento de la validación.
- **Validación Rigurosa:** La solución incluirá pruebas extensivas y validación cruzada para asegurar que los modelos sean robustos y generalizables.

La solución final, será expuesta a través de una aplicación que podrá ser utilizada en celulares y computadores.

1.4 Ajuste al Problema:

- **Cobertura Integral:** Al integrar el análisis de texto y de imágenes, nuestro enfoque está diseñado para abordar tanto la desinformación basada en texto como la manipulación visual, lo que representa una solución comprensiva al problema,



independientemente de la opción a realizarlo de hacerlo en forma separada, es decir solo texto o imagen.

- **Mejora Continua:** La infraestructura propuesta de aprendizaje automático permitirá la actualización continua de los modelos para adaptarse a nuevas tácticas empleadas en la creación de fake news.

1.2. **Justificación de los Experimentos:**

- **Relevancia Empírica:** La justificación de los experimentos reside en datos empíricos que indican que una combinación de NLP y análisis de imagen mejora la detección en comparación con enfoques que consideran solo una dimensión.
- **Validación de la Hipótesis:** Cada experimento está diseñado para probar una hipótesis específica sobre la detección de fake news, con el fin de iterar y refinar nuestra solución.

1.3. **Coherencia con el Problema:**

- **Estrategia Basada en Evidencia:** Nuestra estrategia se basa en una revisión exhaustiva de la literatura y evidencia que sugiere la eficacia de los métodos propuestos. Si bien existen nuevas técnicas que pueden tener una mayor eficacia al momento de detectar los Fake news, para efectos del curso con estas son más que suficientes.
- **Impacto Potencial:** Con este enfoque, nuestro objetivo es desarrollar un software que pueda ser desplegado en plataformas de medios de comunicación o para ser consumida por el público en general a través del celular o PC para ayudar a mitigar la propagación de información engañosa.

Sección 2: Análisis Exploratorio de datos

2.1 Resumen General del Dataset:

- **Cantidad de Datos:** De un total de 56.400 registros disponibles con data para este proyecto. Se descargaron en forma correcta 50749 imágenes, el resto de imágenes no existían o presentan problemas de seguridad haciéndola difícil de descargar en forma automática. De este total existen 20475 etiquetados como noticias reales y 30274 como noticias falsas.
- **Fuentes de Datos:** Los datos provienen de diversas plataformas de noticias y redes sociales, verificados por agencias de fact-checking.

2.2 Distribución de Datos:

- **Categorías de Noticias:** Se observa una distribución equitativa de categorías como política, economía, salud, entre otras.



- **Distribución Temporal:** Las noticias no tienen una fecha temporal, ni nada que nos permita determinar su temporalidad.
- **Balance de Clases:** Se cuenta con un dataset desbalanceado con cantidades mayores de noticias falsas.

2.3 Análisis de Texto:

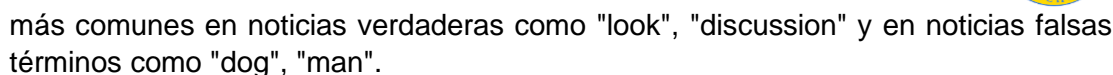
- **Longitud del Texto:** Las estadísticas descriptivas de la cantidad de palabras en los títulos de noticias, nos da los siguientes valores:

count	50749.000000
mean	7.420737
std	5.546227
min	1.000000
25%	3.000000
50%	6.000000
75%	10.000000
max	96.000000

Se tiene las siguientes características encontradas:

- Hay 50.749 títulos en el conjunto de datos analizado.
- En promedio, cada título tiene 7,4 palabras.
- La desviación estándar es 5.5 palabras, indicando dispersión moderada en la longitud de los títulos.
- El título más corto tiene 1 palabra, mientras que el más largo tiene 96 palabras.
- El 50% de los títulos (percentil 50 o mediana) tienen 6 palabras o menos.
- El 75% de los títulos tienen 10 palabras o menos.
- Comparando media (7.4) y mediana (6), parece haber asimetría positiva, con más títulos de longitud menor que la media, pero existen algunos outliers muy largos que elevan la media. La dispersión no es muy alta, concentrándose las longitudes alrededor de 6 a 10 palabras principalmente.

- **Palabras más Frecuentes:** Después de realizar limpieza de texto, remover datos nulos, eliminar caracteres especiales y aplicar lematización se identifican los términos



look, discussion, man,new, one, cat, found



discussion, man, happy, first, cutout, guy, new, cat





Los bigramas más comunes en las noticias falsas de este conjunto de datos parecen ser nombres, eventos o temas que son significativos y relevantes. Por ejemplo, "donald trump", "adolf hitler", "white house", "civil war", "soviet union", y "elon musk" son todos referentes a personas o términos que están comúnmente presentes en el contexto de las noticias.

Algunos bigramas como "look like", "little guy", "first time", "happy see" y "dont know" podrían formar parte de citas directas o expresiones comunes en el lenguaje de las noticias.

El bigrama que ocurre con mayor frecuencia es "elon musk", seguido de "ice cream" y "date unknown", lo cual podría sugerir que estas noticias tienen una cobertura significativa sobre la figura de Elon Musk o temas relacionados con él, así como asuntos que incluyen fechas no especificadas o indeterminadas y, curiosamente, menciones frecuentes a "ice cream" que podría ser un término utilizado en algún contexto específico o simplemente indicar una cobertura frecuente de temas menos serios o más ligeros.

La distribución de las frecuencias muestra que hay una disminución relativamente uniforme desde los bigramas más frecuentes hasta los menos frecuentes dentro de estos top 20, lo que sugiere que mientras algunos temas o términos son prominentes, hay una variedad de temas que son mencionados con cierta frecuencia en las noticias verdaderas.

2. Bigramas con Noticias VERDADERAS

Se observan nombres de figuras políticas como "Donald Trump" y temas de actualidad como "climate change", indicando que las noticias verdaderas suelen usar temas y personajes conocidos.

Bigramas como "year old" y "new york" podrían reflejar el uso de descripciones detalladas de personas y lugares para atraer atención.

Frases comparativas como "look like" y "make look" pueden buscar enfatizar la realidad.

Algunos bigramas como "ice cream" parecen temas aleatorios o absurdos, posiblemente buscando viralidad.

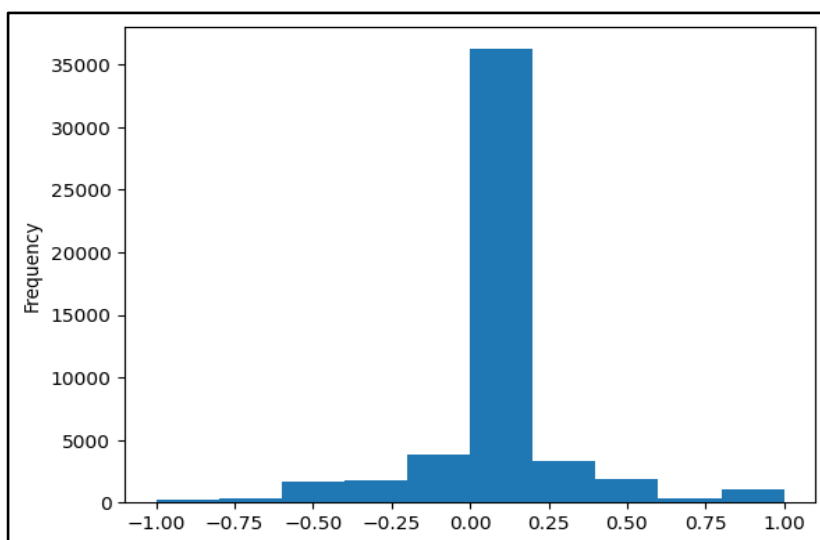
Muchos bigramas son comunes tanto en noticias verdaderas como falsas, lo que dificulta distinguirlas sólo por el lenguaje utilizado.

El bigrama más frecuente "look like" seguido por "Donald Trump" y "year old" sugiere titulares con comparaciones impactantes y figuras conocidas para atraer atención.

• **Errores Comunes:** Las noticias falsas presentan mayor cantidad de errores ortográficos y gramaticales.



- **Análisis de Sentimiento:** Las noticias falsas tienden a tener un tono más emotivo o alarmista.



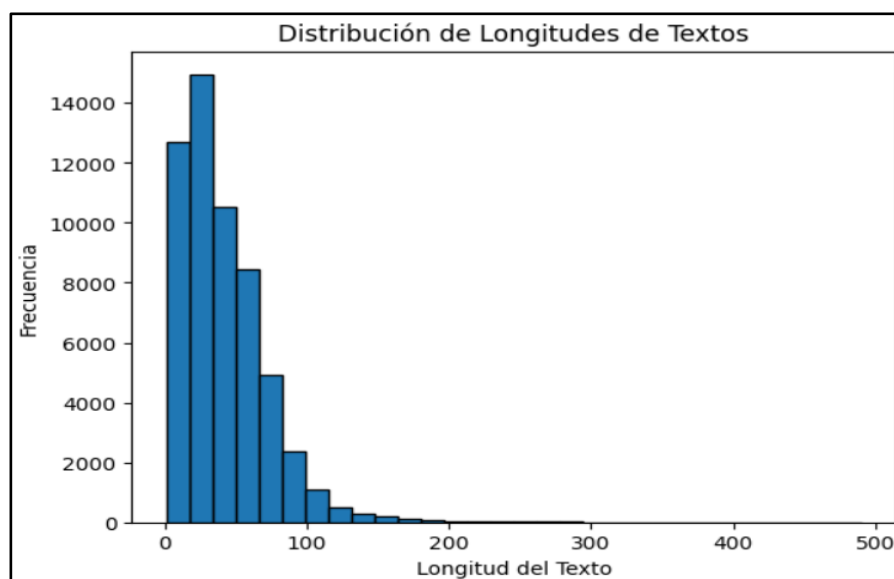
Histograma del análisis de sentimiento del dataset.

Interpretando estos valores junto con el histograma:

Aproximadamente el 78.21% de los textos han sido clasificados como negativos.

Aproximadamente el 21.78% de los textos han sido clasificados como positivos.

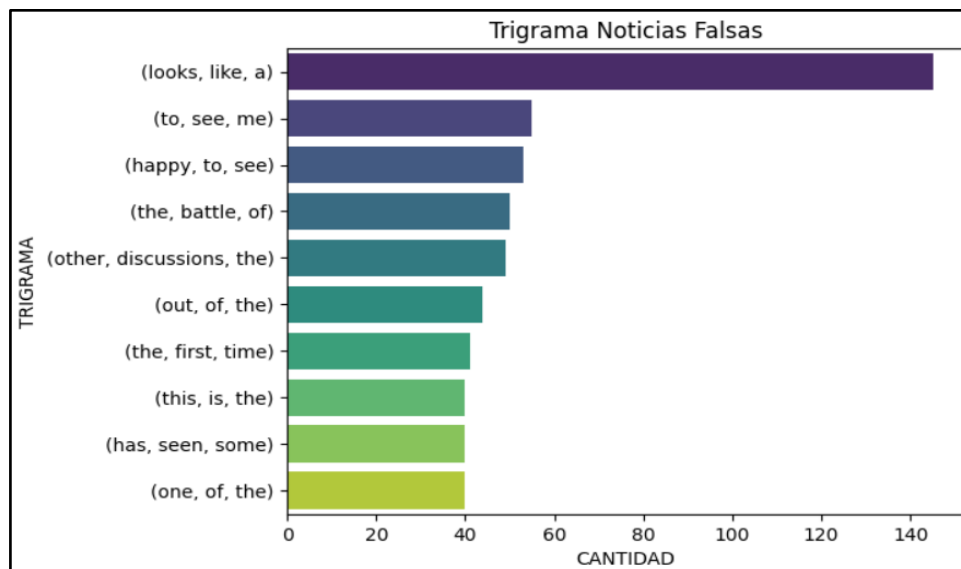
La gran mayoría de los textos son neutrales, lo que sugiere que podrían ser informativos en lugar de opiniones.



Distribución de longitudes de textos del dataset.



En el presente cuadro, vemos que los enunciados presentados son cortos por lo que se hace referencia en mayor cantidad hacia menor cantidad de longitud de texto en el presente dataset.



Análisis del Trigrama de noticias falsas.

En el presente cuadro, se muestran los trigramas de noticias falsas el cual se muestran las palabras consideradas en el análisis de los bigramas los cuales se presentan una combinación de verbos, sustantivos y conjunciones, donde mayormente se muestra las palabras **looks, like, a** que mayormente hacen relación a enunciados de objetos y gustos por personas u objetos.

2.4 Análisis de Imágenes:

- **Dimensiones de Imágenes:** La mayoría de las imágenes tienen resoluciones de 326 x 426 píxeles que son 6728 imágenes.
- **Análisis de Contenido:** Las imágenes en noticias falsas frecuentemente incluyen montajes o alteraciones detectadas por software de análisis de imágenes.

2.5 Correlaciones entre Texto e Imagen:

- **Consistencia:** Se analizará la coherencia entre el contenido del texto y la imagen, estos valores lo entregarán el NLP como visión por computadora, para nuestro caso tenemos una arquitectura que se encargará de obtener estos resultados.
- **Análisis Multimodal:** Se utilizará análisis de sentimiento y detección de objetos para evaluar la congruencia entre la emoción del texto y el contenido visual.



2.6 Problemas Identificados:

- **Datos Erróneos:** Se detectan instancias con enlaces rotos a imágenes o textos incompletos o carentes.
- **Sesgos en Datos:** Hay un sesgo detectado hacia temas políticos debido a la sobre representación de estos en el dataset y porque son figuras conocidas y que despiertan emociones.

2.7 Cumplimiento con el Negocio:

- **Relevancia:** El análisis demuestra que el dataset es representativo de los tipos de noticias que circulan en la esfera pública, lo cual es relevante para el objetivo de detectar Fake News.
- **Preguntas de Profundización:** ¿Hay patrones temporales en la difusión de noticias falsas? ¿Qué características tienen las noticias falsas en comparación con las noticias reales?

Sección 3: Manejo de datos

En base a lo planificado en el desarrollo del proyecto, se han definido como puntos de partida los siguientes aspectos del desarrollo del modelo arquitectónico de la red neuronal para la detección de fake news.

3.1 Estrategia de División de Datos:

- **Split de Train:** 80% de los datos (40.599 artículos) se están utilizando para el conjunto de entrenamiento. Esta proporción asegura que el modelo tenga suficientes ejemplos para aprender las características complejas de las noticias falsas y reales.
- **Split de Validación:** 10% de los datos (5.075 artículos) se ha propuesto destinar para la validación. Este conjunto permite ajustar hiper parámetros y evaluar la generalización del modelo durante la fase de entrenamiento sin sobreajuste.
- **Split de Test:** El 10% restante (5.074 artículos) se ha propuesto utilizar para el conjunto de test, el cual proporcionará una evaluación imparcial del rendimiento del modelo en datos no vistos.

3.2 Criterios de Separación:

- **Aleatoriedad:** La asignación a cada conjunto se realiza de manera aleatoria para evitar sesgos en la distribución de los datos.
- **Estratificación:** Se estratifica según la etiqueta (noticias reales versus noticias falsas) para mantener la proporción de clases en todos los conjuntos de datos, lo que es crucial para el rendimiento del modelo en un problema balanceado.



3.3 Distribución de Datos:

- **Consistencia Temporal:** Los splits reflejan una distribución temporal similar para garantizar que el modelo aprenda a identificar Fake News en diferentes contextos temporales.

3.4 Justificación de Decisiones:

- **Relevancia con el Problema de Negocio:** El objetivo del modelo de detección de noticias falsas es la identificación de artículos u enunciados de noticias sean falsos o verdaderos, por lo que el modelo desarrollado debe estar bien entrenado y eficaz. Esto significa que debe aprender a identificar patrones en los datos de entrenamiento que son indicativos de noticias falsas.

El modelo debe contemplar la división de datos hacia su entrenamiento y evaluación para evaluar el rendimiento del modelo mediante su conjunto de prueba; por lo que se debe considerar si el modelo se entrena solo en el conjunto de entrenamiento, es posible que aprenda a identificar patrones específicos lo que puede generar que el modelo sea menos eficaz para identificar noticias falsas en datos nuevos. Para mitigar este problema, la división de los datos de entrenamiento en conjuntos de entrenamiento y prueba ayuda a evitar este problema. Al utilizar el conjunto de prueba para evaluar el rendimiento del modelo, podemos asegurarnos de que el modelo pueda generalizar sobre nuevos ejemplos.

- **Adaptabilidad:** El mundo de las noticias está en constante cambio. Es importante que un modelo de detección de noticias falsas se mantenga actualizado con las últimas tendencias de noticias.

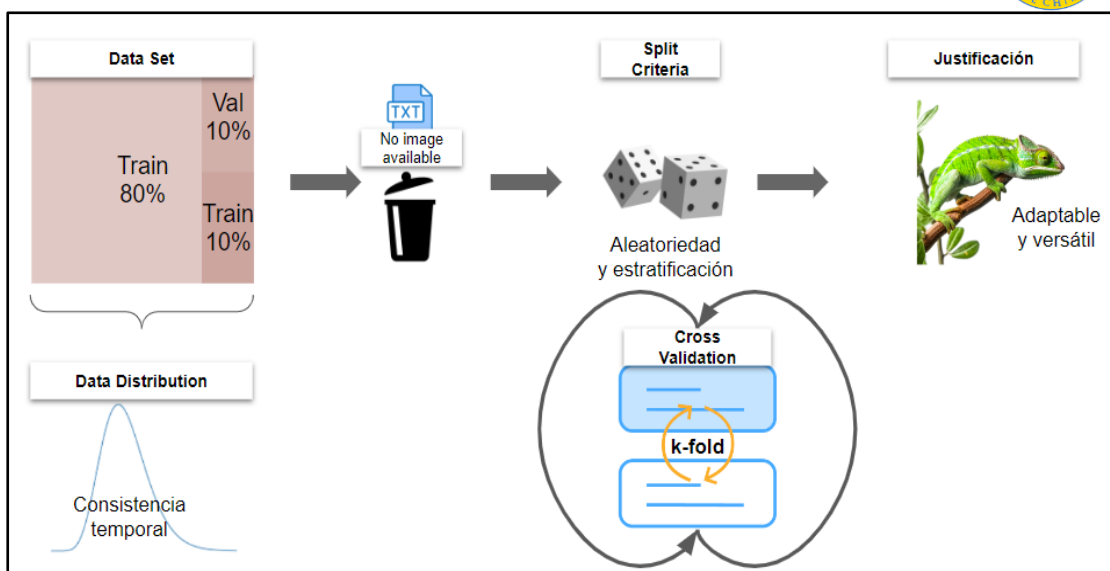
La división de los datos de entrenamiento en conjuntos de entrenamiento y prueba también ayuda a garantizar que el modelo sea adaptable. Esto se debe a que el conjunto de prueba se puede utilizar para identificar nuevos patrones en las noticias falsas.

Si el modelo se entrena solo en el conjunto de entrenamiento, es posible que no pueda identificar nuevos patrones en las noticias falsas. Esto puede hacer que el modelo sea menos eficaz para detectar noticias falsas a medida que cambian las tendencias de noticias.

La división de los datos de entrenamiento en conjuntos de entrenamiento y prueba ayuda a evitar este problema. Al utilizar el conjunto de prueba para identificar nuevos patrones, podemos actualizar el modelo para que sea más eficaz en la detección de noticias falsas.

- **Uso de K-Fold para la validación cruzada:** Se implementará una validación cruzada de 5-fold en el conjunto de entrenamiento para una validación más robusta y para minimizar el sobreajuste.

A continuación, mostramos el esquema usado en el manejo de datos:



Modelo de representación del manejo de datos del proyecto.

Sección 4: Modelos utilizados

Como nuestra solución contempla los 3 casos posibles:

- A. Solo texto.
- B. Solo imagen.
- C. Texto e Imagen.

Proponemos 3 arquitecturas de soluciones diferentes con diferentes modelos cada uno.

4.1 Arquitectura sólo Texto:

En las primeras semanas de desarrollo acerca de la arquitectura sólo texto se emplearon embedding en la capa de entrada de forma pre entrenada donde se aplica una capa SimpleRNN de forma recurrente, seguido por capas de Dropout y una capa Densa con activación ReLU, cuya capa de salida es una capa Densa con activación sigmoide. Este modelo de arquitectura ha generado 23365 parámetros entrenables, de acuerdo a lo mostrado en la ejecución del modelo:



```
Model: "sequential_7"
```

Layer (type)	Output Shape	Param #
embedding (Embedding)	(None, 100, 100)	20148300
simple_rnn_7 (SimpleRNN)	(None, 100)	20100
dropout_14 (Dropout)	(None, 100)	0
dense_14 (Dense)	(None, 32)	3232
dropout_15 (Dropout)	(None, 32)	0
dense_15 (Dense)	(None, 1)	33

```
=====  
Total params: 20171665 (76.95 MB)  
Trainable params: 23365 (91.27 KB)  
Non-trainable params: 20148300 (76.86 MB)
```

Modelo de la red neuronal de la arquitectura sólo texto.

Al realizar evaluaciones de las primeras entregas del proyecto, nos dimos cuenta que no brindaba los resultados esperados y existía sobreajuste por lo que generó valor de accuracy no esperados en base a las noticias esperadas y de forma consistente en base a la validación del enunciado de texto si es certero o no.

Para una mejor entrega de resultados, se rediseñó el modelo usando la técnica de vectorización TF-IDF (TfidfVectorizer de la librería scikit-learn)

El uso de la capa final usando Regresión Logística permite la clasificación binaria (2 clases), por lo que permite la predicción de la variable "class" a partir del texto contenido en la variable "text".

Para la representación del texto numéricamente y poder entrenar el modelo, se utiliza TF-IDF (Term Frequency - Inverse Document Frequency) a través de TfidfVectorizer. Esto genera vectores numéricos para cada texto de entrada.

El conjunto de datos se divide en conjunto de entrenamiento (75%) y validación (25%) para entrenar y evaluar el modelo.

El modelo Regresión Logística se entrena con los vectores TF-IDF de los textos de entrenamiento y las etiquetas reales ("class").

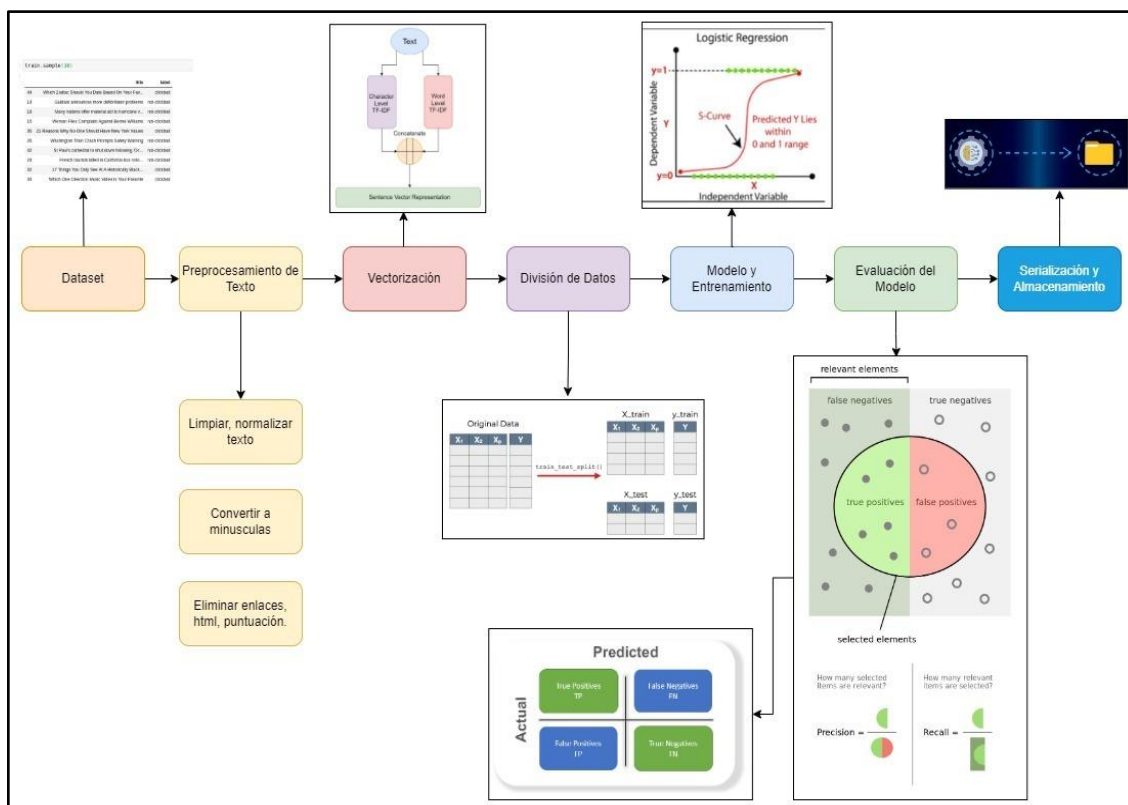
Luego se evalúa el modelo con métricas como precisión, log-loss, AUROC, etc. sobre el conjunto de validación.

Finalmente, se serializa y guarda el modelo entrenado y el vectorizador TF-IDF para usarlos posteriormente en la predicción de nuevos textos.



En resumen, es un modelo de clasificación de textos supervisado bastante estándar, que utiliza Regresión Logística + TF-IDF. El pipeline completo permite predecir una variable categórica binaria a partir de datos textuales.

En el presente gráfico, presentamos el funcionamiento de la arquitectura del modelo sólo texto, se explican las actividades definidas en el modelo donde se destaca el modelo y entrenamiento, la evaluación del modelo donde nos importan las métricas como precisión y recall hasta la serialización y almacenamiento del resultado del modelo.



Nuevo diagrama de la arquitectura sólo texto.

4.2 Arquitectura sólo Imagen:

En las primeras entregas del proyecto, se consideró como punto de partida la aplicación de la red neuronal convolucional (CNN) para la clasificación de imágenes, implementada con Keras. La red neuronal contaba con varias capas:

- Capas de convolución (Conv2D) con una activación ReLU.
- Capas de agrupamiento (MaxPool2D) para reducir la dimensionalidad.
- Capas de regularización (Dropout) para evitar el sobreajuste.
- Capas de aplanamiento (Flatten) para convertir las matrices de características en un vector.



- Capas densas (Dense) para la clasificación, con la última utilizando una activación softmax para la clasificación categórica.

Este modelo de arquitectura generó 29 520 034 parámetros entrenables.

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 124, 124, 32)	2432
conv2d_2 (Conv2D)	(None, 120, 120, 32)	25632
max_pooling2d_1 (MaxPooling2D)	(None, 60, 60, 32)	0
dropout_1 (Dropout)	(None, 60, 60, 32)	0
flatten_1 (Flatten)	(None, 115200)	0
dense_1 (Dense)	(None, 256)	29491456
dropout_2 (Dropout)	(None, 256)	0
dense_2 (Dense)	(None, 2)	514
Total params: 29,520,034		
Trainable params: 29,520,034		
Non-trainable params: 0		

Modelo de la red neuronal de la arquitectura sólo imagen.

En el presente gráfico, nos sirve para explicar los nuevos ítems añadidos hacia la arquitectura de sólo imagen, cuya punto de partida es el dataset que pasa hacia el proceso de preprocesamiento de la imagen donde tiene puntos particulares donde se incluyen actividades como la conversión de la imagen a formato de análisis de nivel de error denominado ELA (Es una técnica para identificar manipulaciones en imágenes al revelar diferencias en los niveles de compresión), luego se realiza la redimensión de imágenes donde se ajustan el tamaño de todas las imágenes para ser procesados por la red neuronal, luego se aplica la normalización de píxeles de las imágenes divididos por 255 ya que dicha función permite a convertir los píxeles hacia 0 y 1 para tener un entrenamiento eficiente y estable. Luego de ello se realiza la división de datos para separar los datos en conjuntos de entrenamiento, validación y prueba (train, test, split),

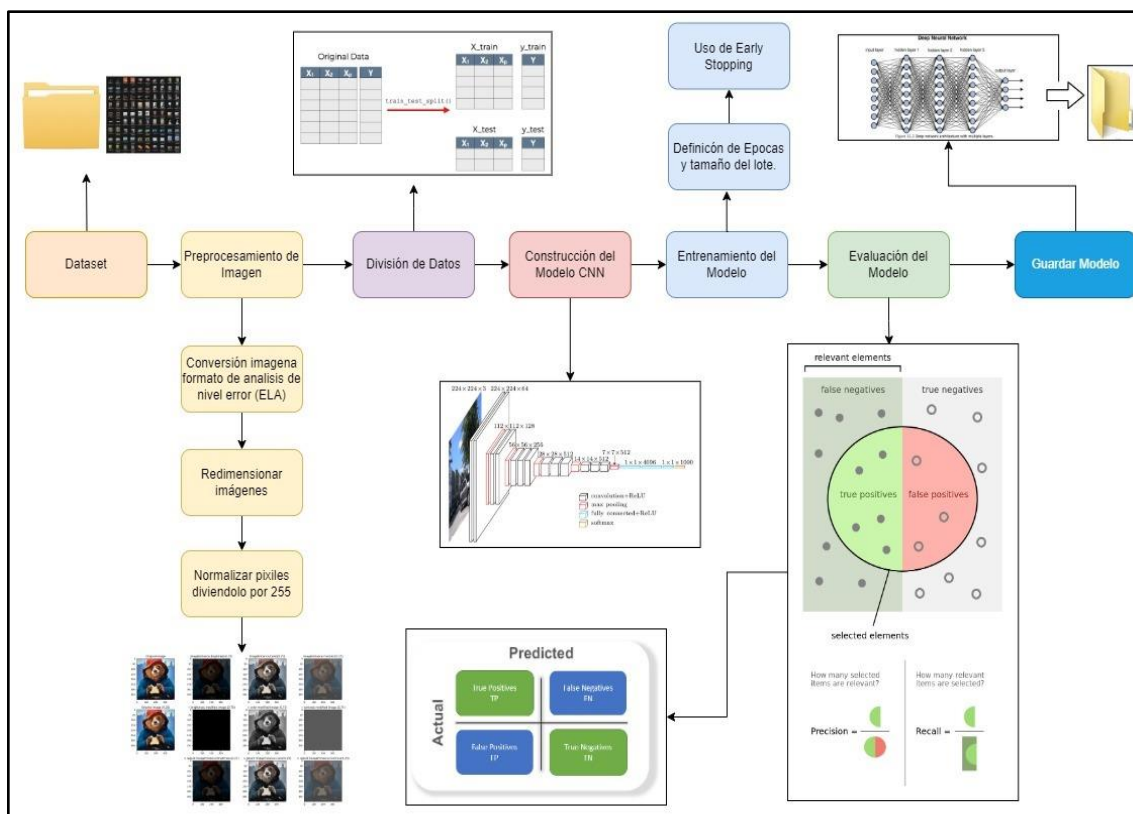
Tener en cuenta que el corazón del modelo es la construcción mediante el uso de CNNs y el uso del early stopping cuya técnica nos sirve para detener el entrenamiento cuando el rendimiento en el conjunto de validación deja de mejorar, lo que previene el sobreajuste.

Asimismo, la definición de épocas y tamaño del lote nos sirve como configuración de los parámetros del entrenamiento, donde una época representa una iteración completa a través del conjunto de datos y el tamaño del lote es la cantidad de muestras procesadas antes de actualizar el modelo, lo que permite que el entrenamiento del modelo aprende a clasificar las imágenes o a predecir resultados a partir de ellas.

La evaluación del Modelo se centra en la evaluación de las métricas de la matriz de confusión para evaluar el modelo teniendo como conceptos importantes a evaluar como



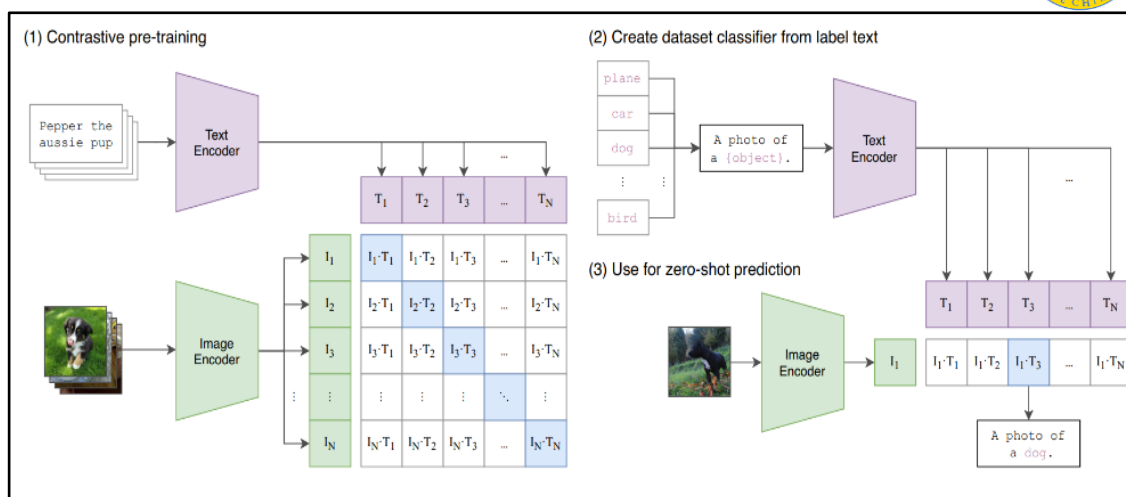
la precisión y recall para evaluar la calidad de los modelos de clasificación. Al finalizar, permite guardar el modelo que obtenga rendimientos satisfactorios para la evaluación de imágenes sean del dataset y pertenecientes a fuentes externas.



Nuevo diagrama de la arquitectura sólo imagen.

4.3. Arquitectura Híbrida de Texto e Imagen:

La arquitectura empleada, es la aplicación de una red neuronal personalizada basada en el modelo CLIP de OpenAI, que integra visión y procesamiento de lenguaje natural (NLP). Usa una capa bilineal para fusionar las incrustaciones (embeddings) de texto e imagen producidas por CLIP, seguida de capas ReLU y lineales para la clasificación final.



Modelo de la arquitectura CLIP.

Este modelo de arquitectura ha generado 285 758 722 parámetros entrenables.

Layer (type:depth-idx)	Output Shape	Param #
ClassificationModel	[1, 1]	--
CLIPModel: 1-1	[1, 50, 768]	1
CLIPVisionTransformer: 2-1	[1, 768]	--
CLIPVisionEmbeddings: 3-1	[1, 50, 768]	2,398,464
LayerNorm: 3-2	[1, 50, 768]	1,536
CLIPEncoder: 3-3	[1, 50, 768]	85,054,464
LayerNorm: 3-4	[1, 768]	1,536
CLIPTextTransformer: 2-2	[1, 512]	--
CLIPTextEmbeddings: 3-5	[1, 77, 512]	25,336,320
CLIPEncoder: 3-6	[1, 77, 512]	37,828,608
LayerNorm: 3-7	[1, 77, 512]	1,024
Linear: 2-3	[1, 512]	393,216
Linear: 2-4	[1, 512]	262,144
Bilinear: 1-2	[1, 512]	134,218,240
ReLU: 1-3	[1, 512]	--
Linear: 1-4	[1, 512]	262,656
ReLU: 1-5	[1, 512]	--
Linear: 1-6	[1, 1]	513
Total params: 285,758,722		
Trainable params: 285,758,722		
Non-trainable params: 0		
Total mult-adds (M): 399.00		
Input size (MB): 0.60		
Forward/backward pass size (MB): 84.07		
Params size (MB): 1143.03		
Estimated Total Size (MB): 1227.70		

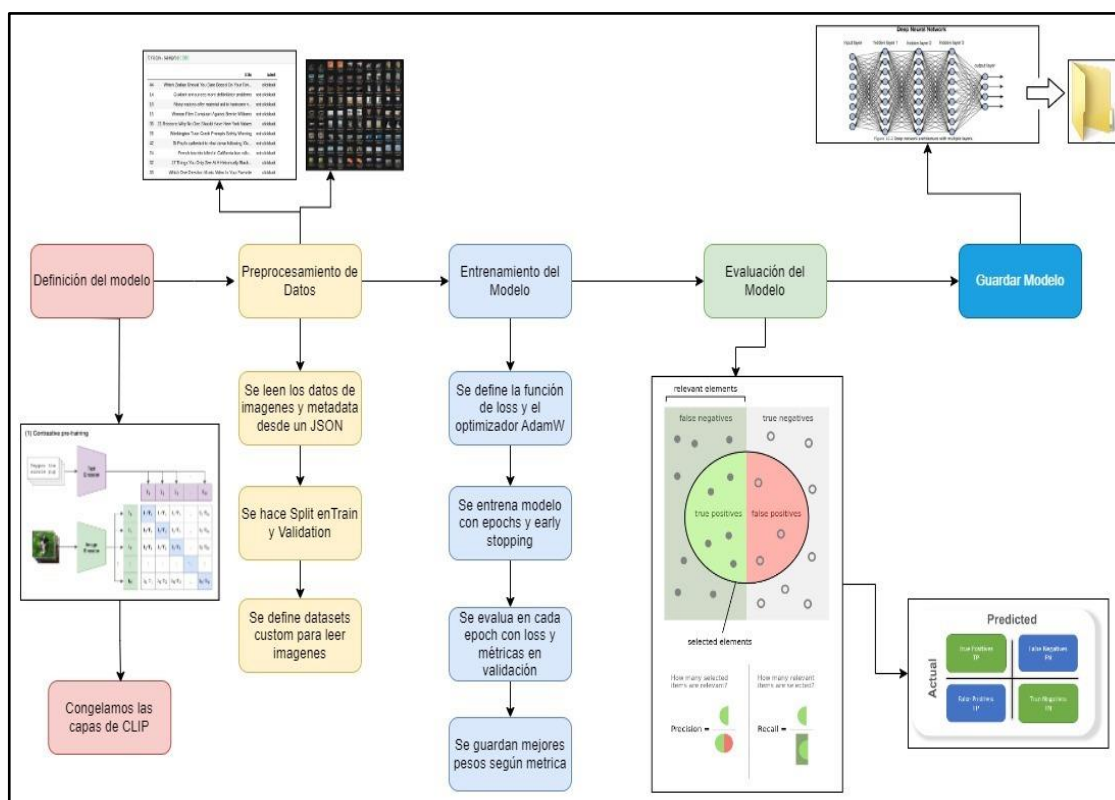
Modelo de la red neuronal de la arquitectura sólo imagen.

Tomar en cuenta que el modelo CLIPModel de Open AI, es un modelo preentrenado que procesa y entiende simultáneamente imágenes y texto; lo que usa la clase ClassificationModel personalizada que permite la la incrustación tanto de texto como imágenes hacia su procesamiento y aplicación de la clasificación binaria. Este modelo ha demostrado ser eficaz para entornos ya que la tarea de clasificación texto-imagen ya que congela el entrenamiento de las capas de CLIP para mantener sus pesos pre entrenados y luego solo permite el entrenamiento en capas recién agregadas.



En base a lo trabajado en el diseño de la arquitectura, luego de haber definido el uso de CLIP, en el preprocesamiento de datos se realizan las lecturas de los datos de imágenes y metadatos desde un JSON donde se extraen los datos necesarios para el entrenamiento del modelo; la aplicación de split en la división de datos para su entrenamiento y validación del rendimiento. Un punto clave es la definición de datasets custom para la carga y procesamiento de imágenes de forma adecuada.

Durante el entrenamiento del modelo, se definen la función de pérdida o loss y el optimizador AdamW para el ajuste de los pesos del modelo. La evaluación se realiza en cada época para medir el rendimiento del modelo utilizando la función de pérdida y otras métricas en el conjunto de validación. Se guardan los mejores pesos obtenidos, se procede con la evaluación del modelo utilizando las métricas de rendimiento para su almacenamiento para su futuro uso en la API. Tener en cuenta que se usan dos métricas importantes como precisión y recall, que son indicadores comunes del rendimiento de un modelo de clasificación.



Nuevo diagrama de la arquitectura híbrida.



Sección 5: Métricas

En esta sección se desarrollan la definición y los objetivos alcanzables para la medición de la evaluación de los modelos propuestos en el presente proyecto:

5.1 Definición de Métricas propuestas para la evaluación de los modelos:

- **Precisión (Accuracy):** La proporción de noticias correctamente identificadas (tanto verdaderas como falsas). El objetivo es alcanzar una precisión de al menos el 90% (para el modelo híbrido) para garantizar que la mayoría de las noticias sean clasificadas correctamente. Para los otros casos de modelos, nos conformamos con un 60%.
- **Recall (Sensibilidad):** La proporción de noticias falsas que el sistema identifica correctamente. Un valor de recall alto minimiza el riesgo de que las fake news pasen desapercibidas. Establecemos un objetivo de recall de al menos el 80%.
- **Precisión (Precisión):** De las noticias identificadas como falsas, cuántas realmente lo son. Una precisión alta es esencial para evitar la clasificación errónea de noticias legítimas como falsas. Nuestro objetivo es una precisión de al menos el 90%.
- **F1-Score:** Una medida que combina precisión y recall en una única métrica, útil cuando se desea buscar un equilibrio entre ambas. Apuntamos a un F1-Score por encima de 0.85.

5.2 Objetivos y Justificación:

- Los objetivos establecidos para la evaluación del modelo son alcanzables hacia el enfoque híbrido propuesto para el análisis de noticias, no así para las otras propuestas.
- La precisión y el recall son particularmente importantes para aplicaciones en la vida real donde el costo de un falso negativo (dejar pasar una fake news) y de un falso positivo (bloquear una noticia verdadera) son significativos.
- El F1-Score es crucial cuando se necesita un balance entre la precisión y el recall, lo que suele ser el caso en la detección de fake news.

5.3 Coherencia con el Problema a Resolver:

- Estas métricas reflejan directamente la capacidad del modelo para funcionar bien en el mundo real, donde el costo de errores es alto tanto social como económicamente.



- A través de la optimización de estas métricas, aseguramos que el modelo no sólo es teóricamente sólido, sino que también es práctico y aplicable en entornos dinámicos y a gran escala, que es donde las fake news tienen el impacto más perjudicial.

Sección 6: Análisis de Resultados

En esta sección hemos considerado los siguientes análisis de las matrices de confusión realizados de la evaluación de cada modelo bajo las 3 arquitecturas desarrolladas:

6.1 Resultados obtenidos del diseño e implementación de la Arquitectura sólo Texto:

Basándonos en la rúbrica proporcionada, vamos a describir y analizar el modelo de clasificación de texto y los resultados obtenidos del código proporcionado.

Descripción del Modelo Completo:

El modelo es un clasificador de texto basado en regresión logística. El flujo de trabajo es el siguiente:

1. Lectura y Preprocesamiento de Datos:

- Los datos se leen desde un archivo CSV y se eliminan varias columnas que no son necesarias para la clasificación.
- Se renombran las columnas relevantes a 'text' y 'class'.
- Se limpia el texto mediante la función **wordopt** que realiza operaciones como convertir a minúsculas, eliminar URLs, etiquetas HTML, puntuación y números.

2. Vectorización:

- Se utiliza **TfidfVectorizer** para convertir el texto a una representación numérica que puede ser utilizada por el modelo de aprendizaje automático. Este proceso convierte el texto en un conjunto de características TF-IDF.

3. Modelo de Clasificación:

- El modelo de clasificación utilizado es la regresión logística.
- El modelo se entrena con los datos de entrenamiento vectorizados.

4. Evaluación del Modelo:

- Se evalúa el modelo utilizando la precisión (**score**).
- Se genera un informe de clasificación, que incluye medidas como precisión, recuperación y puntuación **F1** para cada clase.
- Se calcula la pérdida logarítmica (**log_loss**) y el área bajo la curva ROC (**roc_auc_score**) para las predicciones probabilísticas del modelo.

5. Persistencia del Modelo:

- El modelo y el vectorizador se guardan en el disco utilizando **joblib**.



Análisis de Resultados:

De acuerdo con la rúbrica, para una evaluación completa de 5 puntos en cada categoría, debemos asegurarnos de lo siguiente:

1. Descripción del Modelo:

- Todos los componentes del sistema de IA (modelo, vectorizador, preprocesamiento) están claramente descritos y se entiende el propósito de cada uno.
- Las métricas asociadas (precisión, recuperación, puntuación F1, pérdida logarítmica y **AUROC**) están definidas y se explica cómo se utilizan para entrenar y evaluar el modelo.

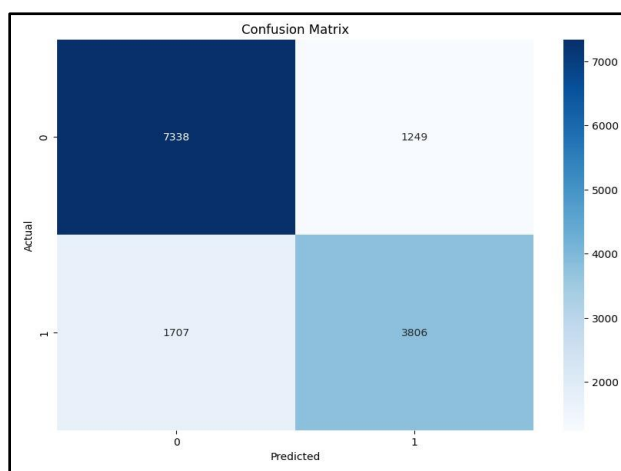
2. Análisis de Resultados:

- Los errores de clasificación deben ser identificados y analizados. Esto incluye examinar los falsos positivos y falsos negativos.
- El informe de clasificación proporcionado por **classification_report** puede ayudar a identificar qué clase se predice bien y cuál no.
- Debe explicarse cuál es el problema más importante (por ejemplo, si un falso positivo es más crítico que un falso negativo, dependiendo del contexto de la aplicación) y cuál es más fácil de abordar.
- Las acciones futuras deben estar justificadas en función de los resultados del análisis.

Por ejemplo, si el modelo tiene muchos falsos positivos, se podrían considerar estrategias para equilibrar las clases o mejorar el preprocesamiento del texto.

6.2 Resultados obtenidos de la evaluación del modelo de la Arquitectura sólo Texto:

Matriz de Confusión:



Matriz de Confusión para modelo de Arquitectura sólo Texto.



- **Verdaderos Negativos:** El modelo identificó correctamente 7338 ítems como noticias verdaderas.
- **Falsos Positivos:** El modelo clasificó incorrectamente 1249 ítems verdaderos como falsos.
- **Falsos Negativos:** El modelo no detectó 1707 ítems de noticias falsas, clasificándolas erróneamente como verdaderas.
- **Verdaderos Positivos:** El modelo identificó correctamente 3806 ítems como noticias falsas.

Interpretación de Métricas calculadas:

- **Precisión:** De todos los ítems que el modelo predijo como noticias falsas, aproximadamente el 75.3% de esas predicciones fueron correctas, lo que indica la calidad de las predicciones positivas del modelo.
- **Sensibilidad:** De todas las noticias falsas reales, el modelo fue capaz de identificar correctamente el 69.0% de ellas, lo que indica la capacidad del modelo para detectar noticias falsas.
- **Especificidad:** De todas las noticias verdaderas reales, el modelo identificó correctamente el 85.5% de ellas, lo que permite la no clasificación errónea de las noticias verdaderas como falsas.
- **Puntuación F1:** La puntuación F1 es una medida armónica de la precisión y la sensibilidad. Un valor de 0.720 sugiere un equilibrio razonable entre precisión y sensibilidad, indicando un rendimiento decente del modelo.

Interpretación general de la matriz de confusión:

El modelo bajo la arquitectura sólo texto presenta un rendimiento decente en la clasificación de noticias como verdaderas o falsas. Sin embargo, se puede mejorar en aumentar la sensibilidad (reduciendo los Falsos Negativos) y la precisión (reduciendo los Falsos Positivos). La relativamente alta especificidad sugiere que el modelo es efectivo en identificar noticias verdaderas, pero podría mejorar en su capacidad para identificar noticias falsas (como se refleja en la sensibilidad).

Se debe considerar que, en la clasificación de noticias falsas, tanto los Falsos Positivos como los Falsos Negativos pueden tener implicaciones serias. Los Falsos Positivos pueden llevar a la censura de información verdadera, mientras que los Falsos Negativos pueden permitir la propagación de desinformación. Por lo tanto, es crucial buscar un equilibrio entre estas métricas, optimizando el modelo para reducir ambos tipos de errores, lo cual se irá trabajando en las siguientes iteraciones del modelo.



6.3 Resultados obtenidos del diseño e implementación de la Arquitectura sólo Imagen:

El modelo es una red neuronal convolucional (CNN) diseñada para el análisis de imágenes, para clasificar imágenes en dos categorías (como indica el '2_way_label'). Vamos a describir el modelo y analizar los resultados según los criterios de la rúbrica proporcionada en la imagen, la cual se enfoca en:

1. Descripción del modelo completo.
2. Análisis de resultados.

Descripción del Modelo Completo:

El modelo utiliza la librería Keras y se estructura de la siguiente manera:

- **Preprocesamiento de Imágenes:**
 - Las imágenes se pre procesan convirtiéndolas a un formato **ELA** (Error Level Analysis), lo que puede ayudar a detectar manipulaciones en las imágenes. Cada imagen se redimensiona a 128x128 píxeles y se normaliza dividiendo por 255.
- **Arquitectura de la Red:**
 - La red se inicia con una capa convolucional **Conv2D** con 32 filtros de tamaño 5x5 y activación ReLU, seguida de una segunda capa convolucional idéntica.
 - Después de las capas convolucionales, hay una capa de agrupación máxima **MaxPool2D** con un tamaño de 2x2 para reducir la dimensionalidad.
 - Se emplea una capa **Dropout** con un ratio de 0.25 para reducir el sobreajuste.
 - La capa **Flatten** convierte los mapas de características en un vector.
 - Una capa densa **Dense** con 256 neuronas sigue al aplanamiento, con una función de activación ReLU.
 - Se agrega otra capa **Dropout** con un ratio de 0.5 para regularización.
 - La última capa es una capa densa con 2 neuronas, una por cada clase posible, utilizando la función de activación softmax para la clasificación.
- **Compilación del Modelo:**
 - Se utiliza el optimizador **RMSprop** con una tasa de aprendizaje de 0.0005.
 - La función de pérdida es la "**categorical_crossentropy**", adecuada para la clasificación de múltiples clases.
 - La métrica de rendimiento es la "accuracy" (precisión).
- **Entrenamiento del Modelo:**
 - El modelo se entrena con un **EarlyStopping** monitorizando la 'val_accuracy' para evitar el sobreajuste.
 - Se define un número de épocas de 30 y un tamaño de lote de 100.
 - Se utilizan datos de entrenamiento y validación con una división del 80-20%.
- **Evaluación del Modelo:**
 - Se utiliza una matriz de confusión para evaluar el rendimiento del modelo en la clasificación.
 - Se generan gráficos para visualizar la pérdida y precisión durante el entrenamiento y la validación.



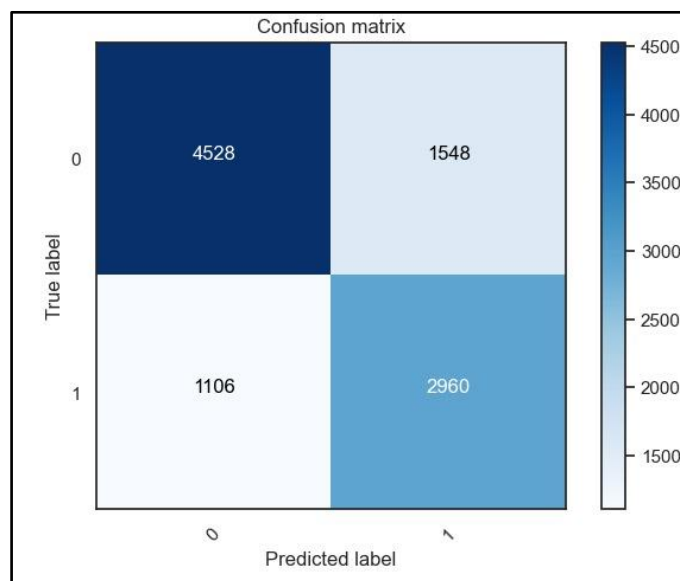
Análisis de Resultados:

El análisis se debe basar en los siguientes puntos:

- **Errores de Tipo I y Tipo II:**
 - Deberíamos examinar los errores de falso positivo y falso negativo en la matriz de confusión para identificar cuál es más prevalente y en qué categoría.
- **Importancia y Facilidad del Problema:**
 - Se debe determinar cuál de los problemas (falso positivo o falso negativo) es más crítico para la aplicación del modelo y cuál podría ser más fácil de solucionar con modificaciones en el modelo o en los datos.
- **Acciones Siguientes:**
 - Se deben proponer pasos futuros basados en el análisis de la matriz de confusión y las curvas de aprendizaje, como ajustar la red, obtener más datos o realizar más ingeniería de características.

6.4 Resultados obtenidos de la evaluación del modelo de la Arquitectura sólo Imagen:

Matriz de Confusión:



Matriz de Confusión para modelo de Arquitectura sólo Imagen.

- **Verdaderos Negativos:** El modelo identificó correctamente 4528 ítems como noticias verdaderas.



- **Falsos Positivos:** El modelo clasificó incorrectamente 1548 ítems verdaderos como falsos.
- **Falsos Negativos:** El modelo no detectó 1106 ítems de noticias falsas, clasificándolas erróneamente como verdaderas.
- **Verdaderos Positivos:** El modelo identificó correctamente 2960 ítems como noticias falsas.

Interpretación de Métricas calculadas:

- **Precisión:** De todos los ítems que el modelo predijo como noticias falsas, aproximadamente el 65.7% de esas predicciones fueron correctas.
- **Sensibilidad:** De todas las noticias falsas reales, el modelo fue capaz de identificar correctamente el 72.8% de ellas.
- **Especificidad:** De todas las noticias verdaderas reales, el modelo identificó correctamente el 74.5% de ellas.
- **Puntuación F1:** La puntuación F1 es una medida armónica de la precisión y la sensibilidad. Un valor de 0.690 sugiere un equilibrio razonable entre precisión y sensibilidad, indicando un rendimiento moderado del modelo.

Interpretación general de la matriz de confusión:

El modelo de la arquitectura sólo imagen, presenta un rendimiento moderado en la clasificación de noticias como verdaderas o falsas. La precisión y la especificidad son relativamente moderadas, lo que indica que hay una cantidad significativa de Falsos Positivos, es decir, noticias verdaderas que se están clasificando incorrectamente como falsas. Asimismo, la sensibilidad es relativamente más alta, lo que implica que el modelo es más efectivo en identificar noticias falsas que en identificar noticias verdaderas correctamente. La puntuación F1, que es una medida del equilibrio entre precisión y sensibilidad, refuerza esta interpretación.

Hay que considerar mediante la clasificación de noticias falsas basada en imágenes, tanto los Falsos Positivos como los Falsos Negativos pueden tener implicaciones serias. Los Falsos Positivos pueden llevar a la censura o desacreditación de información verdadera, mientras que los Falsos Negativos pueden permitir la propagación de desinformación. Por lo tanto, es crucial buscar un equilibrio entre estas métricas, optimizando el modelo para reducir ambos tipos de errores; el cual se realizarán mejoras en la arquitectura del modelo.

6.5 Resultados obtenidos del diseño e implementación de la Arquitectura Híbrida de Texto e Imagen

El modelo descrito en el código proporcionado es un modelo de clasificación personalizado que utiliza el modelo CLIP preentrenado de OpenAI como base. Se trata de una red neuronal que ha sido adaptada para realizar tareas de clasificación binaria. A continuación,



se describe el modelo y se realiza un análisis de los resultados de acuerdo con la rúbrica proporcionada en la imagen.

Descripción del Modelo Completo:

1. Modelo Base:

- **CLIPModel:** Se utiliza el modelo preentrenado CLIP (Contrastive Language-Image Pretraining) que es capaz de entender y asociar texto e imágenes.
- **Congelación de Pesos:** Se congelan los pesos del modelo CLIP durante el entrenamiento para mantener las características aprendidas previamente.

2. Arquitectura de Clasificación:

- **Capas Personalizadas:** Después de las salidas de CLIP (embeddings de texto e imagen), se pasa a través de una capa bilineal, seguida de capas de activación ReLU y capas lineales. La última capa lineal tiene una sola salida, adecuada para la clasificación binaria.
- **Función de Activación:** Se utiliza ReLU para introducir no linealidad en el modelo.
- **Capa de Salida:** La última capa lineal se usa para obtener la predicción final.

3. Procesamiento de Datos:

- **Dataset Personalizado:** Se implementa una clase de conjunto de datos que utiliza el procesador CLIP para procesar texto e imágenes, adecuándose para ser pasados al modelo.
- **Padding Contextual:** Se realiza un padding o recorte al texto para asegurar una longitud uniforme de secuencia.

4. Entrenamiento y Validación:

- Se establecen **hiperparámetros** como la tasa de aprendizaje y la semilla para la reproducibilidad.
- Se utiliza la función de pérdida **BCEWithLogitsLoss**, adecuada para la clasificación binaria.
- Se emplea el optimizador **AdamW** y un programador de tasa de aprendizaje para ajustar la tasa durante el entrenamiento.
- Se utiliza un enfoque de entrenamiento y validación estándar, con los datos divididos en conjuntos de entrenamiento, validación y prueba.

5. Evaluación del Modelo:

- Se calculan métricas estándar como la precisión y el área bajo la curva ROC (AUROC) para evaluar el rendimiento del modelo.

Análisis de resultado obtenido de la ejecución de la implementación del modelo:

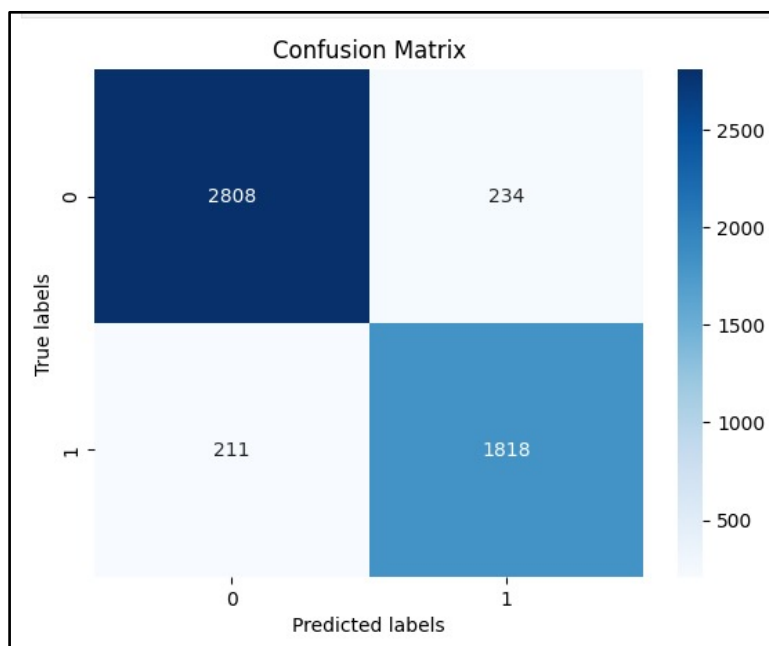
- Se registran las pérdidas y las métricas de precisión y AUROC durante la validación. Se identifica el mejor rendimiento en términos de precisión y AUROC.
- Se toman acciones basadas en el rendimiento del modelo, como guardar el mejor estado del modelo y ajustar la tasa de aprendizaje en función de la pérdida de validación.



- La descripción del análisis sería más completa si se incluyeran detalles sobre los tipos de errores (falsos positivos y falsos negativos) y cómo podrían abordarse en el futuro, como a través de un ajuste más fino del modelo o una mejora en el preprocesamiento de los datos.

6.6 Resultados obtenidos de la evaluación del modelo de Arquitectura Híbrida de Texto e Imagen

Matriz de Confusión:



Matriz de Confusión para modelo de Arquitectura Híbrida.

- **Verdaderos Negativos:** El modelo identificó correctamente 2808 ítems como noticias verdaderas.
- **Falsos Positivos:** El modelo clasificó incorrectamente 234 ítems verdaderos como falsos.
- **Falsos Negativos:** El modelo no detectó 211 ítems de noticias falsas, clasificándolas erróneamente como verdaderas.
- **Verdaderos Positivos:** El modelo identificó correctamente 1818 ítems como noticias falsas.

Interpretación de Métricas calculadas:

- **Precisión:** De todas las instancias que el modelo predijo como noticias falsas, aproximadamente el 88.6% de esas predicciones fueron correctas.



- **Sensibilidad:** De todas las noticias falsas reales, el modelo fue capaz de identificar correctamente el 89.6% de ellas.
- **Especificidad:** De todas las noticias verdaderas reales, el modelo identificó correctamente el 92.3% de ellas. Una alta especificidad sugiere que el modelo es muy bueno en reconocer y no clasificar erróneamente las noticias verdaderas como falsas.
- **Puntuación F1:** La puntuación F1 es una medida armónica de la precisión y la sensibilidad. Un valor de 0.891 indica un excelente equilibrio entre precisión y sensibilidad, lo cual es particularmente importante en aplicaciones donde tanto la identificación correcta de noticias falsas como la no clasificación errónea de noticias verdaderas son críticas.

Interpretación general de la matriz de confusión:

El modelo de la arquitectura híbrida, presenta un rendimiento muy bueno en la clasificación de noticias como verdaderas o falsas. La alta precisión y sensibilidad sugieren que el modelo es efectivo tanto en identificar noticias falsas como en no clasificar erróneamente las verdaderas. La alta especificidad y la puntuación F1, son métricas que permiten respaldar la eficacia general del modelo en este contexto al realizarlo de forma conjunta.

Sección 7: Prototipo de solución

Para el diseño arquitectónico de la solución, se han considerado los siguientes elementos basados en una arquitectura que combina modelos de IA y buenas prácticas de Ingeniería de Software hacia el desarrollo de un API para realizar las consultas respectivas. A continuación, se describen los elementos del flujo de trabajo propuesto por nuestro equipo:

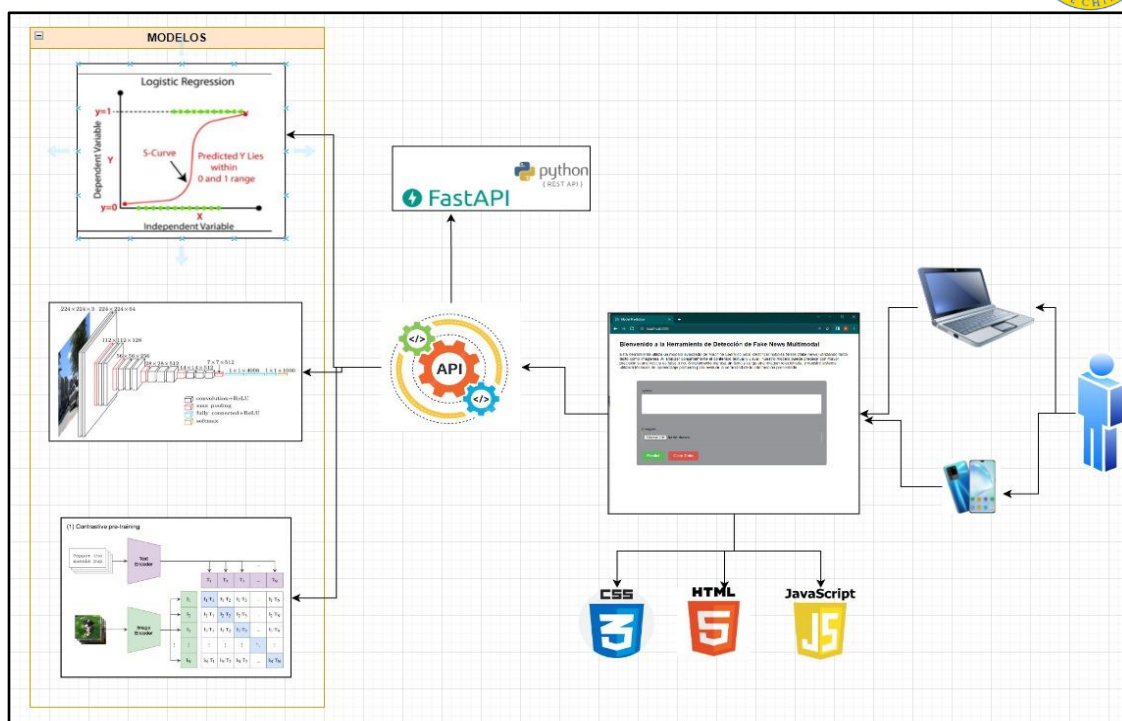


Diagrama de una solución de aprendizaje supervisado aplicado a la detección de noticias falsas (fake news).

Flujo de Trabajo General:

Interfaz de Usuario (UI): Los usuarios interactúan con una página web donde pueden cargar y enviar texto, imágenes, o ambos. Esta interfaz está construida con tecnologías web estándar como HTML, CSS y JavaScript.

Para ser consistente con nuestros datos de entrenamiento, se traspasó la interfaz de la aplicación a sólo inglés, para mantener la consistencia de la misma entre inputs, interfaz y respuesta.

API:

Recepción de consultas: La página web envía las consultas de los usuarios a un API.

Enrutamiento: El API, posiblemente construido con FastAPI (un moderno framework de construcción de APIs con Python), deriva la consulta al modelo respectivo dependiendo del tipo de contenido ingresado (texto, imagen o ambos).

Modelos:

Regresión Logística: Para el texto, se utiliza un modelo de regresión logística, una técnica estadística que predice la probabilidad de un resultado binario (como verdadero o falso). Este modelo puede estar manejando la parte textual de la consulta, es decir, analizando las palabras del texto para determinar si es fake news o no.



Redes Convolucionales (CNN): Para las imágenes, se utiliza una red convolucional. Las CNNs son especialmente buenas para trabajar con datos visuales y pueden identificar patrones y características en las imágenes que podrían indicar manipulación o elementos comunes en imágenes asociadas con noticias falsas.

CLIP Model: Este es un modelo de aprendizaje profundo multimodal que puede entender y relacionar conceptos tanto de texto como de imágenes. CLIP podría estar siendo utilizado para casos donde se ingresa tanto texto como imágenes, proporcionando una evaluación combinada y holística para determinar si la entrada es fake news.

Asimismo, mostramos el prototipo de las interfaces de usuario desarrollados basado en la propuesta de solución del proyecto que consta de un aplicativo web hacia el ingreso de datos sea mediante texto o imagen donde se encuentra embebido con los pesos entrenados usando las 3 arquitecturas para la detección del enunciado si es Fake news o no.

Como vemos en el presente gráfico, se ha considerado las siguientes funcionalidades de usabilidad como:

- La aplicación solo permite ingreso de texto y proceso de carga de imágenes.
- La aplicación no permite ingreso de otras extensiones de archivos para el análisis.
- La aplicación emite resultado, luego de haber realizado análisis.

7.1 Uso del módulo de Arquitectura sólo Texto:

Si el usuario ingresa sólo texto en los campos de entrada, se entrega un resultado de texto que muestra si el enunciado es considerado una noticia falsa o no. Dado que estamos utilizando una regresión logística en este caso, el resultado que obtenemos es una clase y no un porcentaje. A continuación, mostramos la interfaz y un ejemplo para este escenario:



Advanced Fake News Detection Tool

Protect yourself from misinformation and make informed decisions

Our fake news detection tool helps you separate truth from fiction. It analyzes both text and images to more accurately identify fake news. Simply enter text or upload an image, and our system will use deep learning techniques to assess the veracity of the information presented.

Model Prediction

Text:

king femme dirty kitty this is my first time its just for fun please dont be mean

Image:

Seleccionar archivo Sin archivos seleccionados

Predict

Prediction

Rated as Fake News

This tool is ideal for verifying the authenticity of news in a world where misinformation spreads rapidly. Our goal is to provide a quick and reliable way to filter out disinformation and promote greater trust in the media.

Multimodal Fake News Detection Model by VeritasCorp © 2023

Uso del módulo de Arquitectura sólo Texto.

7.2 Uso del módulo de Arquitectura sólo Imagen:

En este escenario, el usuario ingresa solo una imagen en los campos de entrada, donde la aplicación envía este input y el backend selecciona el modelo convolucional. La respuesta corresponde a la clase con el porcentaje de predicción más alto, indicándose con qué grado de certeza se predice la veracidad o falsedad de la noticia. A continuación, mostramos la interfaz y un ejemplo para este escenario:



Advanced Fake News Detection Tool

Protect yourself from misinformation and make informed decisions

Our fake news detection tool helps you separate truth from fiction. It analyzes both text and images to more accurately identify fake news.

Simply enter text or upload an image, and our system will use deep learning techniques to assess the veracity of the information presented.

Model Prediction

Text:

Image:

Seleccionar archivo cd9y6e

Predict

Prediction

Rated as Fake News with 90.07% confidence.

This tool is ideal for verifying the authenticity of news in a world where misinformation spreads rapidly. Our goal is to provide a quick and reliable way to filter out disinformation and promote greater trust in the media.

Multimodal Fake News Detection Model by VeritasCorp © 2023

Uso del módulo de Arquitectura sólo Imagen.

7.3 Uso del módulo de Arquitectura Híbrida de Texto e Imagen

En este último escenario, el usuario ingresa input en formato texto y carga una imagen a la aplicación, donde esta es renderizada. Al enviar al backend, se identifica un input multiple, por lo que se utiliza el modelo multimodal.

Dado que nuestro resultado entrega un valor decimal entre 0 y 1 (haciendo una predicción del label, el modelo entrega la respuesta junto con qué grado de certeza se predice la veracidad o falsedad de la noticia. A continuación, mostramos la interfaz y un ejemplo para este último escenario:



Advanced Fake News Detection Tool

Protect yourself from misinformation and make informed decisions

Our fake news detection tool helps you separate truth from fiction. It analyzes both text and images to more accurately identify fake news. Simply enter text or upload an image, and our system will use deep learning techniques to assess the veracity of the information presented.

Model Prediction

Text:


angry human yelling at peaceful human

Image:

Seleccionar archivo

3gduaq

Predict



Prediction

Rated as Fake News with 90.07% confidence.

This tool is ideal for verifying the authenticity of news in a world where misinformation spreads rapidly. Our goal is to provide a quick and reliable way to filter out disinformation and promote greater trust in the media.
Multimodal Fake News Detection Model by VeritasCorp © 2023

Model Prediction

Text:


Michelle Bachelet is impeached and SuperTanker is appointed President of the Republic

Image:

Seleccionar archivo

michelle.jpeg

Predict



Prediction

Rated as Fake News with 99.13% confidence.

Uso del módulo de Arquitectura híbrida.

Para la entrega final, se va a realizar un análisis para fijar el threshold óptimo en el cual se considera que una noticia es falsa o si es real, apuntando a mejorar nuestro rendimiento. Junto con esto, se va a generar un nuevo dataset de entrenamiento desde internet con noticias e imágenes que el modelo no ha visto y que pueden tener estructura un poco más compleja que el dataset utilizado. De esta manera, podremos realizar una prueba en vivo sometiendo nuestra solución a un caso de uso real.