

Proyecto 2. Resultados Parciales y Visualizaciones Estáticas

Integrantes:

- Paul Belches - 17088
- José Cifuentes - 17509
- Oscar Juárez - 17315
- Diego Sevilla - 17238
- Alejandro Tejada - 17584

Investigación algoritmos

Redes Neuronales convolucionales

Las redes neuronales convolucionales es un algoritmo de Deep Learning que está diseñado para trabajar con imágenes, tomando estas como input, asignándole importancias (pesos) a ciertos elementos en la imagen para así poder diferenciar unos de otros. Este es uno de los principales algoritmos que ha contribuido en el desarrollo y perfeccionamiento del campo de observar imágenes en la computadora.

Las redes convolucionales contienen varias hidden layers (o capas ocultas), donde las primeras pueden detectar líneas, curvas y así se van especializando hasta poder reconocer formas complejas como un rostro, siluetas, etc.

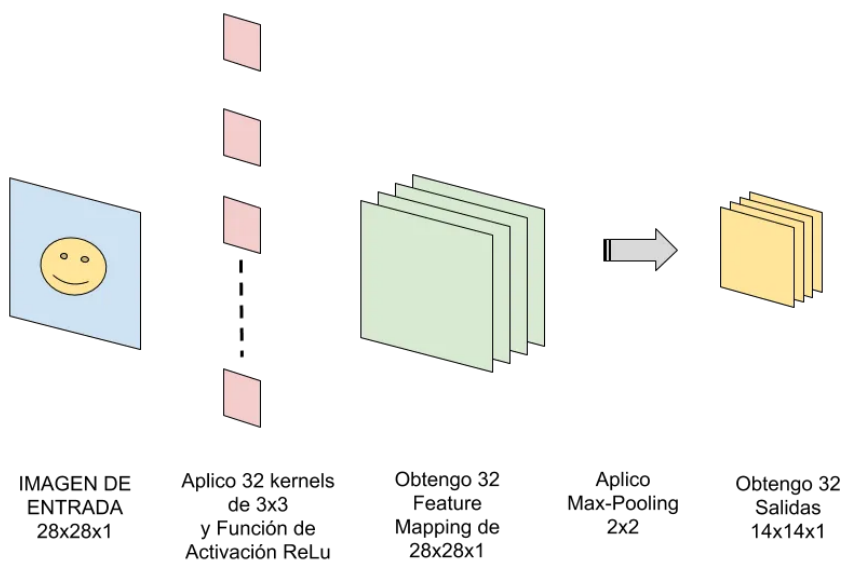
La CNN (Convolutional Neural Network) es un tipo de Red Neuronal Artificial con aprendizaje supervisado que procesa sus capas imitando a la corteza visual del ojo humano para identificar distintas características en las entradas que en definitiva hacen que pueda identificar objetos y "ver". Para ello, la CNN contiene varias capas ocultas especializadas y con una jerarquía: esto quiere decir que las primeras capas pueden

detectar líneas, curvas y se van especializando hasta llegar a capas más profundas que reconocen formas complejas como un rostro o la silueta de un animal.

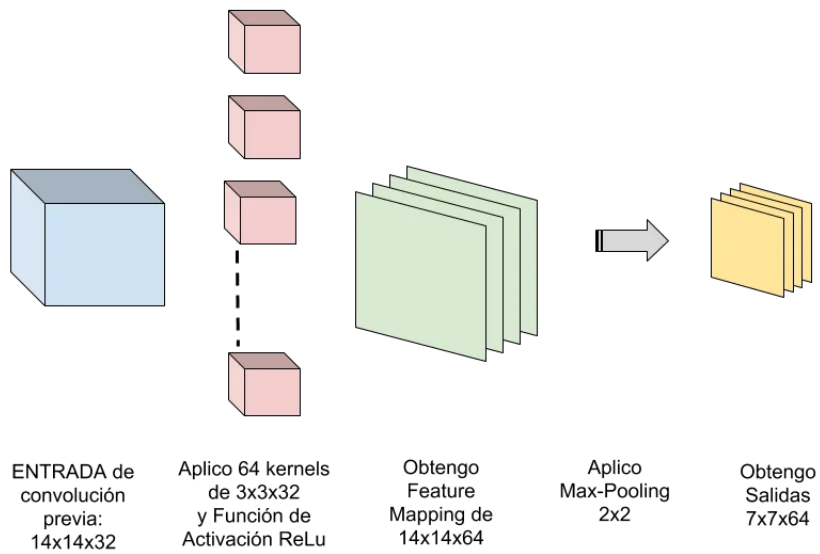
Etapas o Procesos

- Obtener la Imagen
- Preprocesar
- Convolucionar
- Filtrar
- Aplicar función de activación
- Subsampling
- Repetir con más convoluciones

PRIMERA CONVOLUCIÓN



SEGUNDA CONVOLUCIÓN (y sucesivas)



Aprendizaje por transferencia

Este no es tanto un algoritmo, sino una técnica usada en medicina. La idea parte de una premisa. Al llegar a entenderse algo, es posible transmitir ese conocimiento a otras personas. Claro, hablar de humanos es algo totalmente normal.

Ahora, aplicado a una máquina, simulemos que una inteligencia hace cierto trabajo y lo hace bastante bien. ¿Es posible usar ese conocimiento previo para otra aplicación? claro, existen algunas limitantes. Una es que todas las aplicaciones no son iguales, si bien comparten rasgos, es difícil encasillar una inteligencia que resuelva todo.

Algunos algoritmos que están disponibles para transferencia son las redes lógicas de Markov y redes bayesianas. Como se mencionó antes, las aplicaciones más prácticas van orientadas a la medicina, clasificación de textos, etc.

Una interesante aplicación se da en el siguiente artículo: **Transfer learning extensions for the probabilistic classification vector machine.**

En el siguiente link: <http://search.ebscohost.com/login.aspx?direct=true&db=aci&AN=143310182&lang=es&site=ehost-live>

Data Augmentation

Un problema habitual en deep learning es la gran cantidad de datos necesaria para el entrenamiento de modelos, lo que podemos solucionar con datos aumentados.

Imaginemos el siguiente caso: perritos y gatos. A partir de imágenes correctamente etiquetadas, entrenar una red que prediga si dada una imagen es perrito o gatito.

Es aquí donde entra la técnica Data Augmentation que, como indica su nombre, nos permite aumentar nuestro dataset de dos formas:

- Introduciendo perturbaciones en los datos originales. En el ejemplo de gatos y perros, tomando una imagen original centrada, la replicamos descentrada, invirtiendo ejes, etc.
- Utilizando distintas distribuciones. Ej: si queremos entrenar un modelo para clasificar imágenes de alta resolución, añadimos imágenes de baja resolución, con ruido... siempre manteniendo la mayor proporción de alta resolución.

Validación de modelos

Al desarrollar un algoritmo en machine learning, partimos de 3 datasets:

- Training set: es el conjunto de datos sobre el que se entrena el modelo. Consiste en pares correctamente etiquetados, es decir, un vector de valores y la variable respuesta.
- Validation set: el modelo ajustado en el paso anterior es usado para predecir la variable respuesta en este segundo dataset. Esto nos permite tener una evaluación no sesgada del modelo. En esta fase podemos aplicar regularización si se observa overfitting (el modelo tiene más capacidad de memorizar los datos de entrenamiento que de aprender de los nuevos).
- Test set: evaluamos el modelo final ajustado en el training set.

Al emplear data augmentation tendremos 4 datasets, añadiendo al training set el training-validation set, ambos formados por datos reales y aumentados. El validation set y el test set deben proceder de datos reales.

Long Short-Term Memory (LSTM) networks

LSTM es un tipo de red neuronal usado en Deep learning. A diferencia de una red neuronal como Feedforward, LSTM tiene conexiones de Feedback. No solo procesa data como imágenes, sino también **secuencias** de data. Esto implica que puede procesar habla o video. Algunas aplicaciones son la detección de escritura o la detección de discursos, tráfico, etc.

La conforman 4 partes

- Una célula o neurona
- Una input gate
- Un Output Gate
- Y una "forget" gate

Se usan principalmente en clasificar, procesar y hacer predicciones basadas en series de tiempo.

Como se dijo se usan para aplicaciones de secuencias o plazos largos, ya que guardan la información en bucles internos.

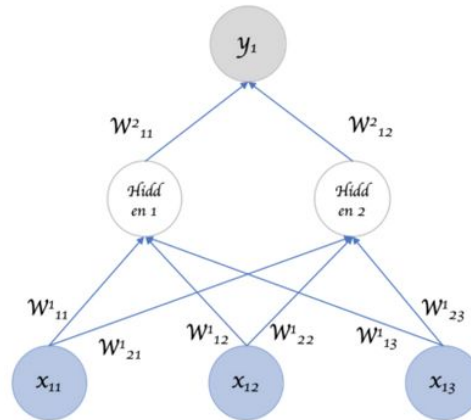
Algunas interesantes aplicaciones:

- <http://search.ebscohost.com/login.aspx?direct=true&db=aci&AN=131542910&lang=es&site=ehost-live>

- <http://search.ebscohost.com/login.aspx?direct=true&db=aci&AN=146051241&lang=es&site=ehost-live>
- <http://search.ebscohost.com/login.aspx?direct=true&db=aci&AN=143384440&lang=es&site=ehost-live>
- <http://search.ebscohost.com/login.aspx?direct=true&db=a9h&AN=143441807&lang=es&site=ehost-live>
- <http://search.ebscohost.com/login.aspx?direct=true&db=aci&AN=145159955&lang=es&site=ehost-live>

Recurrent neural network (RNN)

Las redes neuronales recurrentes (RNN) agregan un giro interesante a las redes neuronales básicas. Una red neuronal básica toma como entrada un vector de tamaño fijo que limita su uso en situaciones que involucren una entrada de tipo "serie" sin un tamaño predeterminado.



Los RNN están diseñados para aceptar una serie de entradas sin límite predeterminado de tamaño. Uno podría preguntarse cuál es el problema, ¿Puedo llamar repetidamente a una red neuronal básica también, y el problema estaría resuelto no?

Claro que sí, pero la parte "serie" de la entrada significa algo. Un solo elemento de entrada de la serie está relacionado con otros y probablemente influya en sus vecinos. De lo contrario, son solo "muchas" entradas, no una entrada en "serie".

Por tanto, necesitamos algo que capture esta relación entre las entradas de forma significativa.

La red neuronal recurrente recuerda el pasado y sus decisiones están influenciadas por lo que ha aprendido del pasado. Nota: Las redes de forward propagation básicas también "recuerdan" cosas, pero recuerdan cosas que aprendieron durante el entrenamiento. Por ejemplo, un clasificador de imágenes aprende cómo se ve un "1" durante el entrenamiento y luego usa ese conocimiento para clasificar las cosas en producción.

En resumen, en una red neuronal básica, un vector de entrada de tamaño fijo se transforma en un vector de salida de tamaño fijo. Una red de este tipo se vuelve "recurrente" cuando aplica repetidamente las transformaciones a una serie de entradas dadas y produce una serie de vectores de salida. No hay una limitación preestablecida para el tamaño del vector. Y, además de generar la salida que es una función de la entrada y el estado oculto, actualizamos el estado oculto en sí en función de la entrada y lo usamos para procesar la siguiente entrada.

Existen muchas aplicaciones de este tipo de redes neuronales hoy en día. En el siguiente link se mencionan varias:

<https://iq.opengenus.org/applications-of-rnn/>

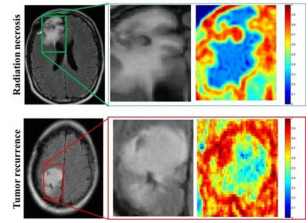
Problemas extras que son presentados en medicina

La medicina y la salud es una industria revolucionaria y prometedora para implementar soluciones con data science. El análisis de datos está llevando la ciencia médica a un nivel completamente nuevo, desde la digitalización de los registros médicos hasta el descubrimiento de fármacos y la exploración de enfermedades genéticas. Y esto es sólo el principio.

La atención médica y data science a menudo están vinculadas a través de las finanzas, ya que la industria intenta reducir sus gastos con la ayuda de grandes cantidades de datos. La ciencia de datos y la medicina se están desarrollando rápidamente y es importante que avancen juntas.

Análisis de imágenes médicas

El sector de la salud recibe grandes beneficios de la aplicación de la ciencia de datos en imágenes médicas. Hay mucha investigación en esta área, y uno de los estudios más importantes es Big Data Analytics in Healthcare, publicado en BioMed Research International. Según el estudio, las técnicas de imágenes más populares incluyen la resonancia magnética (IRM), los rayos X, la tomografía computarizada, la mamografía, etc. Se utilizan numerosos métodos para abordar la diferencia de modalidad, resolución y dimensión de estas imágenes. Se están desarrollando muchos más para mejorar la calidad de la imagen, extraer datos de las imágenes de manera más eficiente y proporcionar la interpretación más precisa. Los algoritmos basados en deep learning aumentan la precisión del diagnóstico al aprender de los ejemplos anteriores y luego sugieren mejores soluciones de tratamiento.



Genética y Genómica

La investigación en genética y genómica permite un nivel avanzado de personalización del tratamiento. El objetivo es comprender el impacto del ADN en nuestra salud y encontrar conexiones biológicas individuales entre la genética, las enfermedades y la respuesta a los medicamentos. Las técnicas de ciencia de datos permiten la integración de diferentes tipos de datos con datos genómicos en la investigación de enfermedades, lo que proporciona una comprensión más profunda de los problemas genéticos en las reacciones a medicamentos y enfermedades particulares.

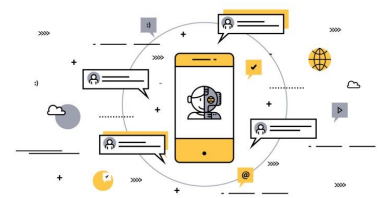
Asistente virtual para pacientes y atención al cliente

La optimización del proceso clínico se basa en el concepto de que en muchos casos no es realmente necesario que los pacientes visiten al médico en persona. El uso de una aplicación móvil puede ofrecer una solución más eficaz al "llevar al médico al paciente". Las aplicaciones móviles impulsadas por IA pueden proporcionar asistencia sanitaria básica, normalmente como chatbots. Simplemente describe sus síntomas o hace preguntas y luego recibe información clave sobre su condición médica derivada de una amplia red que vincula los síntomas con las causas. Las aplicaciones pueden recordarle que debe tomar su medicamento a tiempo y, si es necesario, asignar una cita con un médico. Este

enfocado promueve un estilo de vida saludable al alentar a los pacientes a tomar decisiones saludables, les ahorra tiempo esperando en la fila para una cita y les permite a los médicos enfocarse en casos más críticos.

Creación de medicamentos

El proceso de descubrimiento de fármacos es muy complicado e involucra muchas disciplinas. Las mejores ideas a menudo están limitadas por millones de pruebas, una gran inversión financiera y de tiempo. En promedio, se necesitan doce años para que un medicamento se presente oficialmente. Los algoritmos de ciencia de datos y aprendizaje automático simplifican y acortan este proceso, agregando una perspectiva a cada paso, desde la selección inicial de compuestos farmacológicos hasta la predicción de la tasa de éxito basada en factores biológicos. Estos algoritmos pueden pronosticar cómo actuará el compuesto en el cuerpo utilizando modelos y simulaciones matemáticas avanzadas en lugar de los "experimentos de laboratorio". La idea detrás del descubrimiento de fármacos computacionales es crear simulaciones de modelos informáticos como una red biológicamente relevante que simplifica la predicción de resultados futuros con alta precisión. Permite elegir qué experimentos se deben realizar e incorpora toda la información nueva en un ciclo de aprendizaje continuo. Se utilizan técnicas análogas para predecir los efectos secundarios de algunas combinaciones químicas particulares.



Medicina predictiva: pronóstico y precisión diagnóstica

Los métodos de análisis predictivo de la ciencia de datos aprenden de los datos históricos y hacen predicciones precisas sobre los resultados. Procesan los datos del paciente, dan sentido a las notas clínicas, encuentran las correlaciones, asociaciones de síntomas, antecedentes familiares, hábitos, enfermedades y luego hacen predicciones. Los impactos de determinados factores biomédicos como la estructura del genoma o las variables clínicas se tienen en cuenta para predecir la evolución de determinadas enfermedades. Los casos comunes incluyen el pronóstico del progreso de la enfermedad



la prevención para reducir el riesgo y los resultados negativos. El principal beneficio es la mejora de la calidad de vida de los pacientes y la calidad de las condiciones laborales de los médicos.

Class Imbalance (Weighted loss / Resampling)

Un problema de class imbalance es un ejemplo de un problema de clasificación en el que la distribución de ejemplos entre las clases conocidas está sesgada. La distribución puede variar desde un ligero sesgo hasta un desequilibrio severo cuando hay un ejemplo en la clase minoritaria por cientos, miles o millones de ejemplos en la clase o clases mayoritarias.

Las clasificaciones desequilibradas plantean un desafío para el modelado predictivo, ya que la mayoría de los algoritmos de aprendizaje automático utilizados para la clasificación se diseñaron en torno a la suposición de un número igual de ejemplos para cada clase. Esto da como resultado modelos que tienen un rendimiento predictivo deficiente, específicamente para la clase minoritaria. Esto es un problema porque típicamente, la clase minoritaria es más importante y, por lo tanto, el problema es más sensible a los errores de clasificación para la clase minoritaria que para la clase mayoritaria.

Por ejemplo, podemos recopilar medidas de flores y tener 80 ejemplos de una especie de flor y 20 ejemplos de una segunda especie de flor, y solo estos ejemplos comprenden nuestro conjunto de datos de entrenamiento. Esto representa un ejemplo de un problema de clasificación desequilibrado.

“Se produce un desequilibrio cuando una o más clases tienen proporciones muy bajas en los datos de entrenamiento en comparación con las otras clases.”

Hay otros nombres menos generales que pueden usarse para describir este tipo de problemas de clasificación, como:

- Predicción de eventos raros.
- Predicción de eventos extremos.
- Desequilibrio de clases severo.

Causas del desequilibrio de clases

El desequilibrio en la distribución de clases en un problema de modelado predictivo de clasificación desequilibrada puede tener muchas causas. Quizás haya dos grupos principales de causas para el desequilibrio que tal vez queramos considerar; son muestras de datos y propiedades del dominio.

Es posible que el desequilibrio en los ejemplos en las clases se deba a la forma en que se recopilaron o muestrearon los ejemplos del dominio del problema. Esto puede implicar sesgos introducidos durante la recopilación de datos y errores cometidos durante la recopilación de datos.

- Muestreo sesgado.
- Errores de medición.

Muchos problemas de clasificación pueden tener un grave desequilibrio en la distribución de clases; sin embargo, observar los dominios de problemas comunes que están inherentemente desequilibrados hará que las ideas y los desafíos del desequilibrio de clases sean concretos.

- Detección de fraudes.
- Predicción de reclamos
- Predicción predeterminada.
- Predicción de abandono.
- Detección de spam.
- Detección de anomalías.
- Detección de valores atípicos.
- Detección de intrusiones
- Predicción de conversión.

Modelo Inicial y Selección de algoritmos

Tabla 1: Estructura de red neuronal inicial.

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 148, 148, 64)	1792
max_pooling2d (MaxPooling2D)	(None, 74, 74, 64)	0
conv2d_1 (Conv2D)	(None, 72, 72, 64)	36928
max_pooling2d_1 (MaxPooling2D)	(None, 36, 36, 64)	0
conv2d_2 (Conv2D)	(None, 34, 34, 128)	73856
max_pooling2d_2 (MaxPooling2D)	(None, 17, 17, 128)	0
conv2d_3 (Conv2D)	(None, 15, 15, 128)	147584
max_pooling2d_3 (MaxPooling2D)	(None, 7, 7, 128)	0
flatten (Flatten)	(None, 6272)	0
dropout (Dropout)	(None, 6272)	0
dense (Dense)	(None, 512)	3211776
dense_1 (Dense)	(None, 3)	1539

En los comienzos por resolver este problema, aún sabiendo que no era buena idea, se intentó usar la red neuronal descrita por la tabla 1. Como podemos observar, esta es una red neuronal convolucional bastante básica. Los resultados, como era de esperar, no superan el 10%

de rendimiento y 46 de MAE. Esto es dado que las imágenes no están distribuidas uniformemente y son muy pocas. Por esto mismo se optó por usar las siguientes técnicas para intentar mejorar estos rendimientos:

Aumentación de datos

Para los modelos que se presentarán a continuación se utilizó la aumentación de datos, esta es una técnica en donde se varían las imágenes que ya se tienen con el fin de crear nuevas para el entrenamiento. Los parámetros modificados para crear más volumen de entrenamiento son los siguientes:

Tabla 2: ajustes de parámetros de la aumentación de datos utilizada.

Parámetro	Valor
rescale	1./255
rotation_range	40
width_shift_range	0.2
height_shift_range	0.2
shear_range	0.2
zoom_range	0.2

Al utilizar esta técnica hay que tener mucho cuidado de modificar los parámetros que generan imágenes que son coherentes con el problema que estamos resolviendo. Por ejemplo, no es buena idea girar las radiografías verticalmente puesto que eso nunca va a pasar en la vida real.

Aprendizaje transferido

También se utilizó el aprendizaje transferido ya que al no contar con un buen set de entrenamiento era muy difícil entrenar desde cero una red neuronal. En esta ocasión se utilizó la red neuronal convolucional llamada **Inception V3** la cual es una red neuronal pre entrenada con mil clases y alrededor de un millón de imágenes (Milton, 2019).

En la forma en que estas redes se le pueden sacar provecho es que como podremos notar al ser entrenada con tan buen set de datos, esta ya es una red especializada en detectar vértices y formas por lo que podemos congelar casi toda su arquitectura y solamente volver a entrenar las últimas capas para afinar los resultados a nuestro problema.

Modelos Finales y Eficiencia de los Mismos

Tomando en cuenta las técnicas anteriormente mencionadas, se realizaron 3 modelos. Lo primero que quisimos variar es el resolver el problema tanto como un problema de regresión como uno de clasificación. También intentamos variar el rango de las etiquetas de cada imagen, es decir si normalmente las imágenes estaban etiquetadas en un rango de 1-228, reducir dicho rango a números de 0-1. A continuación se detallan los modelos realizados con las modificaciones y se muestran los resultados obtenidos:

Tomando el problema como un problema de regresión

Modelo 1

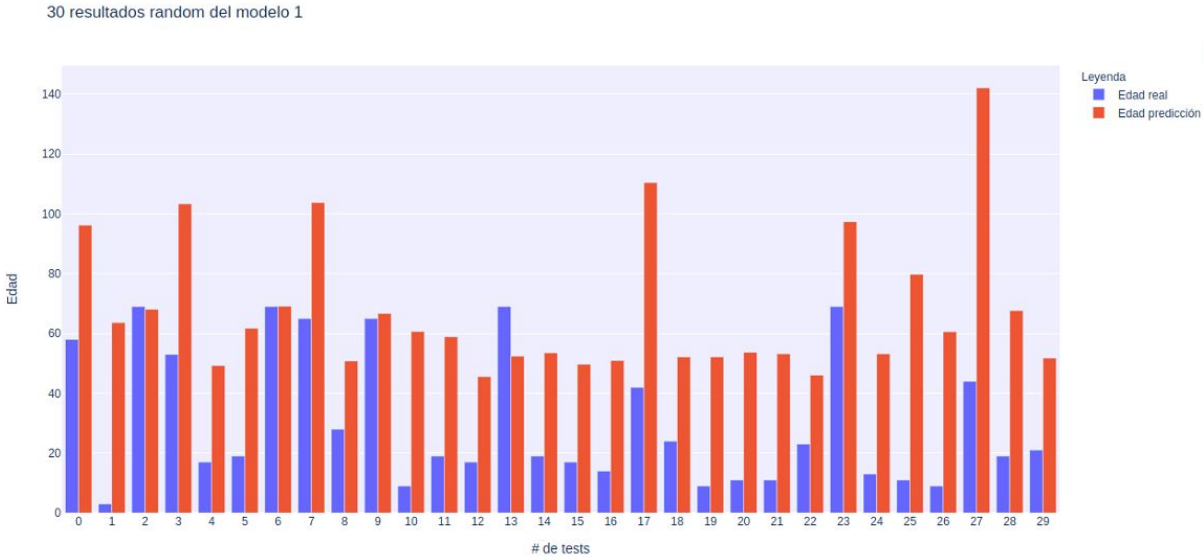
Para este modelo se dejaron las etiquetas de cada imagen en rangos de 1-228. Y por ser tomado como un problema de regresión la última capa de nuestro modelo solo tenía una neurona con activación lineal. Las predicciones obtenidas por este modelo son aproximaciones por lo que el indicador de rendimiento utilizado fue el error promedio absoluto (MAE).

Tabla 3: Resultados del modelo 1.

Set de datos	MAE
Entrenamiento	33.0112
Test	41.6575

En la tabla 3 observamos los resultados obtenidos del modelo, siendo 33 el error promedio absoluto durante el entrenamiento y 41.66 en el caso del test.

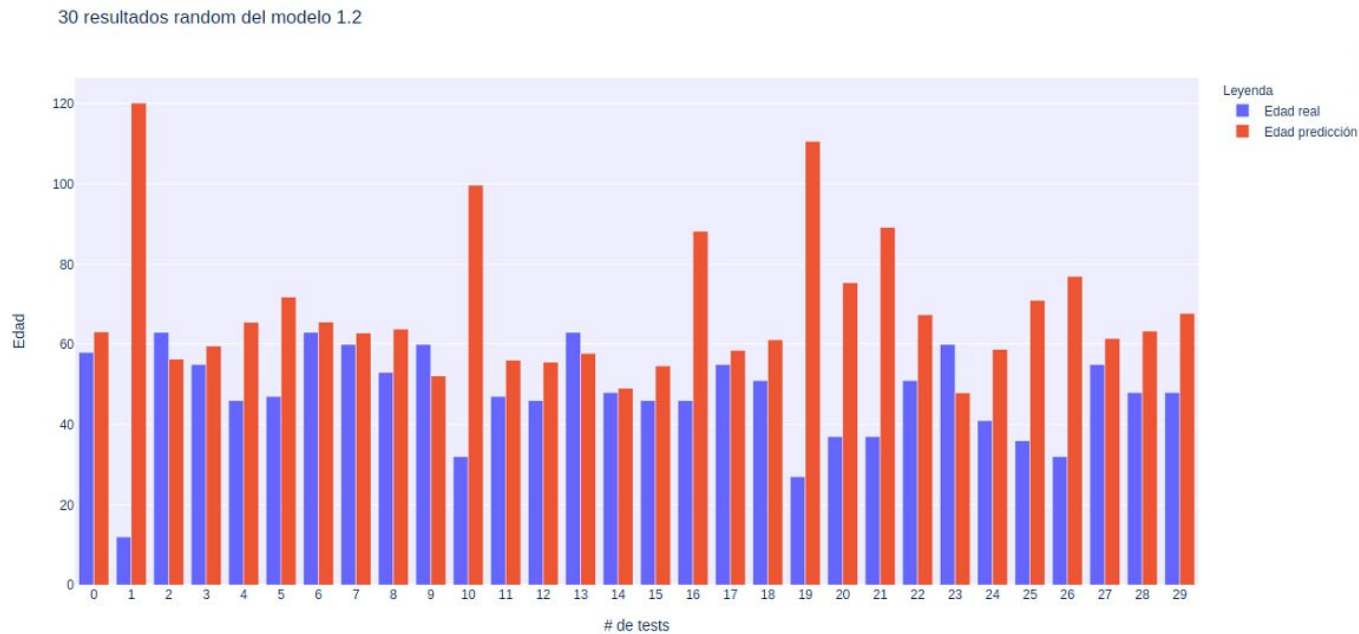
Figura 1: 30 resultados random del modelo 1.



En la figura 1 se muestran los resultados de 30 imágenes aleatorias. Muy pocas las predicciones se asemejan al valor real.

Posteriormente, se realizó una pequeña modificación a este primer modelo, entrenando con imágenes etiquetadas en rango de 1-228 y el test realizado con etiquetas de 0-1.

Figura 2: 30 resultados random del modelo 1 con modificación.



En la figura 2, podemos observar que las predicciones son mucho más cercanas al valor real, esta prueba obtuvo un MAE de 26.46. Los resultados presentados en la presente gráfica son interesantes puesto que el modelo no es preciso con edades muy pequeñas, como en el caso del test 1. Una hipótesis es que no se presentan muchas imágenes de edades pequeñas y el modelo no logró aprender cómo identificar esas edades.

Modelo 2

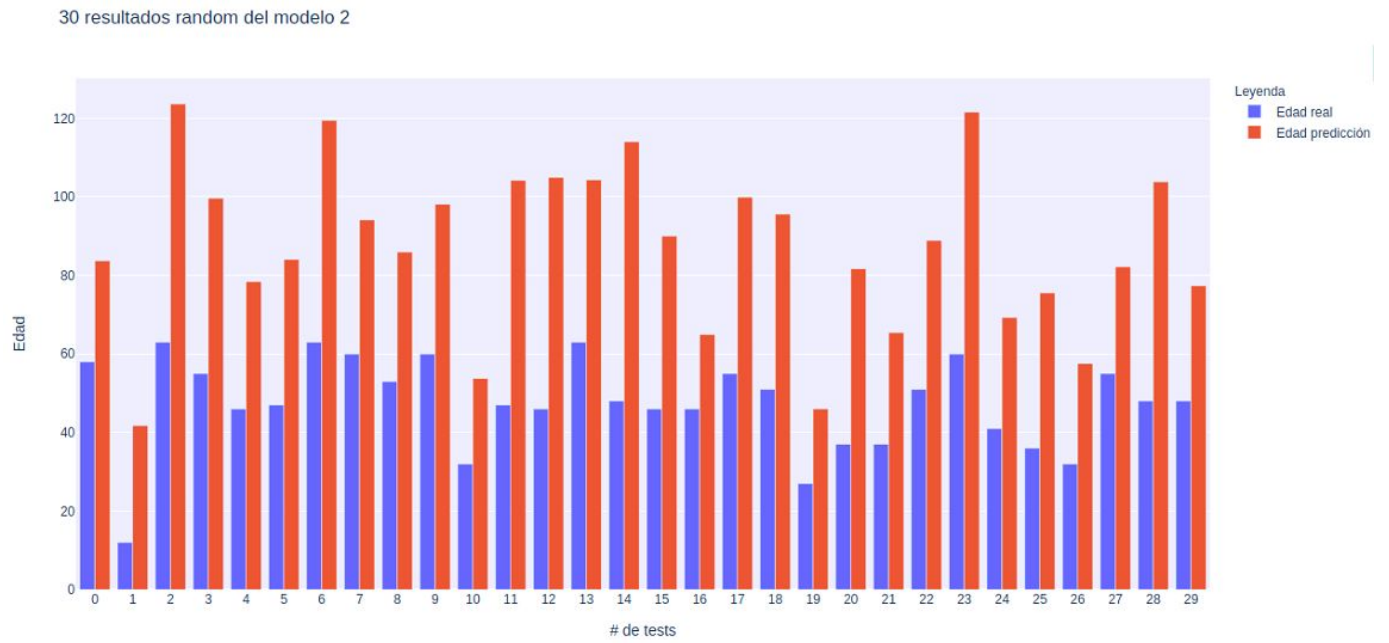
Para este modelo el único cambio que se realizó es que en lugar de tener las etiquetas de las imágenes de entre 1-228 se redujo a el rango de 0-1.

Tabla 4: resultados del modelo 2.

Set de datos	MAE
Entrenamiento	17.1465
Test	41.8897

En la tabla 4 podemos observar los resultados del modelo, si bien es cierto se obtuvo una mejora significativa en el MAE de entrenamiento, los resultados del test son casi iguales que el modelo anterior.

Figura 3: 30 resultados random del modelo 2.



Podemos observar que este modelo es un poco más consistente en cuanto a los resultados, es decir las predicciones en general son más cercanas al valor real. Sin embargo, la diferencia sigue siendo grande.

Tomando el problema como un problema de clasificación

Modelo 3

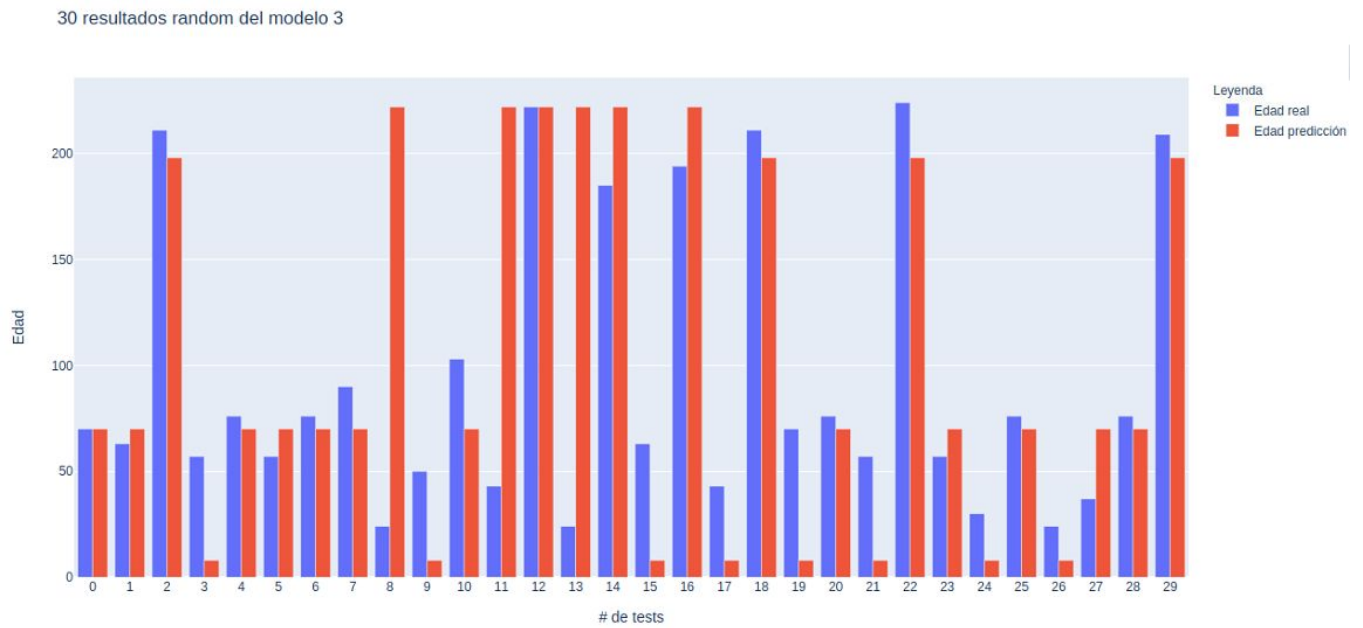
Para este modelo al ser tomado como un problema de clasificación, la última capa de nuestra arquitectura contaba con 228 neuronas y una activación de softmax.

Tabla 5: resultados del modelo 3.

Set de datos	Rendimiento
Entrenamiento	10.35%
Test	10.30%

En la tabla 5 podemos observar los resultados del modelo 3, si bien es cierto el rendimiento es super bajo los resultados no tenían tanta diferencia con el valor real, este modelo obtuvo un MAE de 91.20.

Figura 4: 30 resultados random del modelo 3.



En la figura 4, podemos notar que las predicciones si bien no son exactas, son muy similares al valor real. El MAE tan alto de este modelo puede ser consecuencia de tests como el número 8, en donde la predicción es muy distinta al valor real.

Recomendaciones de modelos

Para seguir mejorando el rendimiento de modelos que resuelvan este problema se podría optar por probar otras redes neuronales pre entrenadas para evaluar posibles mejoras de las predicciones. También se podría programar una función de error personalizada con el fin de nivelar el aporte de error para clases que cuentan con pocas imágenes y que no perjudiquen tanto el error global del modelo.

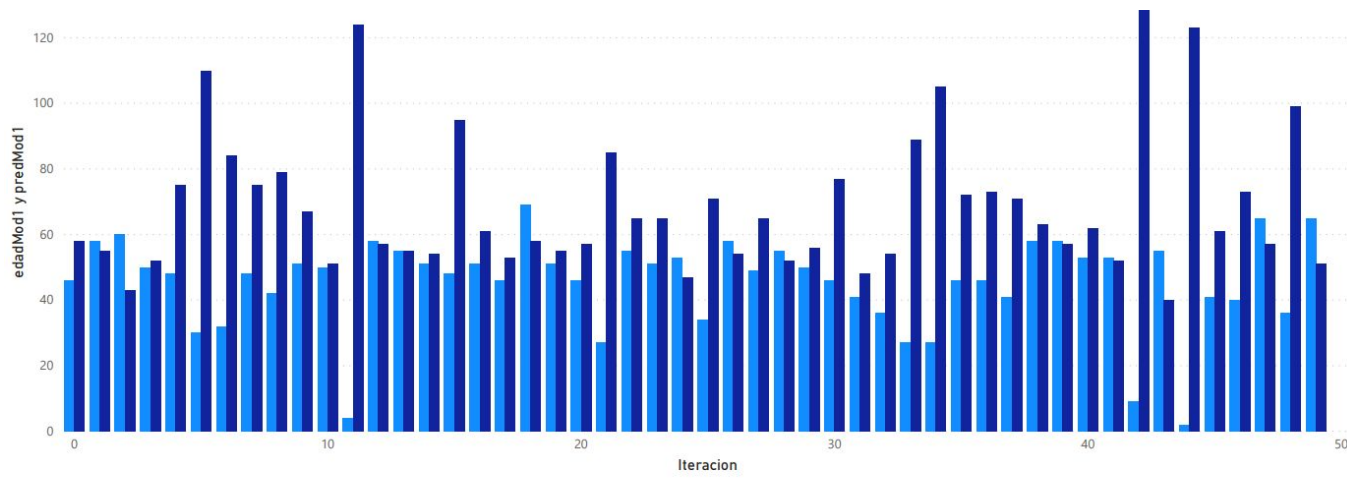
Discusión y Visualizaciones estáticas

Para esta sección, se tomaron dos modelos. El modelo #1, y el modelo#3. Luego de ver que ambos representaban un mejor rendimiento para su discusión. A partir de este punto, se utilizará la siguiente nomenclatura:

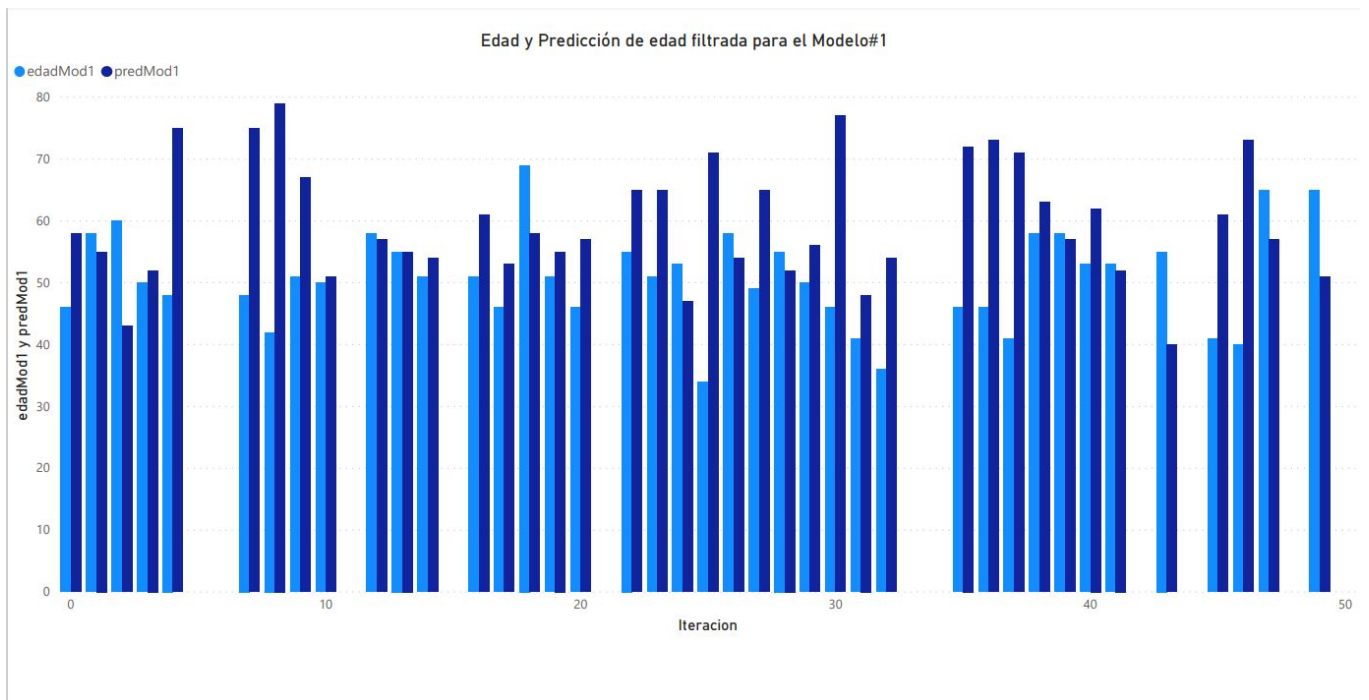
- **Modelo#1**: hace referencia al modelo#1.
- **Modelo#2**: hace referencia al modelo#3.
- **edadMod1**: hace referencia a la edad correspondiente en el modelo#1.
- **predMod1**: hace referencia a la edad que predice el modelo#1.
- **edadMod2**: hace referencia a la edad correspondiente en el modelo#3.
- **predMod2** hace referencia a la edad que predice el modelo#3.

Ahora las gráficas.

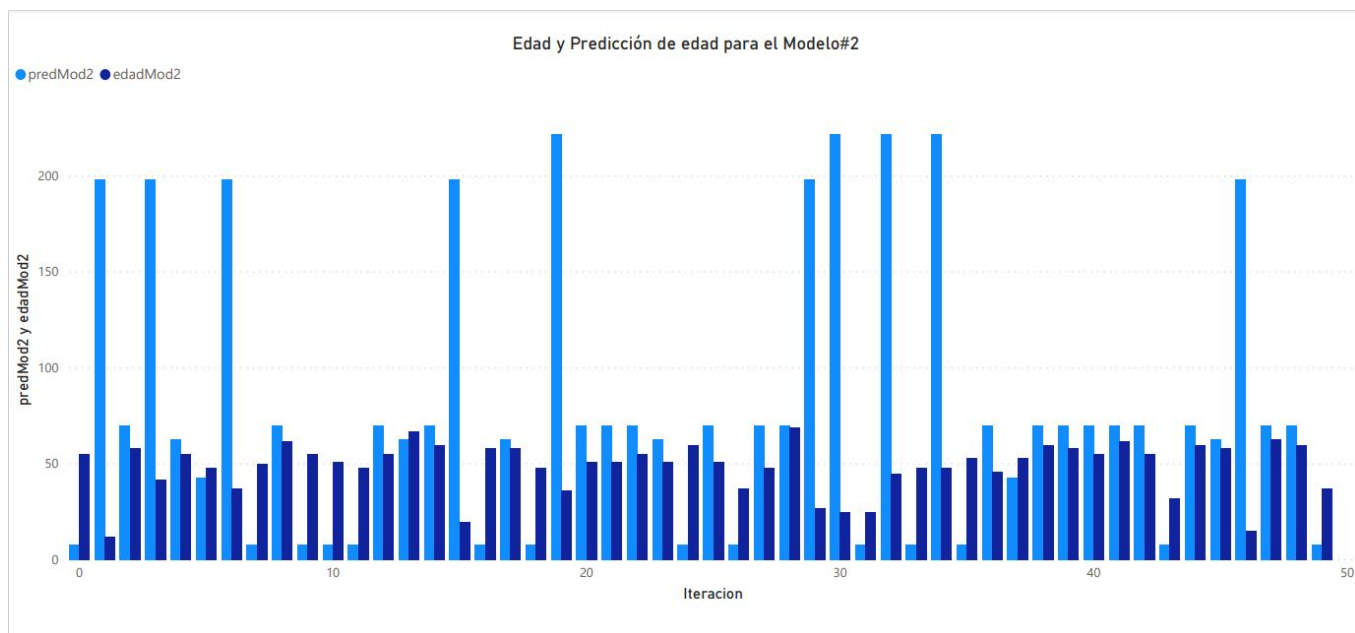
Gráfica 1: Edad y Predicción de edad para el Modelo#1



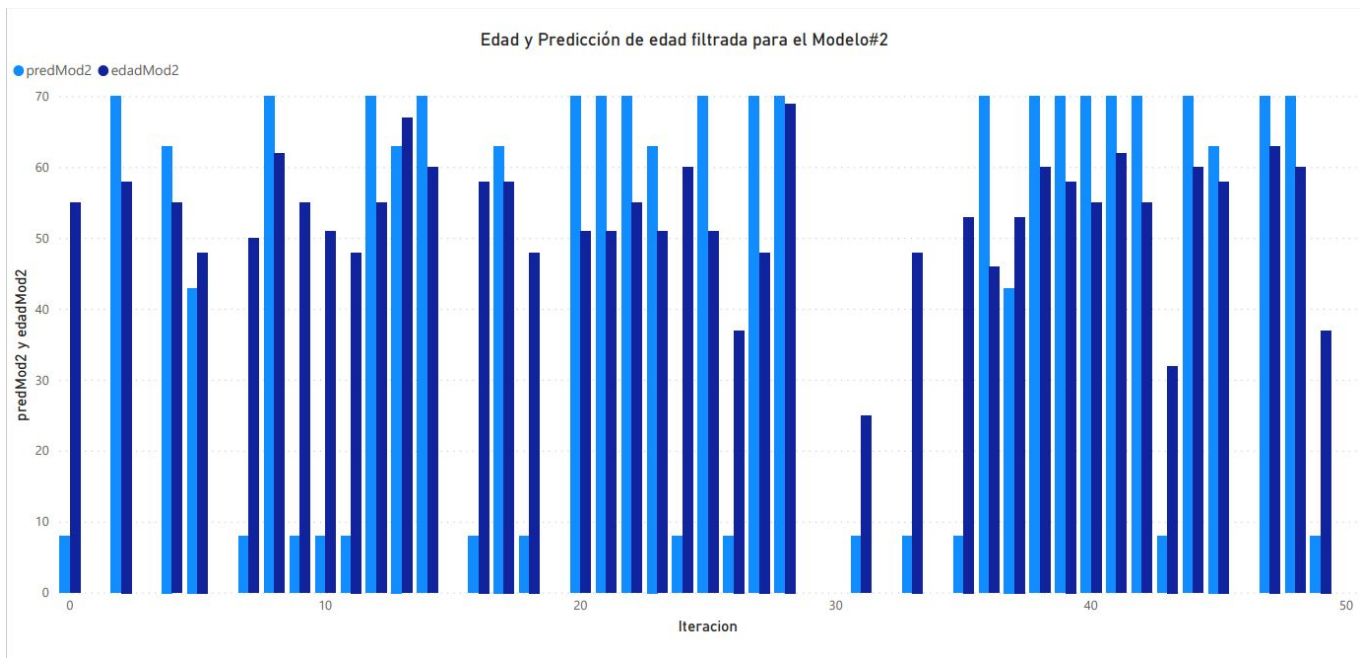
Gráfica 2: Edad y Predicción filtrada para el Modelo#1



Gráfica 3: Edad y Predicción para el Modelo#2



Gráfica 4: Edad y Predicción filtrada para el Modelo#2

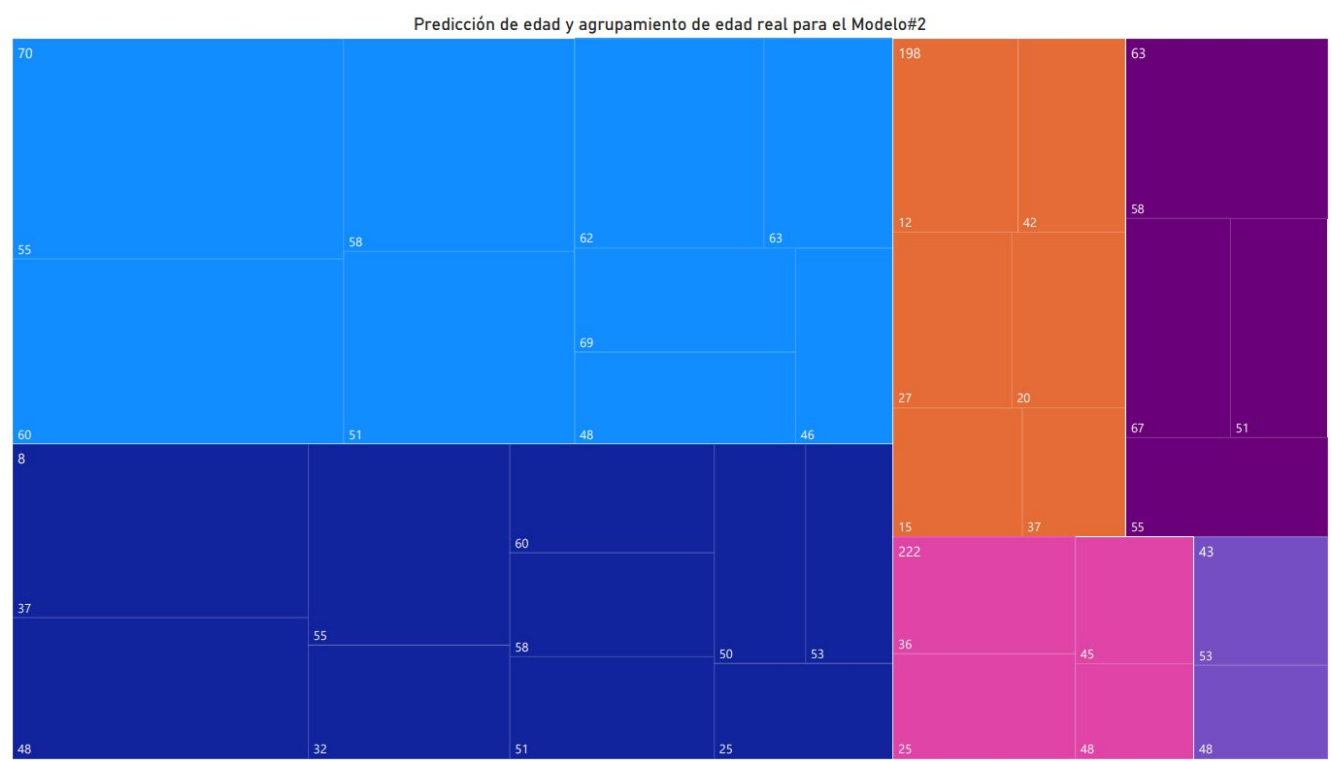


Gráfica 5: Predicción de edad y agrupamiento para el Modelo#1

Predicción de edad y agrupamiento de edad real para el Modelo#1



Gráfica 6: Predicción de edad y agrupamiento para el Modelo#2



Discusión

Los primeros cuatro gráficos muestran el contraste entre la predicción y el valor real para ambos modelos. Los siguientes cuatro muestran cómo se agrupan las predicciones según categorías. Al inicio, es subjetivo decir que tan acertado es una predicción conforme a su valor real. ¿Son 50 meses, 30 meses? ¿más o menos? Es la ambigüedad lo que llevó a la necesidad de establecer criterios de “aceptación” para las predicciones. Los criterios son los siguientes:

- Criterio#1: Una estimación será considerada atípica si el valor de la misma es igual al doble del valor real.
- Criterio#2: Una estimación será considerada no aceptable si cumple con el primer criterio, pero hay una incertidumbre de ± 20 meses entre el valor estimado y el real.

Para fines de escalas y de análisis, se realizaron 50 pruebas en ambos modelos.

La gráfica 1 muestra el resultado de las 50 pruebas en el modelo #1. En ella se muestran en contraste el valor real (azul claro) y el valor estimado (azul oscuro). Cada iteración es mostrada con su valor real y su valor estimado, por eso se ven agrupadas. Se nota que hay que valores atípicos, por ejemplo la iteración 11 muestra la mayor diferencia entre el valor real y el estimado, con un error de más de 100 meses. ¿Qué se puede extraer de la gráfica? Hay predicciones atípicas. ¿Cuántas? ¿En qué medida?

El gráfico 2 es el resultado de aplicar el criterio #1 al modelo #1 y por ende al gráfico #1. Es notorio la pérdida de datos ya que hay pequeños espacios donde faltan datos. Esos datos faltantes fueron omitidos porque no cumplieron el criterio #1. También la escala ha sido adaptada porque no hay valores estimados demasiado altos.

La cantidad de datos omitidos por el filtro es de 10 en total. Esto significa que un 80% de las estimaciones poseen una diferencia no mayor del doble del valor real. Al aplicar el segundo criterio se cuantifica un 23% que no cumple. Entonces, el global que si cumple es del 80% y de ese porcentaje, un 77% muestra una diferencia no mayor a la acordada para el modelo #1.(Pérez R. , 2011)

El gráfico 3 y 4 son los mismos criterios y filtros, pero aplicados al modelo#2. Los resultados de aplicar el primer criterio son iguales que el modelo #1: el 80% de las estimaciones lo cumplen. La diferencia radica en que ahora un 33% no cumple con el segundo criterio. Esto da un total de 80% global, y un 67% que cumple el segundo para el modelo #2. 10% menos que el primer modelo, lo que se traduce en 3 o 4 estimaciones crudas. De nuevo, se hace énfasis en que los gráficos 1 y 2 son el resultado del modelo y de aplicar filtros de mejora. Los gráficos 3 y 4 siguen la misma tendencia.

El gráfico 5 es interesante. Es un gráfico de área para el modelo #1. Primero, cada cuadro grande color representa **una predicción**, el valor dentro de ella (los subcuadros) es **la edad real**. De esa forma, se puede “agrupar” las edades que fueron estimadas como otra. Este

comportamiento es interesante de observar, ya que algunas edades fueron estimadas como otras y se agrupan dentro de los cuadros de las estimaciones. Por ejemplo el cuadro naranja tiene tres cuadros dentro: 55,53,50. Significa que tres edades fueron estimadas como el valor de ese cuadro : 52. Esto nos da una escala visual de que tan acertado es el algoritmo al momento de acercarse al valor. ¿Qué sería lo ideal? Que hubiera un cuadro por cada estimación y que fuera la correcta: un 100% de aciertos. Sería un overfitting pero muy agradable de ver. (BENBRAHIM, et al., 2020)

¿Habrá alguna diferencia entre el modelo#1 y el modelo#2 en el gráfico de área?

El gráfico 6 también es un gráfico de área pero para el modelo#2. Este muestra una notoria diferencia: hay menos cuadros grandes de colores. Esto indica que el algoritmo toma edades reales y las acierta con una cantidad menor de estimaciones. Mientras que el modelo#1 tenía cerca de 15 estimaciones distintas, el modelo#2 solo tiene 6. Esas 6 categorías se agrupan a las edades reales. En el gráfico 6 se nota que la edad estimada de 70 fue aplicada para muchas edades reales, y la edad de 8 también. Las edades estimadas que más agrupan edades reales son los 70 meses y los 8 meses. Ahí se concentra el 60% de los datos del modelo#2. Hay similitud de colores, pero **la cantidad de colores es menor**. Esa menor cantidad señala un comportamiento no muy acertado del modelo#2: tiene menos estimaciones posibles para distintas edades reales. El modelo#1 tiene más colores, o sea mas posibles valores, pero el modelo#2 no es tan bueno diferenciando las edades reales de las estimadas.

A partir de estas diferencias y consideraciones, se considera que el modelo#1 es mejor. Por razones como un mayor grado de aceptación de criterios, un alto porcentaje de aciertos, una mayor variedad de valores estimados y una mejor agrupación de sus valores respecto a los reales.

Conclusiones

- Por razones como un mayor grado de aceptación de criterios, un alto porcentaje de aciertos, una mayor variedad de valores estimados y una mejor agrupación de sus valores respecto a los reales el modelo #1 es mejor.
- El modelo#1 muestra un cumplimiento global del 80% para el primer criterio y un 77% para el segundo.
- El modelo#2 muestra un cumplimiento global del 80% para el primer criterio y un 67% para el segundo.
- El modelo#1 tiene mayor variedad de estimaciones y es más certero al agrupar.
- El modelo#2 tiene menor variedad de estimaciones y muchos datos se agrupan en solo dos categorías: 70 meses y 8 meses.

Referencias

- Durani, Y. (2016). Radiografía: estudio de la edad ósea. Extraído 07 de septiembre del 2020, de: <https://kidshealth.org/es/parents/xray-bone-age-esp.html#:~:text=La%20edad%20%C3%B3sea%20de%20un.de%20un%20problema%20de%20crecimiento>
- Pérez, R. (2011). Valoración y utilidad de la edad ósea en la práctica clínica. Extraído 05 de septiembre del 2020 de: <https://fapap.es/articulo/180/valoracion-y-utilidad-de-la-edad-osea-en-la-practica-clinica>
- BENBRAHIM, H., HACHIMI, H., & AMINE, A. (2020). Deep Convolutional Neural Network with Tensorflow and Keras to Classify Skin Cancer Images. Scalable Computing: Practice & Experience, 21(3), 379–389. <https://doi.org/10.12694/scpe.v21i3.1725>
- <https://www.aprendemachinelearning.com/como-funcionan-las-convolutional-neural-networks-vision-por-ordenador/>
- <https://medium.com/@bootcampai/redes-neuronales-convolucionales-5e0ce960caf8>
- https://es.wikipedia.org/wiki/Redes_neuronales_convolucionales
- Raab, C., & Schleif, F.-M. (2020). Transfer learning extensions for the probabilistic classification vector machine. Neurocomputing, 397, 320–330. <https://doi.org/10.1016/j.neucom.2019.09.104>
- <https://ecosistemahuawei.xataka.com/transferir-inteligencia-siguiente-reto-machine-learning/>
- Chaudhari, P., Agrawal, H., & Kotecha, K. (2020). Data augmentation using MG-GAN for improved cancer classification on gene expression data. Soft Computing - A Fusion of Foundations, Methodologies & Applications, 24(15), 11381–11391. <https://doi.org/10.1007/s00500-019-04602-2>
- Ke, X., & Du, P. (2020). Vehicle Logo Recognition with Small Sample Problem in Complex Scene Based on Data Augmentation. Mathematical Problems in Engineering, 1–10. <https://doi.org/10.1155/2020/6591873>
- <https://machinelearningmastery.com/how-to-configure-image-data-augmentation-when-training-deep-learning-neural-networks/>
- <https://www.doctormetrics.com/datos-aumentados/>
- Zhao, H., Hou, C., Tiwari, S., Trivedi, M., & Kohle, M. L. (2018). Recognition of motion state by smartphone sensors using Bi-LSTM neural network. Journal of Intelligent & Fuzzy Systems, 35(2), 1733–1742. <https://doi.org/10.3233/JIFS-169709>
- https://en.wikipedia.org/wiki/Long_short-term_memory
- <https://blog.gft.com/es/2018/11/06/como-usar-redes-neuronales-lstm-en-la-prediccion-de-averias-en-las-maquinas/>
- <https://es.slideshare.net/PlainConcepts/lstm-qu-son-y-cundo-se-usan>

- Milton, A. (2019). Inception V3 Deep Convolutional Architecture For Classifying Acute Myeloid/Lymphoblastic Leukemia. Extraído de: <https://software.intel.com/content/www/us/en/develop/articles/inception-v3-deep-convolutional-architecture-for-classifying-acute-myeloid-lymphoblastic.html>