Análisis de sesgos en predicción de género y raza en imágenes

Hansel Blanco
Lázaro Alejandro Castro Arango
Elena Rodríguez Horta
Daniela Rodríguez Cepero
Karel Camilo Manresa León
Karlos Alejandro Alfonso Rodríguez

1. Introducción

Hace unos años, al hablar de modelos de inteligencia artificial (IA), prácticamente se asumía que en la sociedad se utilizaba en unos contextos muy concretos y exclusivos: grandes empresas tecnológicas, centros de investigación en la materia y entornos similares. Esta situación ha cambiado drásticamente.

En la actualidad los modelos de IA están en los teléfonos móviles, la smart TV, hablando exclusivamente de dispositivos físicos. También están integrados en softwares de correo, en las redes sociales y muchos de los servicios telemáticos que se usan a diario. Fuera del ámbito personal, si se abre el abanico a entornos empresariales e institucionales, los procesos en los que se recurre a modelos generados por algoritmos de aprendizaje automático son incontables.

El creciente impacto de los modelos de aprendizaje automático en la sociedad, así como la variedad de campos en los que son aplicados, ha llamado la atención de la comunidad hacia las consecuencias que estos algoritmos pueden traer para las personas y la sociedad de forma general. En particular, preocupan los efectos que puedan tener en áreas como las de la educación y formación, el empleo, los servicios, la legalidad y los sistemas judiciales. Se ha vuelto necesario que los sistemas de inteligencia artificial sean desarrollados con responsabilidad, teniendo en cuenta aspectos como la imparcialidad, seguridad y privacidad, comprensibilidad honestidad y ecuanimidad resultan esenciales. [**?**].

Algunos gobiernos usan los modelos de aprendizaje automático en la predicción de brotes virales y focos de crímenes. Los sesgos existentes en estos modelos tienen repercusiones trascendentales, lo cual es preocupante por el hecho de que pueden reforzarse o derivar en resultados inesperados [**?**].

El proceso de contratación se ha convertido en un proceso automatizado en muchos casos, desde la selección de currículums hasta la elección de los candidatos. Como consecuencia de la participación cada vez mayor de sistemas automatizados y de aprendizaje automático en esta tarea de gran repercusión para las personas, es necesario controlar cuándo una decisión es injusta [21].

De manera general, en la comunidad existen posiciones diferentes acerca del uso de los algoritmos de aprendizaje automático en la toma de decisiones. Algunos plantean que el uso de algoritmos en la toma de decisiones, en particular en la evaluación de riesgo en escenarios judiciales, debe ser detenido, pues estos modelos son fuentes de incertidumbres y alteraciones [13].

Considerando las definiciones de equidad presentes en la literatura, existen numerosos algoritmos de clasificación que tienen un buen desempeño. Sin embargo, debido a que estos se enfocan en el tiempo presente y dado que incorporan el sesgo humano, existe el riesgo y la propensión a repetir y aumentar dichos sesgos [18].

Teniendo en cuenta los riesgos que supone la utilización de los modelos de aprendizaje automático en la toma de decisiones, es necesario enfrentar los sesgos y prejuicios que contienen para que no se mantengan y extiendan. Es importante implementar conceptos éticos fundamentales en la inteligencia artificial [9] y construir sistemas que se comprometan con los valores sociales [24]. Es decir, determinar cuán dañino puede ser un sesgo y, por tanto, cuáles no pueden ser permitidos en los algoritmos. Es una tarea difícil pero sumamente necesaria por lo que requiere la atención de todos.

2. Estado del arte

Durante los últimos años, con el creciente uso de este tipo de tecnologías, se han ido descubriendo múltiples sesgos de Machine Learning que nos deberían dar qué pensar. Si bien el aprendizaje automático ofrece una fuente de información muy valiosa y dota de herramientas de gran utilidad para comprender el mundo, los sesgos descubiertos podrían resultar contraproducentes para el interés general o beneficiar a algunos en detrimento de otros.

El sesgo en aprendizaje automático, también conocido como sesgo de modelo, es la diferencia entre el valor medio predicho por el modelo y el valor medio real, aparece cuando un modelo produce resultados erróneos de forma sistemática. La aparición de estos es debida a que los modelos son desarrollados por personas. Las cuales tiene tienen preferencias que transfieren a los modelos. Tanto sean conscientes como inconscientes. Muchas veces estas pueden pasar desapercibidos hasta que los modelos se ponen en producción.

En los procesos de toma de decisiones el término sesgo tiene generalmente

connotaciones negativas. No es deseable que un proceso automático lo tenga de ningún tipo. La palabra sesgo procede de sesgar, un verbo que hace referencia a torcer o atravesar algo hacia uno de sus lados. Por lo que una decisión sesgada, que se tuerce en algún sentido, no es deseable. Los modelos de aprendizaje automático ("machine learnig") no están exentos de este problema, ya que son desarrollados por personas. Así que es importante conocer qué es el sesgo en aprendizaje automático y cómo se puede minimizar su aparición.

Debido a todo esto se ha dirigido el estudio de muchos investigadores hacia la detección de sesgos en sistemas existentes por ejemplo existe un estudio que analiza y evalúa los sesgos existentes en los algoritmos y conjuntos de datos de análisis facial automatizados (tales como IARPA Janus Benchmark-A(IJB-A) y Adience [5]). En él se plantea la relevancia que puede tener en el análisis de los datos los subgrupos que se consideran, que pueden constituir la diferencia entre encontrar sesgos o no.

Varios sistemas de NLP usan los vectores (embeddings) de palabras, que son una forma de representarlas y capturar sus asociaciones y semántica. Estos vectores son construidos a partir de los contextos en que aparecen las palabras en los textos. Varios estudios se han enfocado en el reconocimiento de sesgos en estos vectores. Uno de ellos reconoce sesgos presentes en vectores generados por modelos como GloVe [19], y Word2Vec [16], entrenado con los artículos de Google News [4].

Los prejuicios y sesgos existentes en la sociedad han sido detectados también en cuerpos documentales como Wikipedia. Esto se refleja en las diferencias entre las formas en las que las mujeres y los hombres son descritos . Anuncios generados por modelos de aprendizaje automático [25], así como los motores de búsqueda en la Web [11] han sido detectados como fuentes de sesgos, por ejemplo, de género. Otros ejemplos de sistemas en los que se han detectado sesgos son los chatbots, los sistemas de reconocimiento facial y de voz , y algunos sistemas empleados en concursos de belleza.

COMPAS (Correctional Offender Management Profiling for Alternative Sanctions) es un software usado en las cortes de Estados Unidos para determinar la probabilidad de que una persona reincurra en un crimen. Se encontró que el software estaba sesgado hacia los afroamericanos [14]. Esto quiere decir que, dados dos individuos con el mismo perfil, bastaba con que se diferenciaran únicamente en que uno fuera afroamericano y el otro no para asociarle un mayor riesgo al primero. COMPAS es usado por los jueces para decidir si liberar a un prisionero o mantenerlo en la cárcel. Se ha mostrado que COMPAS no es más preciso que otros sistemas más simples que tienen el mismo objetivo, ni siquiera toma decisiones más imparciales que las que pueden tomar personas con ningún conocimiento judicial [8].

3. Conjunto de datos

3.1. Análisis

Para el desarrollo de este proyecto se seleccionó el dataset *UTKFace*. Este conjunto de datos incluye una amplia variedad de imágenes de más de 20,000 rostros de personas. Cada imagen está etiquetada con metadatos que describen la edad, el género y la raza de la persona en la fotografía. La raza se clasifica en las categorías: blanco, negro, asiático, indio y otra. *UTKFace* es ampliamente utilizado en investigaciones y aplicaciones relacionadas con el reconocimiento facial, análisis de edad y género.

UTKFace es un conjunto de datos especialmente adecuado para estudiar y abordar el sesgo en aplicaciones de reconocimiento facial y análisis de imágenes por su enfoque en la diversidad demográfica. Esto permitirá examinar y analizar los sesgos relacionados con estas variables demográficas en algoritmos de reconocimiento facial y otros sistemas de visión por computadora. Otro punto a favor es que contiene una amplia cantidad de imágenes, lo que brinda una cantidad significativa de datos para trabajar permitiendo realizar análisis estadísticos robustos y obtener conclusiones más confiables sobre los sesgos presentes en los algoritmos.

Figura 1: Muestra de las imágenes de UTKFace

3.2. Métricas de sesgo

Existen varias métricas y enfoques que se utilizan para analizar sesgos en un dataset, principalmente se analizarán *Proporción de clases* y la *Distribución demográfica*.

Al analizar gráficamente el conjunto de datos, se observa que tanto las edades, los géneros y las razas presentan una desproporción en la cantidad de imágenes. Se puede apreciar que existe un número considerable de imágenes en los rangos de edades (1, 4) y (25, 35), mientras que para edades mayores a 80, la presencia de datos es escasa o prácticamente nula.

En cuanto al género, se observa una marcada diferencia en la cantidad de datos entre hombres y mujeres, al igual que en el caso de la raza, donde los individuos de raza blanca presentan una considerable disparidad en comparación con los de raza negra, asiática, entre otras.

Estos hallazgos evidencian la existencia de desequilibrios significativos en el dataset en términos de edades, géneros y razas.

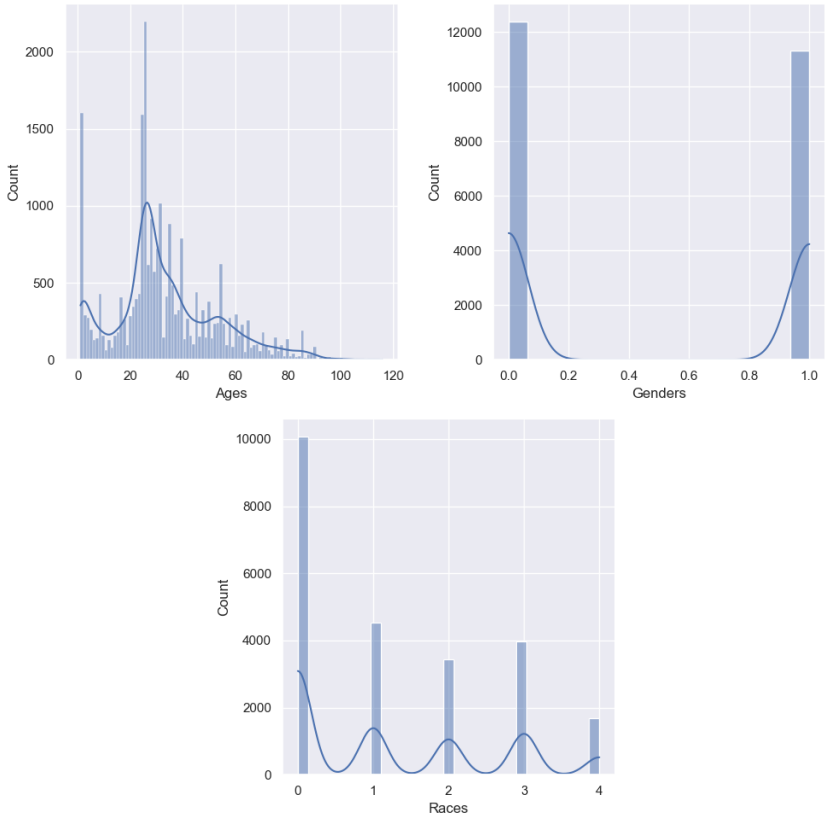

Figura 2: Distribución de las diferentes clases del dataset original

3.3. Mitigación de sesgo

Para el proceso de mitigación de sesgos, se implementaron dos enfoques que difieren en la prioridad otorgada a las clases a equilibrar.

En el primer enfoque, se prioriza en primer lugar el equilibrio de las edades y, con igual importancia, el género y la raza. Para lograr esto, se construye una estructura de datos que almacena la cantidad de elementos por cada edad, y se selecciona la edad con la menor y la mayor cantidad de datos. Con estos dos valores determinados, se procede a buscar las combinaciones de género y raza que presenten la menor y la mayor cantidad de datos. Al finalizar este proceso, se obtienen dos conjuntos de tríos que indican cuáles son las combinaciones de edad-género-raza con la menor cantidad de datos para aplicar "Data Augmentation"(aumento de datos) y cuáles son las de mayor cantidad de datos para eliminar elementos de manera aleatoria.

El aumento de datos se realizó aplicando *reflejo horizontal* [22], *rotación aleatoria*, *desenfoque gaussiano* [29] y *ruido gaussiano* [30] a las imágenes de los grupos con menor cantidad. Nótese que si el grupo no tiene al menos un elemento no se podrá aplicar este proceso ya que no habría imagen de partida.

Figura 3: A la izquierda la imagen original de UTKFace y a la derecha la resultante del algoritmo de generación.

En los resultados obtenidos mostrados en 4 se alcanza un balance significativo con respecto a las edades. Sin embargo, no se observa una mejora sustancial en la diferencia entre los dos géneros y entre las distintas razas.

En el segundo enfoque se cambió el orden de prioridades a la hora de balancear, priorizando primero la raza, luego el género y por último la edad. Para lograr esto se utilizó una estructura especial para acceder fácilmente a los datos según las prioridades establecidas. En primer lugar, se emplea una matriz llamada *races_genders*, donde en la fila i (correspondiente a la i-ésima raza) y la columna j (correspondiente al j-ésimo género), se almacena la cantidad de imágenes que pertenecen a las respectivas combinaciones de raza y género. A continuación, se utiliza un diccionario en el que las tuplas (género, raza) se emplean como llaves para obtener una lista de tamaño igual a la cantidad de edades presente en el dataset. Esta lista contiene la cantidad de imágenes que poseen una edad igual a la posición i en la lista. Por último, se emplea otro diccionario que relaciona las triplas (raza, género, edad) con la lista de imágenes que cumplen con las tres características mencionadas.

Una vez que las diferentes estructuras de datos contienen la información de todas las imágenes el procedimiento es el siguiente:

- De la matriz *races_genders* se obtienen cuáles son la raza y género menor/mayor cantidad de datos.
- Luego dado el par obtenido se busca la edad con la mínima/maxima cantidad de datos.
- Una vez que tenemos la tripla (raza,género,edad) se selecciona una imagen aleatoria de la lista y se genera/elimina a partir de ella.

Luego de graficar el conjunto de datos 5 destaca que se logró un buen balance de raza y género, no así con la edad ya que delegada como última prioridad.

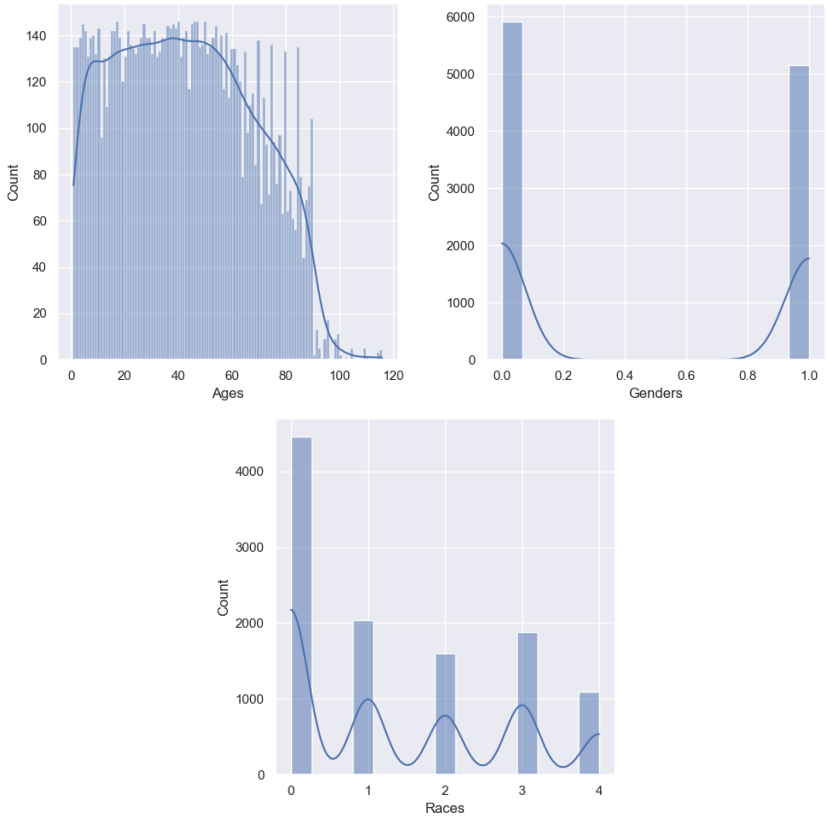

Figura 4: Distribución resultante de aplicar el primer enfoque de mitigación de sesgos a UTKFace

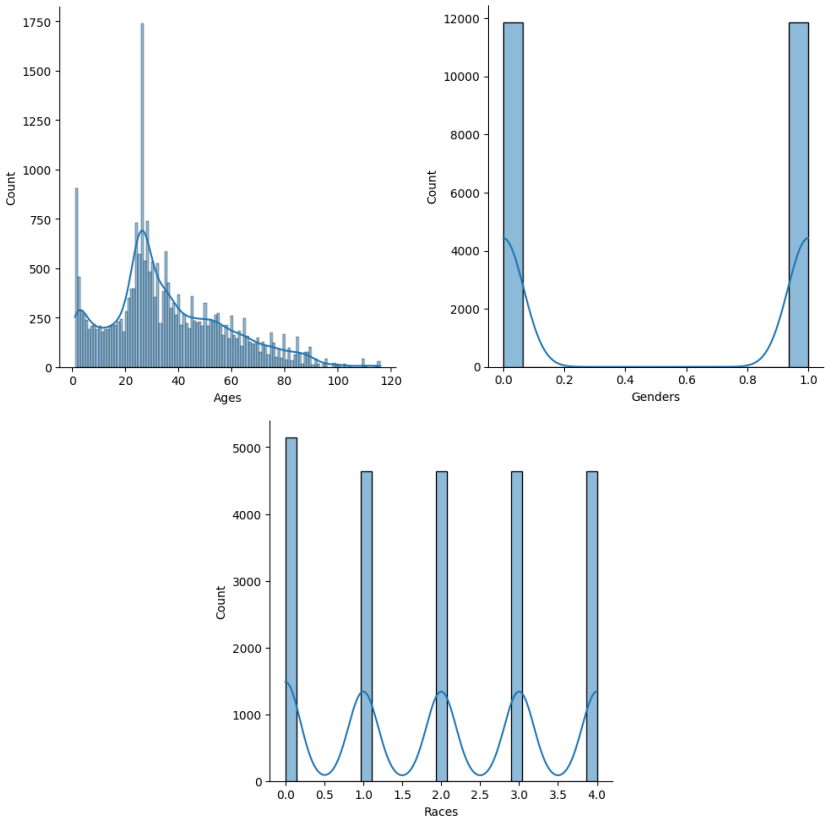

Figura 5: Distribución resultante de aplicar el segundo enfoque de mitigación de sesgos a UTKFace

4. Métricas de equidad en algoritmos

Medir la equidad en los modelos de aprendizaje automático (ML) es un área de investigación compleja y en curso, y existen muchos enfoques diferentes que se pueden tomar según el contexto específico y los objetivos del modelo.

La mayoría de las medidas de equidad dependen de diferentes métricas, de modo que se comienza por definirlas. Cuando se trabaja con un clasificador binario, tanto la clase predicha por el algoritmo como la real pueden tomar dos valores: positivo y negativo, por tanto se explican las posibles relaciones entre el resultado predicho y el real:

- **Verdadero positivo (TP):** Cuando el resultado predicho y el real pertenecen a la clase positiva.

- **Verdadero negativo (TN):** Cuando el resultado predicho y el real pertenecen a la clase negativa.

- **Falso positivo (FP):** Cuando el resultado predicho es positivo pero el real pertenece a la clase negativa.

- **Falso negativo (FN):** Cuando el resultado predicho es negativo pero el real pertenece a la clase positiva.

Estas relaciones pueden ser representadas fácilmente con una matriz de confusión, una tabla que describe la precisión de un modelo de clasificación. En esta matriz, las columnas y las filas representan instancias de las clases predichas y reales, respectivamente.

Utilizando estas relaciones, podemos definir múltiples métricas que podemos usar después para medir la equidad de un algoritmo:

- **Valor predicho positivo (PPV):** la fracción de casos positivos que han sido predichos correctamente de entre todas las predicciones positivas. Con frecuencia, se denomina como precisión, y representa la probabilidad de que una predicción positiva sea correcta. Viene dada por la siguiente fórmula:
  $$PPV = P(actual = +|prediction = +) = \frac{TP}{TP + FP}$$

- **Tasa de descubrimiento de falsos (FDR):** la fracción de predicciones positivas que eran en realidad negativas de entre todas las predicciones positivas. Representa la probabilidad de que una predicción positiva sea errónea, y viene dada por la siguiente fórmula:
  $$FDR = P(actual = -|prediction = +) = \frac{FP}{TP + FP}$$

- **Valor predicho negativo (NPV):** la fracción de casos negativos que han sido predichos correctamente de entre todas las predicciones negativas. Representa la probabilidad de que una predicción negativa sea correcta, y viene dada por la siguiente fórmula:
  $$NPV = P(actual = -|prediction = -) = \frac{TN}{TN + FN}$$

- **Tasa de omisión de falsos (FOR):** la fracción de predicciones negativas que eran en realidad positivas de entre todas las predicciones negativas.

Representa la probabilidad de que una predicción negativa sea errónea, y viene dada por la siguiente fórmula:

$$FOR = P(actual = +|prediction = -) = \frac{FN}{TN + FN}$$

- **Tasa de verdaderos positivos (TPR):** la fracción de casos positivos que han sido predichos correctamente de entre todos los casos positivos. Con frecuencia, se denomina como exhaustividad, y representa la probabilidad de que los sujetos positivos sean clasificados correctamente como tales. Viene dada por la fórmula:

$$TPR = P(prediction = +|actual = +) = \frac{TP}{TP + FN}$$

- **Tasa de falsos negativos (FNR):** la fracción de casos positivos que han sido predichos de forma errónea como negativos de entre todos los casos positivos. Representa la probabilidad de que los sujetos positivos sean clasificados erróneamente como negativos, y viene dada por la fórmula:

$$FNR = P(prediction = -|actual = +) = \frac{FN}{TP + FN}$$

- **Tasa de verdaderos negativos (TNR):** la fracción de casos negativos que han sido predichos correctamente de entre todos los casos negativos. Representa la probabilidad de que los sujetos negativos sean clasificados correctamente como tales, y viene dada por la fórmula:

$$TNR = P(prediction = -|actual = -) = \frac{TN}{TN + FP}$$

- **Tasa de falsos positivos (FPR):** la fracción de casos negativos que han sido predichos de forma errónea como positivos de entre todos los casos negativos. Representa la probabilidad de que los sujetos negativos sean clasificados erróneamente como positivos, y viene dada por la fórmula:

$$FPR = P(prediction = +|actual = -) = \frac{FP}{TN + FP}$$

Hay muchas definiciones de equidad que se han propuesto en la literatura. Sin embargo, la mayoría de ellos se basan en los siguientes [15]:

1. Igualdad de probabilidades (Equalized Odds)
2. Igualdad de oportunidades (Equal Opportunity)
3. Paridad demográfica (Demographic Parity)
4. Igualdad de trato (Treatment Equality)
5. Equidad individual

6. Equidad contrafactual (Counterfactual Fairness)

Estas definiciones corresponden en el caso de las cuatro primeras a una categoría más amplia llamada .eqquidad de grupoz las últimas dos corresponden a la categoría .eqquidad individual". [15]

En el caso del presente trabajo, se abordarán las definiciones de equidad de grupo, ya que se estará analizando la equidad de modelos de clasificación para género y raza.

4.0.1. Equidad de grupo

Paridad demográfica:

Un algoritmo predictor Y cumple con Demographic Parity con respecto a un atributo A con valores en el conjunto 0,1 si se cumple: $P(Y|A = 0) = P(Y|A = 1)$. Esto implica que el valor del atributo protegido no influye en el resultado del algoritmo Y.

La paridad demográfica establece que la proporción de cada segmento de una clase protegida (por ejemplo, el género) debe recibir el resultado positivo a tasas iguales. [7] Un resultado positivo es la decisión preferida, como .obtener un préstamo", "que se le muestre el anuncio.o en el caso del modelo que se presenta "predecir correctamente la clase a la que pertenece". Esta diferencia debería ser idealmente cero, pero este no suele ser el caso y se acepta mientras esté por debajo del 80 %. [34] Esta métrica suele usarse cuando se sabe que los sesgos históricos pueden haber afectado la calidad de nuestros datos o cuando se quiere que los grupos minoritarios obtengan mejores resultados en el modelo. [7]

Esta métrica aparece en algunas bibliografías bajo el nombre de impacto dispar o paridad estadística. En el presente documento se hará referencia a esta medida como impacto dispar.

Igualdad de oportunidades:

La Igualdad de Oportunidades establece que cada grupo debe obtener el resultado positivo en proporciones iguales suponiendo que las personas de este grupo califiquen para ello. La Igualdad de Oportunidades requiere que el resultado positivo sea independiente de la clase A protegida, condicionado a que Y sea un resultado positivo real. $P(\tilde{Y}|A = 0, Y = 1) = P(\tilde{Y}|A = 1, Y = 1)$

Según la matriz de confusión, lo que se quiere es que la tasa de verdaderos positivos (TPR) sea la misma para cada segmento de la clase protegida.

$$TPR_{(A=0)} = TPR_{(A=1)}$$

En la práctica, es posible que no se exija que la diferencia en las tasas positivas sea igual a cero, pero si que se intente minimizar la brecha.

Esta métrica suele usarse cuando hay un fuerte énfasis en predecir el resultado positivo correctamente y la introducción de falsos positivos no es costosa. [7]

Igualdad de probabilidades:

Este concepto establece que el modelo debe: identificar correctamente el resultado positivo a tasas iguales en todos los grupos (igual que en Igualdad de Oportunidades), pero también clasificar erróneamente el resultado positivo a tasas iguales en todos los grupos (creando la misma proporción de falsos positivos en todos los grupos)

Según la matriz de confusión, lo que se quiere es que la tasa de verdaderos positivos (TPR) y la tasa de falsos positivos (FPR) sean las mismas para cada segmento de la clase protegida.

$$TPR_{(A=0)} = TPR_{(A=1)}$$

$$FPR_{(A=0)} = FPR_{(A=1)}$$

Al igual que en el ejemplo anterior no siempre se requiere que la diferencia sea 0, pero sí que sea lo menor posible.

Dado que esta es la más restrictiva de las definiciones, tratar de lograr TPR y FPR iguales para cada grupo puede conducir a una caída en la precisión. Esto se debe a que el rendimiento del modelo podría verse comprometido al no poder optimizar la precisión en el grupo mayoritario. [7]

Esta métrica suele usarse cuando hay un fuerte énfasis en predecir el resultado positivo correctamente y hay preocupación por minimizar los falsos positivos costosos.

4.0.2. Métricas de equidad en modelos en cuestión.

Los modelos que se presentan son clasificadores de género y/o raza en imágenes. Lo que se quiere evaluar es la equidad en la clasificación por cada grupo, por lo que se utilizan las métricas de equidad de grupos. Además al tratarse de un clasificador, las métricas que se considerarán serán impacto dispar e igualdad de oportunidades.

5. Técnicas de mitigación de sesgos en algoritmos

La mitigación de sesgo en algoritmos de aprendizaje automático (ML) suele dividirse en tres clases [15]:

1. **Pre-procesamiento:** Las técnicas de preprocesamiento intentan transformar los datos para eliminar la discriminación subyacente. Si se permite que el algoritmo modifique los datos de entrenamiento, entonces se puede usar el preprocesamiento.

2. **Durante el procesamiento:** Las técnicas de procesamiento intentan modificar y cambiar los algoritmos de aprendizaje para eliminar la discriminación durante el proceso de entrenamiento del modelo. Si se permite cambiar el procedimiento de aprendizaje para un modelo de aprendizaje automático, entonces se puede usar el procesamiento interno durante el entrenamiento de un modelo, ya sea incorporando cambios en la función objetivo o imponiendo una restricción

3. **Pos-procesamiento:** El posprocesamiento se realiza después del entrenamiento accediendo a un conjunto de espera que no estuvo involucrado durante el entrenamiento del modelo. Si el algoritmo solo puede tratar el modelo aprendido como una caja negra sin ninguna capacidad para modificar los datos de entrenamiento o el algoritmo de aprendizaje, entonces solo se puede usar el posprocesamiento en el que las etiquetas asignadas por el modelo de caja negra inicialmente se reasignan según en una función durante la fase de posprocesamiento

6. Análisis de sesgo en modelos sobre UTKFace

6.1. Predicción de género

Se selecionó un modelo existente en Kaggle [27] para predecir edad y género sobre el dataset UTKFace. La edad es una variable compleja de predecir en modelos de ML, más aún cuando no se tiene información completa sobre los factores que influyen en la edad; por esta razón se modificó el modelo para solo predecir género.

6.1.1. Descripción

El modelo define una red neuronal convolucional (CNN) para la clasificación binaria de imágenes utilizando la biblioteca Keras [12]. Esta red se compone de varias capas que se encargan de extraer características de las imágenes de entrada y, finalmente, clasificarlas en dos categorías posibles.

La primera capa es una capa Conv2D, que se encarga de extraer características de las imágenes de entrada mediante la aplicación de filtros convolucionales. En este caso, se utilizan 32 filtros de tamaño 3x3 y una función de activación ReLU.

La segunda capa es una capa MaxPooling2D, que reduce la dimensión espacial de la salida de la capa anterior. En este caso, se utiliza una reducción de tamaño de 2x2. Esta reducción de tamaño ayuda a reducir el número de parámetros de la red y a disminuir el costo computacional de la misma.

A continuación, se repite el mismo patrón de capas: una capa Conv2D seguida de una capa MaxPooling2D. En la segunda capa Conv2D se utilizan 64 filtros y en la tercera se utilizan 128 filtros, ambas con una función de activación ReLU.

Después de las capas convolucionales, se utiliza una capa de aplastamiento (Flatten) para convertir la salida de la última capa convolucional en un vector unidimensional. Este vector se utiliza como entrada para dos capas totalmente conectadas (Dense). La primera capa Dense tiene 64 neuronas y una función de activación ReLU. La segunda capa Dense tiene una sola neurona y una función de activación sigmoide, lo que significa que el modelo producirá una salida en el rango de 0 a 1, que se puede interpretar como la probabilidad de que la imagen pertenezca a la clase positiva.

Para evitar el sobreajuste, se utiliza una capa de abandono (Dropout) con una probabilidad de abandono del 50 % después de la primera capa Dense.

La estructura de la red neuronal se puede observar en la Figura 6.

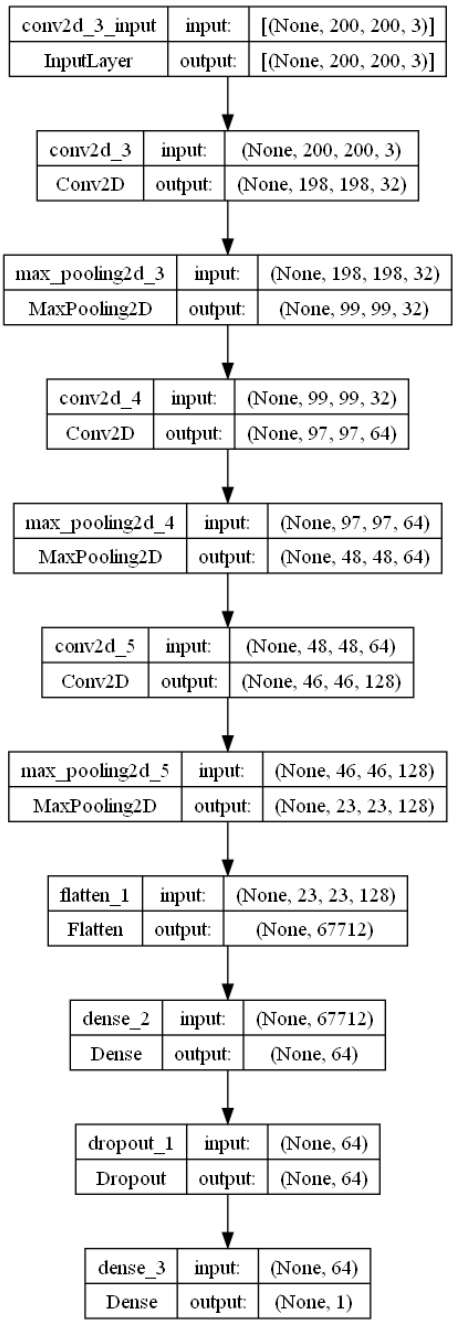

Figura 6: Estructura de la red neuronal.

6.1.2. Selección y procesamiento de datos

Por limitaciones de hardware se seleccionaron 8000 imágenes del dataset UTKFace, se utilizaron criterios específicos para asegurar una distribución representativa de las imágenes por edad y género. En concreto, se eligieron imágenes para cada rango de edad de 10 años cuidando su proporción. Se verificó además el total de hombres y mujeres quedando 55 % a 45 % respectivamente.

Se dividen los datos en subconjuntos de entrenamientos y pruebas en proporción 80:20 respectivamente y se realizan aumentos de datos y normalización en estos conjuntos. El aumento de datos implica la aplicación de transformaciones aleatorias a las imágenes de entrenamiento, como desplazamientos horizontales y verticales o volteos horizontales. La normalización escala los valores de los píxeles de las imágenes al rango [0,1] lo que facilita el entrenamiento de la red y mejora la convergencia. Sin la normalización, los valores de los píxeles pueden ser muy diferentes entre las imágenes, lo que puede hacer que la red tenga dificultades para aprender y converger.

6.1.3. Entrenamiento

Para entrenar la red se realizaron 50 *epochs*. Se compila el modelo utilizando la función de pérdida de entropía cruzada binaria (binary cross-entropy) y el optimizador Adam con una tasa de aprendizaje de 0,0001.

La función de pérdida mide la diferencia entre la etiqueta predicha y la verdadera, mientras que el optimizador Adam es un algoritmo de optimización de tasa de aprendizaje adaptativa que se utiliza comúnmente en las redes neuronales.

6.1.4. Métricas

Evaluación

Se evaluó el rendimiento del modelo utilizando la precisión macro (macro accuracy) y el F1 macro (macro F1-score). Para el conjunto de prueba, se obtuvo una precisión macro de 82,6 % y un macro F1 de 81,9 %.

6.1.5. Justicia

En cuanto a la evaluación de la equidad del modelo, se utilizaron las métricas de Probabilidades igualadas e Impacto Dispar. Para el conjunto de prueba, se obtuvo un valor de 0.103 para la tasa de verdaderos positivos y la tasa de falsos positivos, lo que indica que no se cumplen las condiciones de igualdad de oportunidades entre hombres y mujeres, siendo el grupo privilegiado los hombres.

Para el conjunto de prueba, se obtuvo un valor de 0.681, lo que indica la presencia de un impacto desproporcionado en la clasificación entre hombres y mujeres.

En resumen, se puede observar que el modelo presenta problemas de equidad en la clasificación de género, lo que sugiere una necesidad de ajustar el modelo.

6.1.6. Datos curados

Aunque se utilizó un nuevo dataset curado para el entrenamiento del modelo, no se observó un cambio significativo en el rendimiento del modelo debido a que se limitó el número de imágenes. Se alcanzó una precisión macro de 82,4 % y un macro F1 de 81,8 %, lo que indica que el modelo es robusto a los cambios en el dataset utilizado para su entrenamiento.

En cuanto a las métricas de sesgo, se obtuvo un valor de 0,105 para la métrica de Probabilidades Igualadas y un valor de 0.719 para la métrica de Impacto Dispar, que indican niveles similares a los del conjunto de datos original.

6.2. Clasificador demográfico sobre UTK-face

Se seleccionó el modelo de clasificación demográfica [23] para predecir edad, género y raza sobre el dataset UTKFace.

6.2.1. Descripción

Este modelo está dirigido a la predicción de la edad, la raza y el sexo de una persona a partir de la imagen de su rostro. La figura 7 muestra la arquitectura de la red neuronal empleada.

El modelo consta de tres ramas, una para cada atributo, y una red de extracción de características compartida que extrae características de la imagen de entrada.

La red de extracción de características es una red neuronal convolucional profunda con varias capas convolucionales con funciones de activación de $relu$, seguidas de capas de agrupación máximas (max pooling) con tamaño de grupo 2x2. El número de filtros por capa aumenta de 64 a 512, que es un patrón común en las redes neuronales convolucionales.

La salida de esta red se alimenta en tres ramas diferentes, cada una de las cuales consta de capas completamente conectadas que predicen uno de los atributos.

La rama de edad consta de tres capas completamente conectadas con funciones de activación de $relu$, seguidas de una sola neurona de salida con una función de activación lineal. Esta rama tiene como objetivo predecir un valor numérico que es la edad, de ahí la arquitectra y funciones utilizadas.

La rama de género consta de tres capas completamente conectadas con funciones de activación de $relu$, seguidas de una sola neurona de salida con una función de activación de $softmax$. La función $softmax$ normaliza los valores de salida convirtiéndolos en una distribución de probabilidades, lo que puede interpretarse como la probabilidad de que la imagen de entrada pertenezca a cada género.

La rama de raza es similar a la rama de género, pero tiene una cantidad variable de neuronas de salida según la cantidad de razas en el conjunto de datos. La capa de salida utiliza la función de activación $softmax$ para predecir la probabilidad de que la imagen de entrada pertenezca a cada raza.

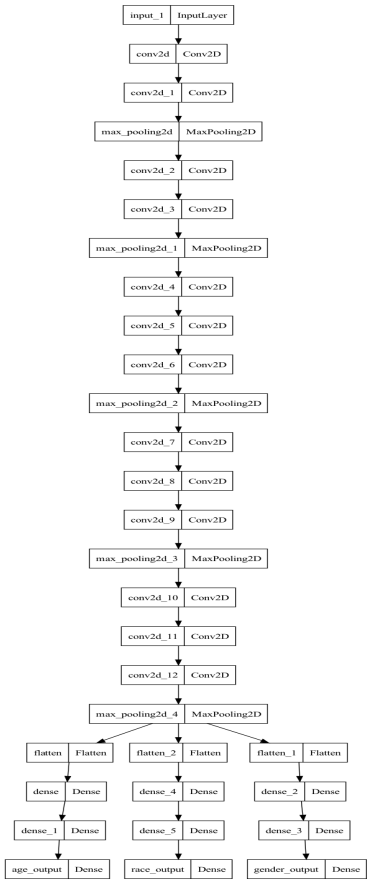

Figura 7: Arquitectura de la red neuronal

Se utiliza el optimizador de Adam con una tasa de aprendizaje de 0,0001 para el entrenamiento. Las funciones de pérdida utilizadas para entrenar el modelo son el error cuadrático medio (MSE) para la rama de edad, la entropía cruzada categórica para la rama de raza y la entropía cruzada binaria para la rama de género. Estas son funciones de pérdida apropiadas para los tipos de salida correspondientes.

6.2.2. Selección y procesamiento de datos

6.2.3. Entrenamiento

6.2.4. Métricas

6.2.5. Justicia

6.3. Modelo de Mayank Tripathi [26]

El autor propone un modelo convolucional (CNN Convolutional Neural Network) entrenado con *UTKFace* que es capaz de predecir el género de la persona en una imagen. Utilizando la biblioteca *Keras* crea una red convolucional con la siguiente estructura:

1. Primero se define la capa de entrada de la red con un tamaño de imagen de 100x100 píxeles y 1 canal (escala de grises). Esto crea un objeto *input* que actuará como entrada para la red.

```
input = Input(shape = (100, 100, 1))
```

2. Se crea la primera capa convolucional con 32 filtros de tamaño 3x3. Se aplica *padding* para mantener el tamaño de la imagen de entrada y se utiliza una regularización L2 con un factor de penalización de 0.001. Esta capa se conecta a la capa de entrada *input* y produce un tensor *conv1* como salida.

```
conv1 = Conv2D(32, (3,3),
               padding = 'same', strides = (1, 1),
               kernel_regularizer = l2(0.001))(input)
```

3. Se agrega una capa de *dropout* con una tasa de 0.1 a "*conv*1". El *dropout* se utiliza para evitar el sobreajuste al apagar aleatoriamente algunas neuronas durante el entrenamiento.

```
conv1 = Dropout(0.1)(conv1)
```

4. Se adiciona una función de activación *ReLU* a "*conv*1". La activación ReLU aplica la función $max(0, x)$ a cada elemento del tensor, introduciendo no linealidad a la red.

```
conv1 = Activation('relu')(conv1)
```

5. Agrega una capa de pooling (agrupamiento) con una ventana de $2x2$ al tensor "*conv*1". La capa de pooling reduce la dimensionalidad de los mapas de características y extrae características relevantes.

```
pool1 = MaxPooling2D(pool_size = (2,2))(conv1)
```

6. Los pasos 2 a 5 se repiten para construir más capas convolucionales (*conv*2, *conv*3, *conv*4) y capas de pooling (*pool*2, *pool*3, *pool*4) con diferentes tamaños de filtros y cantidades de filtros.

```
conv2 = Conv2D(64, (3,3),
                padding = 'same', strides = (1, 1),
                kernel_regularizer = l2(0.001))(pool1)
conv2 = Dropout(0.1)(conv2)
conv2 = Activation('relu')(conv2)
pool2 = MaxPooling2D(pool_size = (2,2))(conv2)

conv3 = Conv2D(128, (3,3),
                padding = 'same', strides = (1, 1),
                kernel_regularizer = l2(0.001))(pool2)
conv3 = Dropout(0.1)(conv3)
conv3 = Activation('relu')(conv3)
pool3 = MaxPooling2D(pool_size = (2,2))(conv3)

conv4 = Conv2D(256, (3,3),
                padding = 'same', strides = (1, 1),
                kernel_regularizer = l2(0.001))(pool3)
conv4 = Dropout(0.1)(conv4)
conv4 = Activation('relu')(conv4)
pool4 = MaxPooling2D(pool_size = (2,2))(conv4)
```

7. Luego se aplana el tensor *pool*4 en un vector unidimensional para alimentarlo a las capas completamente conectadas. Esto se hace utilizando la capa $Flatten()$.

```
flatten = Flatten()(pool4)
```

8. Se agrega una capa completamente conectada con 128 unidades y una función de activación *ReLU*. Esta capa se conecta al tensor aplanado $flatten$ y produce un tensor como salida.

```
dense1 = Dense(128, activation = 'relu')(flatten)
```

9. Después se agrega una capa de *dropout* con una tasa de 0.2 a *dense*1.

```
drop1 = Dropout(0.2)(dense1)
```

10. Y por último se agrega una capa de salida completamente conectada con 2 unidades y una función de activación *sigmoide*. Esta capa produce la salida final de la red.

```
output = Dense(2, activation = 'sigmoid')(drop1)
```

Resultados

Luego de entrenado el modelo de predicción de género y evaluado con el conjunto de datos de prueba se obtiene la matriz de confusión9.

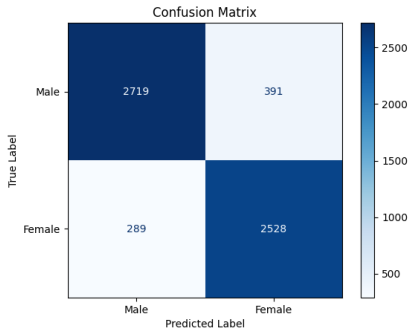

Figura 8: Matriz de confusion resultante de evaluar el modelo entrenado con UTKFace

Luego se utilizan los datos contenidos en la matriz de confusión para analizar la precisión de las predicciones del modelo

$$P = \frac{TP}{TP + FP}$$

de la cual se obtuvo un valor de 0,9.

Para analizar si existe sesgo en género dos de las métricas mas utilizadas son $TruePositiveRate(TPR)$ y $FalsePositiveRate$ donde si la diferencia entre el TPR femenino y masculino no excedea 0,5 se puede decir que las posibilidades de los géneros son parejas.

$$TPR = \frac{TP}{TP + FN}$$

$$FNR = \frac{FP}{FP + TN}$$

Los resultados obtenidos son de 0,023 por lo que se puede concluir que se trata de un modelo con iguales posibilidades para ambos géneros.

6.4. Modelo de Omky Aghav [1]

Omky Aghav presenta dos modelos de redes neuronales convolucionales como solución a la problemática de la predicción de edad y género. Las primeras cpas de ambos modelos coinciden, luego se crean capas con diferentes especializaciones para la solucion de cada una de las tareas.

```
inputs = Input(shape=(sample.shape[0], sample.shape[1], 1))
conv1 = Conv2D(32, kernel_size=(3,3), activation='relu')(inputs)
conv2 = Conv2D(64, kernel_size=(3,3), activation='relu')(conv1)
pool1 = MaxPooling2D(pool_size=(2,2))(conv2)
conv3 = Conv2D(128, kernel_size=(3,3), activation='relu')(pool1)
pool2 = MaxPooling2D(pool_size=(2,2))(conv3)
```

```
x = Dropout(0.25)(pool2)
flat = Flatten()(x)

dropout = Dropout(0.5)
age_model = Dense(128, activation='relu')(flat)
age_model = dropout(age_model)
age_model = Dense(64, activation='relu')(age_model)
age_model = dropout(age_model)
age_model = Dense(32, activation='relu')(age_model)
age_model = dropout(age_model)
age_model = Dense(1, activation='relu')(age_model)

dropout = Dropout(0.5)
gender_model = Dense(128, activation='relu')(flat)
gender_model = dropout(gender_model)
gender_model = Dense(64, activation='relu')(gender_model)
gender_model = dropout(gender_model)
gender_model = Dense(32, activation='relu')(gender_model)
gender_model = dropout(gender_model)
gender_model = Dense(16, activation='relu')(gender_model)
gender_model = dropout(gender_model)
gender_model = Dense(8, activation='relu')(gender_model)
gender_model = dropout(gender_model)
gender_model = Dense(1,
                     activation='sigmoid')(gender_model)
```

La composición de los modelos está dada por varias capas que se describen a continuación:

- Se define el *input* del modelo utilizando la función Input de Keras. El tamaño del input se basa en las dimensiones de una muestra individual, con una profundidad de 1 (escala de grises).
- Se agregan capas convolucionales ($Conv2D$) con diferentes filtros y funciones de activación ($relu$) para extraer características de la imagen.
- Se aplican capas de *pooling*($MaxPooling2D$) para reducir el tamaño espacial de las características extraídas.
- Se aplica una capa de *Dropout* para regularizar el modelo y evitar el sobreajuste.
- Se utiliza la capa *Flatten* para convertir los mapas de características 2D en un vector 1D.
- Se definen las capas de salida para la tarea de predicción de edad. Se utilizan capas densas ($Dense$) con funciones de activación $relu$ y se aplica *Dropout* en cada capa para regularizar el modelo.
- La capa de salida final tiene una sola neurona y utiliza la función de activación $relu$.

- Se definen las capas de salida para la tarea de predicción de género. Al igual que en la predicción de edad, se utilizan capas *densas* con funciones de activación *relu* y *Dropout* para regularización. La capa de salida final tiene una sola neurona y utiliza la función de activación *sigmoid* para la clasificación binaria.

Resultados Para analizar el sesgo en el modelo de predicción de género se realizan los mismos pasos antes planteados.

Dada la matriz de confusión 9

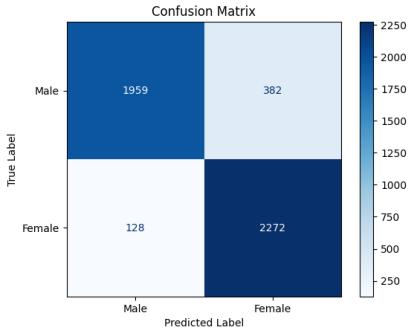

Figura 9: Matriz de confusión resultante de evaluar el modelo entrenado con UTKFace

Se repiten los cálculos de las métricas escogidas y para este modelo el resultado fue de 0,11, valor que es mayor a 0,05 por lo que hay evidencia de una desigualdad en las posibilidades de cada género.

Por este motivo se repite todo el proceso pero esta vez con el dataset resultante del proceso de mitigación de sesgo obteniendo como resultado la matriz de confusión 10

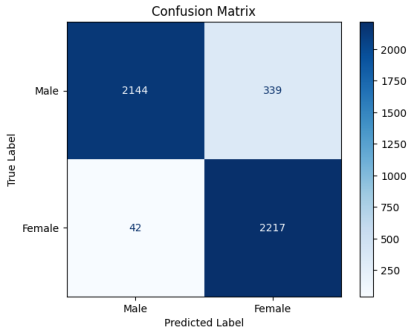

Figura 10: Matriz de confusión resultante de UTKFace debiased

Luego de aplicar las métricas antes mencionadas el resultado alcanzado fue de 0,10. Se puede concluir que las mejoras que plantea el nuevo dataset no

son de gran relevancia para el modelo en cuestión, el cual mantuvo n valor de desigualdad practicamente inmutable.

7. Contrastive Language Image Pretraining (CLIP)

7.1. Breve descripción del modelo

CLIP es un modelo de Aprendizaje Profundo (Deep Learning) desarrollado por OpenAI en 2021, que utiliza una arquitectura de red neuronal convolucional multi-modal para la clasificación de imágenes y el procesamiento del lenguaje natural. Este no está entrenado en una tarea específica como la clasificación de imágenes o la generación de texto, CLIP se entrena para entender la relación entre las imágenes y el lenguaje natural, en un conjunto de datos de imágenes y texto emparejados.

CLIP utiliza una técnica llamada aprendizaje por contrastación, que le permite aprender a relacionar las imágenes y el texto al comparar pares de datos de entrenamiento. En lugar de intentar predecir una etiqueta específica para una imagen determinada, el modelo aprende a clasificar una imagen en relación a un texto determinado y viceversa.

Una de las características más interesantes de CLIP es que, a diferencia de otros modelos de clasificación de imágenes, no necesita etiquetas específicas para cada imagen. En su lugar, el modelo es capaz de clasificar las imágenes en función de los conceptos que se describen en el texto que se le proporciona.

En cuanto a su arquitectura, CLIP utiliza una red neuronal convolucional basada en la arquitectura ViT (Vision Transformer) para procesar las imágenes, y una red neuronal basada en GPT (Generative Pre-trained Transformer) para procesar el texto. La salida de ambas redes se combina en una capa de clasificación final que produce una puntuación de similitud entre la imagen y el texto. [20]

CLIP ha demostrado ser altamente efectivo en una amplia variedad de tareas de clasificación de imágenes, incluyendo la identificación de objetos, detección de emociones, clasificación de eventos, y ha superado a otros modelos de clasificación de imágenes en varios benchmarks. [17]

Las imágenes y textos son representados como vectores en un espacio de características compartido, como se muestra en la Figura 11, lo que permite medir la similitud entre ellos. Este es un espacio de n dimensiones, donde la posición relativa de los vectores refleja la relación semántica entre las imágenes y los textos: si dos imágenes o textos son similares en términos de contenido, sus vectores correspondientes estarán cerca en el espacio compartido.

7.2. Sesgo en CLIP

El artículo .ªudit finds gender and age bias in OpenAI's CLIP model"(Auditoría encuentra sesgo de género y edad en el modelo CLIP de OpenAI) [28] reporta sobre un informe de auditoría encargado por OpenAI, una organización de in-

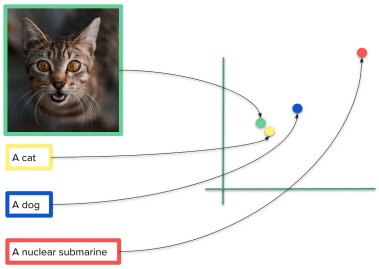

Figura 11: Representación de vectores en el espacio vectorial.

vestigación en inteligencia artificial. El informe fue realizado por la empresa de consultoría en ética de IA, Fiddler AI [10].

El informe de Fiddler AI encontró que el modelo tenía una tendencia a asociar ciertas características con ciertos géneros y edades, lo que podría llevar a una discriminación involuntaria en la selección de imágenes.

Específicamente, el informe encontró que el modelo CLIP tendía a asociar características como maquillaje y joyería con mujeres y características como barbas y bigotes con hombres. Además, el modelo también tendía a asociar características como canas y arrugas con personas mayores.

El informe de Fiddler AI destaca la importancia de abordar estos problemas de sesgo en los modelos de IA, especialmente en aplicaciones donde las decisiones basadas en la clasificación de imágenes pueden tener un gran impacto en la vida de las personas. El informe también destaca la necesidad de una mayor transparencia y explicabilidad en el desarrollo de modelos de IA para abordar estos problemas de sesgo. En respuesta al informe, OpenAI reconoció la importancia de abordar los sesgos en sus modelos y anunció planes para mejorar la transparencia y la responsabilidad en su trabajo futuro. La organización declaró que .^{es}te informe es un recordatorio importante de que todavía queda mucho por hacer para garantizar que la IA se desarrolle y se utilice de manera justa y responsable".

7.3. Resultados sobre UTKFace

7.3.1. Precisión

El modelo sobre el dataset presenta los siguientes resultados en cuanto a precisión:

Medida de precisión	Género	Raza
Macro Accuracy	0,9581212526266386	0,6803203012378448
Macro F1	0,9579553326742262	0,6768578431973549

7.3.2. Justicia

A continuación se presentan algunos de los resultados de CLIP sobre UTK-Face respecto a sesgos de equidad:

Raza

Raza					
Tasa de selección (Selection Rate)	Blanco	Negro	Asiático	Indio	Otra
	0,8734491315136477	0,7637969094922737	0,9432314410480349	0,7811320754716982	0,8259587020648967

La tasa de selección es utilizada para calcular la métrica de impacto dispar (Disparate Impact), a partir de calcular la relación entre cada par de valores (denominado en inglés Disparate Impact Ratio), en este caso, de tipos de raza. A continuación se presentan algunas de las relaciones más interesantes en raza:

Negro/Blanco	Indio/Blanco	Blanco/Asiático
0,8744606662653021	0,8943074614065181	0,926017829243636

En cuanto a Equalized Odds, en el caso de los TPR se ha hallado desigualdad de oportunidad para el caso entre:

1. Blanco y negro
2. Negro y asiático
3. Asiático e indio

En el caso de los FPR, no se ha satisfacido la hipótesis de probabilidades igualadas para ningún par de razas.

Género

Medida de justicia	Género			
Tasa de selección (para Disparate Impact)	Masculino	Femenino		
	0,9515738498789347	0,9624558303886925		
Probabilidades igualadas (Equalized odds)	TPR		FPR	
	Masculino	Femenino	Masculino	Femenino
	0,9515738498789347	0,9624558303886925	0,4146341463414634	0,5853658536585366
	0,108819805097578		0,1707317073170732	

No hay impacto dispar presente entre el género masculino y el femenino, puesto que el valor de la relación de impacto dispar, como se ha expuesto previamente que se calcula, es mayor al 80 por ciento.

No se satisfacen las hipótesis de las probabilidades igualadas para el umbral establecido en los casos de los TPR (True Positive Rate) y FPR (False Positive Rate) que es menor al 10 por ciento. En el caso de TPR, como muestra la tabla, es de alrededor de un 11 por ciento, mientras que para los FPR es de un 17 por ciento.

8. Mitigación de sesgo

Ajuste fino

En el aprendizaje automático, el ajuste fino es un enfoque para el aprendizaje de transferencia (transfer learning) en el que los pesos de un modelo preentrenado se entrenan en nuevos datos [31].

8.1. Ajuste fino con Jina

8.1.1. ¿Qué es Finetuner de Jina?

Jina [3] es un marco de trabajo de *MLOps* (Operaciones de Aprendizaje Automático) para construir aplicaciones de inteligencia artificial multimodales basadas en microservicios, escritas en Python.

Finetuner [2] es una herramienta de Jina que hace que el proceso de fine-tune de modelos de búsqueda basados en redes neuronales sea más fácil y rápido. Con Finetuner el entrenamiento se lleva a cabo en la infraestructura de GPU de Jina AI Cloud, lo que permite a los usuarios gestionar ejecuciones, experimentos y artefactos sin preocuparse por la disponibilidad de recursos, la integración compleja o los costos de infraestructura.

8.1.2. Selección y procesamiento de datos

Gracias al poder de Finetuner se pudo usar todo el dataset para reentrenar CLIP. El conjunto de datos se dividió en varios subconjuntos utilizando la técnica K-Fold estratificado (Stratified K-Fold) para asegurarse de que cada uno de los subconjuntos creados contenga una proporción similar a la del dataset. Cada imagen del conjunto de entrenamiento ha sido etiquetada con una descripción en formato de texto, en la que se especifica el género y la raza de la persona que aparece en la imagen.

8.1.3. Entrenamiento

El entrenamiento se realiza utilizando la función fit() de Finetuner, que ajusta los pesos del modelo a los datos de entrenamiento. En este caso, se está utilizando la función de pérdida *CLIPLoss*, que es una función de pérdida personalizada diseñada específicamente para el modelo CLIP, que mide la similitud entre la representación vectorial de una imagen y su descripción mediante el cálculo de la distancia coseno entre los dos vectores. Además, se ha utilizado una tasa de aprendizaje de 1e-5 buscando una convergencia segura, un batchsize de 32 y 30 como número de epochs.

8.1.4. Hipótesis

Inicialmente, se escogió reentrenar CLIP maximizando su precisión general, bajo la hipótesis de que esto reduciría el sesgo en la clasificación final de pares imagen-texto. Se asumió que si CLIP relacionaba cada imagen con su texto de manera más precisa, esto reduciría el sesgo.

Sin embargo, se concluyó que esta hipótesis no es necesariamente cierta. Que CLIP mejore su precisión no garantiza que vaya a mejorar para todas las clases. Es posible que al maximizar su precisión solo aumente la misma en las clases no afectadas y esto aumentaría el sesgo.

8.1.5. Métricas

Evaluación

Aplicar ajuste fino al modelo CLIP con el dataset original UTKFace produce mejoras leves, de apenas una centésima, en las métricas de evaluación de género. Se observa una mejora notable en cuanto a la raza, aumentando la precisión-macro del 68 % al 90,8 % y la F1-macro del 67,6 %al 85,2 %.

Al realizar el ajuste fino con el dataset curado, se observó una disminución del 4 % en las métricas de género, mientras que en raza se obtuvo un aumento cercano al 10 %.

Dataset	Precisión-macro	F1-macro
Original género	95.9 %	95.8 %
Curado género	91.3 %	91.3 %
Original raza	90.8 %	85.2 %
Curado raza	77.5 %	77.2 %

Cuadro 1: Métricas de evaluación para los distintos datasets.

Equidad

Género

El ajuste logró mitigar los problemas presentes en impacto dispar y las probabilidades igualadas en ambos conjuntos de datos.

Raza

Grupos	**Impacto dispar**
White/Black	0.387
White/Asian	0.264
White/Indian	0.318
Black/Asian	0.682

Cuadro 2: Valores de impacto dispar en UTKFace

Grupos	**Grupo Privilegiado**	**Probabilidades igualadas**
Blanco-Negro	Blanco	0.122
Blanco-Asiatico	Blanco	0.221
Blanco-Indio	Blanco	0.182

Cuadro 3: Valores de probabilidades igualadas en UTKFace

Grupos	Impacto dispar
Blanco/Negro	0.788
Blanco/Asiatico	0.663
Blanco/Indio	0.724

Cuadro 4: Valores de impacto dispar en UTKFace curado

8.1.6. Conclusiones

A pesar de que el dataset curado reduce las métricas de evaluación en el caso del género, consigue muy buenos resultados en las métricas de equidad. Las probabilidades igualadas satifacen su hipótesis, reduce los grupos con impacto dispar presente y los valores que aún no se satisfacen crecen a niveles cercanos a los aceptables.

8.2. Ajuste fino con PyTorch

El funcionamiento de CLIP se basa en llevar las representaciones de imágenes y texto al mismo espacio vectorial, de forma tal que una imagen y su descripción serán vectores cercanos en dicho espacio mientras que las imágenes (o textos) que no coincidan quedan alejados. Siguiendo esta idea y las facilidades que brinda CLIP se ha reentrenado el modelo con el objetivo de mitigar el sesgo el la clasificación de género en imágenes.

8.2.1. Selección y procesamiento de datos

Las imágenes utilizadas corresponden al dataset $UTK - Face$ y como descripción asociada a la imagen un texto que expresa el género de la persona que aparece en la imagen (Ej: "Esto es una persona de género femenino"). Para el entrenamiento se utilizaron 2000 imágenes, 1000 de mujeres e igual cantidad de hombres y para la validación 400 imágenes, 200 de cada género. La cantidad de datos utilizados estuvo limitada por los recursos computacionales disponibles.

8.2.2. Entrenamiento

Se experimentó con distinto número de épocas y finalmente el modelo que se presenta se entrenó durante 10 épocas.

El modelo se entrena utilizando un descenso de gradiente estocástico con una función de pérdida de entropía cruzada. En cada época se procesa el conjunto de datos de entrenamiento en lotes. El modelo recibe por cada imagen dos posibles descripciones, una para cada género, y devuelve los valores de logit correspondientes a cada uno. Luego se evalúa la función de pérdida con dichos valores y el vector de $one - hot - encoding$ con la clase correcta. A continuación el modelo actualiza los pesos del modelo mediante la propagación hacia atrás.

Para tratar de mitigar el sesgo se aplica como restricción que el valor de la igualdad de las probabilidades se mantenga por debajo del 10 %. Si esto no se

cumple, no se actualiza el modelo mejor. De esta forma se trata de maximizar la precisión del modelo manteniendo un nivel de igualdad entre los grupos.

Después de cada época, se evalúa el modelo en el conjunto de datos de validación, calcula la pérdida y actualiza el mejor modelo si mejora la pérdida de validación y se mantiene en los niveles de equidad establecidos. La restricción de equidad se aplica mediante el criterio de probabilidades igualadas, que requiere que la tasa de verdaderos positivos (TPR) y la tasa de falsos positivos (FPR) sean iguales en diferentes grupos definidos por el atributo protegido, como el género en este caso. No se aplica un criterio tan estricto como que sean iguales, sino que se define como umbral que la diferencia absoluta entre TPR o FPR para cualquier par de grupos sea menor que el 10 %. Si es así, se considera que el modelo satisface el criterio de probabilidades igualadas. El mejor modelo (es decir, el modelo con la pérdida de validación más baja que satisface el criterio de probabilidades igualadas) es el que se termina utilizando.

8.2.3. Resultados obtenidos

Una vez obtenido el nuevo modelo, se evaluó en la colección de prueba la cual consistió en casi 5000 imágenes distintas del dataset $UTK - Face$. A continuación se muestran los resultados obtenidos.

Precisión y medida F1

Como métrica de precisión se utilizó precisión macro y se obtuvo 0.93 y como F1 macro se obtuvo también 0.93. Estos valores se quedan por debajo de los obtenidos en el modelo CLIP original.

Medidas de equidad

La figura 12 muestra la matriz de confusión obtenida.

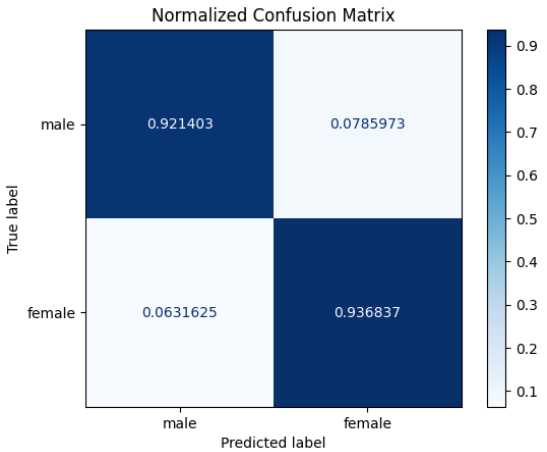

Figura 12: Matriz de confusión.

Las tasas de verdaderos positivos (TPR) para el género femenino y masculino son 0.92 y 0.94, respectivamente. Por otro lado, las tasas de falsos positivos (FPR) son 0.063 y 0.079 para el género femenino y masculino, respectivamente. La diferencia en las tasas de verdaderos positivos y falsos positivos es de 0.015, lo que sugiere que la igualdad en las probabilidades se cumple. Además, se calculó la razón en las tasas de selección para mujeres y hombres, la cual resultó en 0.9. Esto indica que el modelo no presenta impacto dispar, ya que la razón en las tasas de selección es cercana a 1, lo que sugiere que el modelo no favorece a un género por encima del otro.

8.3. Eliminación de sesgo por subespacio

Se propone un método denominado "transformación a nivel subespacial"para la mitigación del sesgo en cuestión, que de manera general, consta de los siguientes pasos: [6]

1. Obtener las representaciones de imagen y texto de diferentes subespacios: Para obtener estas representaciones, se alimenta el modelo con imágenes y texto que han sido etiquetados con información del atributo que se quiera mitigar, que genera un conjunto de representaciones de imagen y texto para cada clase.

2. Calcular la relevancia entre cada subespacio de salida y la información del atributo: Utilizando técnicas de análisis de componentes principales y heurísticas, se identifican los subespacios de salida que están más relacionados con la información necesaria. La relevancia se calcula midiendo la diferencia en los valores de salida del modelo para diferentes clasificaciones del atributo.

3. Eliminar el sesgo de los modelos mediante la manipulación de los subespacios: Una vez que se han identificado los subespacios del atributo relevantes, se procede a reducir el sesgo en el modelo. Se proyecta cada vector de representación de salida en el subespacio del atributo y se resta esta proyección del vector original de representación.

[6]

En lugar de intentar curar los datos de entrenamiento para eliminar el sesgo de género, este enfoque se centra en la eliminación del sesgo durante la inferencia, después de que el modelo ya ha sido entrenado, a partir de los resultados obtenidos.

A continuación se argumenta el trabajo realizado para la mitigación del sesgo presente en el género en el modelo CLIP sobre el dataset UTKFace.

En primer lugar, se obtienen las representaciones de diferentes géneros para el lenguaje y la visión. Se ha tomado para esta tarea un conjunto de cuatrocientas imágenes y cuatrocientos fragmentos de textos clasificados en género, garantizando doscientos de cada tipo de género, dependiendo de si en las imágenes se encuentra una persona de género femenino o masculino, y de si en cada texto se habla de alguien de algún género específico, en cada caso. Para

ello, se han utilizado los datasets de $UTKFace$ [32] y $MDGender(Multi-DimensionalGenderBiasDatasets)$ [33] de imagen y texto respectivamente, clasificados en género. Se ha utilizado dentro de $MDGender$ el dataset denominado $ImageChat$, que se extrajo de descripciones de imágenes, para que tuviera relación contextual con el subespacio que se quiere identificar, y por tanto, con el modelo que se quiere mejorar.

Después de obtener las representaciones de diferentes entradas para cada género, la única diferencia entre los diferentes conjuntos debería ser el atributo de género presente. Se puede asumir que existe un subespacio que responde principalmente a la información de género [6]. Para obtener este subespacio, podemos utilizar el análisis de componentes principales (PCA). Específicamente, primero se calcula la media del conjunto j como $\mu_j = \frac{1}{P}\sum_{w \in R_j} w$. Luego, se puede realizar PCA en la unión de los conjuntos de todas las $|R_j|$ representaciones de género:

$$V = PCA_k\Big(\bigcup_j \bigcup_{w \in R_j} (w - \mu_j)\Big), \tag{1}$$

donde k es un hiperparámetro y V es el subespacio resultante [6]. PCA permite encontrar el subespacio donde las representaciones difieren más, lo que encaja bien con el objetivo y, por lo tanto, V puede tratarse como el subespacio de género.

Para escoger el valor de k se realizó una gráfica con los ejes: valor de k y varianza acumulada; se escogió el menor k que explicaba aproximadamente el 100 % de la varianza.

Después de obtener V, se realiza la mitigación. Dada una representación de salida h, de un vector que representa una imagen o un texto dado por los codificadores del modelo, se puede proyectar en el subespacio de sesgo $h_v = \sum_{j=1}^{k} \langle h, v_j \rangle v_j$ y restar esta proyección de la representación original: $\hat{h} = h - h_v$. El vector resultante será ortogonal al subespacio de sesgo y, por lo tanto, se puede aliviar el sesgo. [6]

Como se observa, los v_j son los vectores propios que representan el subespacio V. Computacionalmente, y a propósito de la eficiencia, se tiene previamente calculado el producto de las matrices de los vectores propios y su transpuesta, para luego, multiplicar la matriz de los vectores obtenidos en los resultados por esta matriz $V * V^T$.

Al experimentar con la fórmula descrita para restar la parte que produce sesgo que se quiere de la proyección original ($\hat{h} = h - h_v$), se pudo observar que se invertía el género completamente, cuando la idea era neutralizarlo en ese sentido; se obtuvo primeramente un modelo que tenía resultados peores en medidas de precisión que un modelo aleatorio. Por tanto, se tomó la decisión de multiplicar el valor h_v de la fórmula por una constante α, que al experimentar, el balance con mejor precisión y mitigación de sesgo se pudo comprobar que era para $\alpha = 0{,}4$, lo que significa que se resta un valor menor, y cada vector resultante se acerca más al original, y produce la neutralización del atributo que se buscaba.

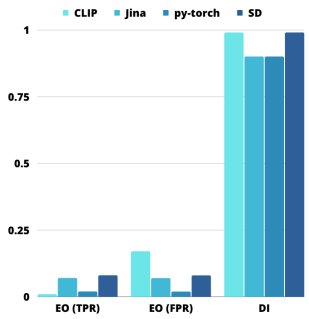

Figura 13: Comparación de sesgo en género para cada estrategia.

8.3.1. Resultados de precisión

Medida	Valor
Macro accuracy	0,778
Macro F1	0,779

8.3.2. Resultados de justicia

Medida	Valor	
Impacto dispar	0,9939674340077338	
Igualdad de probabilidades	TPR	FPR
	0,07596311224720897	0,07596311224720909

Los resultados, a pesar de empeorar en precisión, reducen el sesgo presente en cuanto a género en el modelo original.

8.4. Conclusiones

9. Conclusiones

10. Recomendaciones

Referencias

[1] O. Aghav : Age and gender estimation using cnn. `https://www.kaggle.com/code/omkyaghav/age-and-gender-estimation-using-cnn`, 2023.

[2] J. AI : Finetuner. `https://finetuner.jina.ai/`, 2021.

[3] J. AI : Jina documentation. `https://docs.jina.ai/`, 2021.

[4] T. Bolukbasi, K.-W. Chang, J. Y. Zou, V. Saligrama et A. T. Kalai : Man is to computer programmer as woman is to homemaker? debiasing word embeddings. *Advances in neural information processing systems*, 29, 2016.

[5] J. Buolamwini et T. Gebru : Gender shades: Intersectional accuracy disparities in commercial gender classification. *In Conference on fairness, accountability and transparency*, p. 77–91. PMLR, 2018.

[6] F. Chen et Z.-Y. Dou : Measuring and mitigating bias in vision-and-language models. *In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, p. 701–712, 2021.

[7] V. Cortez : How to define fairness to detect and prevent discriminatory outcomes in machine learning. 2019.

[8] J. Dressel et H. Farid : The accuracy, fairness, and limits of predicting recidivism. *Science advances*, 4(1):eaao5580, 2018.

[9] C. Feijóo, Y. Kwon, J. M. Bauer, E. Bohlin, B. Howell, R. Jain, P. Potgieter, K. Vu, J. Whalley et J. Xia : Harnessing artificial intelligence (ai) to increase wellbeing for all: The case for a new technology diplomacy. *Telecommunications Policy*, 44(6):101988, 2020.

[10] Fiddler AI : Auditing the clip image classification model for bias, 2021.

[11] M. Kay, C. Matuszek et S. A. Munson : Unequal representation and gender stereotypes in image search results for occupations. *In Proceedings of the 33rd annual acm conference on human factors in computing systems*, p. 3819–3828, 2015.

[12] Keras : Convolution2d layer. `https://keras.io/api/layers/convolution_layers/convolution2d/`, 2021. Accessed: 2023-04-07.

[13] P. D. König et G. Wenzelburger : When politicization stops algorithms in criminal justice. *The British Journal of Criminology*, 61(3):832–851, 2021.

[14] J. L. S. M. y. J. A. Lauren Kirchner : Machine bias propublica.

[15] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman et A. Galstyan : A survey on bias and fairness in machine learning. *ACM Computing Surveys, Vol. 54, No. 6, Article 115.*, 2021.

[16] T. Mikolov, K. Chen, G. Corrado et J. Dean : Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[17] OpenAI : Clip: Connecting text and images. `https://openai.com/blog/clip/`, 2021.

[18] B. Paassen, A. Bunge, C. Hainke, L. Sindelar et M. Vogelsang : Dynamic fairness-breaking vicious cycles in automatic decision making. *arXiv preprint arXiv:1902.00375*, 2019.

[19] J. Pennington, R. Socher et C. D. Manning : Glove: Global vectors for word representation. *In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, p. 1532–1543, 2014.

[20] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.* : Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*, 2021.

[21] C. Schumann, J. Foster, N. Mattei et J. Dickerson : We need fairness and explainability in algorithmic hiring. *In International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, 2020.

[22] A. Seehorn : Geniolandia. `https://www.geniolandia.com/13176536/tipos-de-reflexiones-matematicas`, 2018. Reflexión Horizontal.

[23] M. Serawan : Demographics classification. `https://www.kaggle.com/code/mohamadserawan/demographics-classifications`, 2019.

[24] J. Stoyanovich, B. Howe et H. Jagadish : Responsible data management. *Proceedings of the VLDB Endowment*, 13(12), 2020.

[25] L. Sweeney : Discrimination in online ad delivery. *Communications of the ACM*, 56(5):44–54, 2013.

[26] M. Tripathi : Gender detection using cnn. `https://www.kaggle.com/code/dskagglemt/gender-detection-using-cnn`, 2022.

[27] E. Ward : Age and gender prediction on utkface. `https://www.kaggle.com/code/eward96/age-and-gender-prediction-on-utkface`, 2019. Accessed: 2023-04-05.

[28] K. Wiggers : Audit finds gender and age bias in openai's clip model. *VentureBeat*, 2021.

[29] Wikipedia : Gaussian blur. `https://en.wikipedia.org/wiki/Gaussian_blur`, 2023.

[30] Wikipedia : Gaussian noise. `https://en.wikipedia.org/wiki/Gaussian_noise`, 2023.

[31] Wikipedia contributors : Fine-tuning (machine learning) — Wikipedia, the free encyclopedia, 2023. [Online; accessed 4-June-2023].

[32] S. Y. Zhang, Zhifei et Q. Hairong : Age progression/regression by conditional adversarial autoencoder. *In IEEE Conference on Computer Vision and Pattern Recognition Workshops*, p. 208–216, 2017.

[33] J. Zhao, T. Wang, M. Yatskar, V. Ordonez et K.-W. Chang : Gender bias in neural natural language processing. *In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, p. 6214–6223, 2019.

[34] Z. Zhong : A tutorial on fairness in machine learning. 2018.