



# Tecnológico de Monterrey

**Campus Querétaro**

Modelado #3

Gamaliel Marines Olvera	A01708746
Uri Jared Gopar Morales	A01709413
José Antonio Miranda Baños	A01611795
María Fernanda Moreno Gómez	A01708653
Oskar Adolfo Villa López	A01275287
Luis Ángel Cruz García	A01736345

Inteligencia artificial avanzada para la ciencia de datos II  
Grupo 501

---

## Introducción

En este documento se presentan las mejoras implementadas en el Modelo 3, una red neuronal convolucional (CNN) diseñada para clasificar imágenes en tres clases: “cama vacía,” “vaca acostada” y “vaca de pie.” Este modelo busca abordar los problemas identificados en el Modelo 2, específicamente la inconsistencia en la clasificación de la cama 3 y las imágenes tomadas durante la noche.

Estas dificultades fueron detectadas durante el análisis de datos del Modelo 2, donde los resultados de la clasificación no coincidían con la clasificación manual para el conjunto de entrenamiento. En respuesta, el Modelo 3 incluye ajustes en las transformaciones de imágenes y el entrenamiento para mejorar su capacidad de generalización y precisión en condiciones difíciles.

## Técnicas de modelado

La técnica de modelado utilizada es la misma descrita en el Modelo 1, que es la misma utilizada para el Modelo 2. A continuación, se presenta la técnica de modelado descrita en el Modelo 1 a modo de referencia.

Utilizamos una técnica de Red Neuronal Convolucional (CNN) debido a su eficacia para procesar datos en forma de cuadrícula, como las imágenes. La CNN está formada por varias capas: capas convolucionales, capas de agrupación y capas totalmente conectadas. Esta arquitectura se asemeja al procesamiento visual del cerebro humano y es adecuada para capturar patrones jerárquicos y dependencias espaciales dentro de imágenes.

Las capas que se utilizaron para esta arquitectura fueron las siguientes:

- **Capas convolucionales:** Aplican operaciones convolucionales a las imágenes. Sirven para detectar bordes, texturas y patrones complejos.
- **Capas de agrupación:** Reducen el tamaño de las dimensiones espaciales de la entrada con el fin de reducir la complejidad computacional.
- **Funciones de activación:** Utilizamos Rectified Linear Unit (ReLU) para introducir una propiedad no lineal al modelo con el fin de aprender patrones más complejos, evita el vanishing gradient<sup>1</sup> al no saturarse de entradas positivas, mitigando este problema y acelerando la convergencia durante el entrenamiento, además que es computacionalmente más eficiente que otras funciones al tener una forma matemática directa.

---

<sup>1</sup> **Vanising gradient:** Es un problema que ocurre en los entrenamientos de las CNN cuando los gradientes de la función de pérdida con respecto a los pesos de las primeras capas se vuelven extremadamente pequeños, resultando que las capas reciban poca o nula información de los pesos en la retropropagación (backpropagation).

- **Capas completamente conectadas:** Estas son las responsables de hacer las predicciones basadas en los aprendizajes de las capas anteriores.

Las capas son entrenadas por medio de carpetas con imágenes clasificadas de acuerdo a las etiquetas de “cama vacía”, “vaca acostada” y “vaca de pie”, por lo que la red aprende a reconocer patrones y características asociadas a cada clase.

Para mejorar la generalización del modelo, se implementó la técnica de aumento de datos (data augmentation), que es el proceso de generar artificialmente nuevos datos a partir de datos existentes para entrenar un modelo, ya que los modelos de Machine Learning requieren de conjuntos de datos reales diversos y vastos para la generalización del modelo. En nuestro modelo, se realizaron las siguientes aumentaciones: redimensionamiento, giro horizontal y vertical de la imagen y rotación, esto con el fin de aumentar artificialmente los datos y tener un conjunto de datos de entrenamiento más robusto para que pueda realizar mejores predicciones el modelo.

## Suposiciones

Las suposiciones continúan siendo las mismas que las descritas en el Modelo 1. A continuación, se presentan las suposiciones descritas en el Modelo 1 a modo de referencia.

- Las imágenes de las camas tienen el mismo tamaño, 950 x 450 px.
- Las imágenes muestran una sola cama.
- Las imágenes no muestran personas u objetos diferentes.
- Las imágenes de las camas se toman desde arriba.
- Las imágenes fueron tomadas por un modelo específico de cámara y fueron pasadas por un proceso de aplanado, que coincide con el ojo de pescado<sup>2</sup> generado por dicho tipo de cámara en específico.

## Diseño de pruebas

Para las pruebas se utilizarán los mismos elementos que se consideraron en el modelo 1 y 2, agregando dos métricas más que fueron de utilidad para identificar de manera cuantitativa problemas que notamos en la iteración anterior y así poder demostrar su posterior corrección.

---

<sup>2</sup> **Ojo de pescado:** Se refiere a las distancias focales que tienen ángulos muy anchos y que consiguen producir una distorsión visual de grandes capacidades.

## F1 score

Esta métrica representa la media armónica de precisión (accuracy) y exhaustividad (recall), balanceando la estabilidad de ambos. Es útil para verificar que la precisión en las diferentes clases es adecuada.

En donde:

$$\text{Precisión} = \frac{\text{Verdaderos Positivos}}{\text{Verdaderos Positivos} + \text{Falsos Positivos}}$$

$$\text{Exhaustividad} = \frac{\text{Verdaderos Positivos}}{\text{Verdaderos Positivos} + \text{Falsos Negativos}}$$

$$F1 = 2 * \frac{\text{Precisión} * \text{Exhaustividad}}{\text{Precisión} + \text{Exhaustividad}}$$

## Accuracy promedio por cama

Esta métrica nos ayuda a ver que tan bueno es el modelo para predecir en las diferentes camas, ya que en la iteración anterior, vimos que existía un problema para predecir en una cama en específico.

$$\text{Accuracy promedio por cama} = \frac{\text{total de coincidencias camas A,B y C}}{\text{total real de camas A,B y C}}$$

El diseño de pruebas que se utilizó en los Modelos 2 y 3 y, que continúa siendo útil para evaluar este modelo, se presenta a continuación a modo de referencia.

Para las pruebas se utilizará el conjunto de datos que fue seleccionado para dicho propósito desde un inicio, lo que brinda la posibilidad de ver el rendimiento del modelo en imágenes que no fueron utilizadas para su entrenamiento, logrando de esta forma identificar si existe un sobreajuste<sup>3</sup>.

Específicamente, se cuenta con la siguiente cantidad de imágenes por cada clase para el conjunto de prueba:

- Cama vacía: 1323
- Vaca acostada: 394
- Vaca de pie: 60

Una vez utilizado el modelo para evaluar los nuevos datos, se obtendrá como métrica principal el accuracy en forma de porcentaje, el cual se obtiene implementando la siguiente fórmula:

---

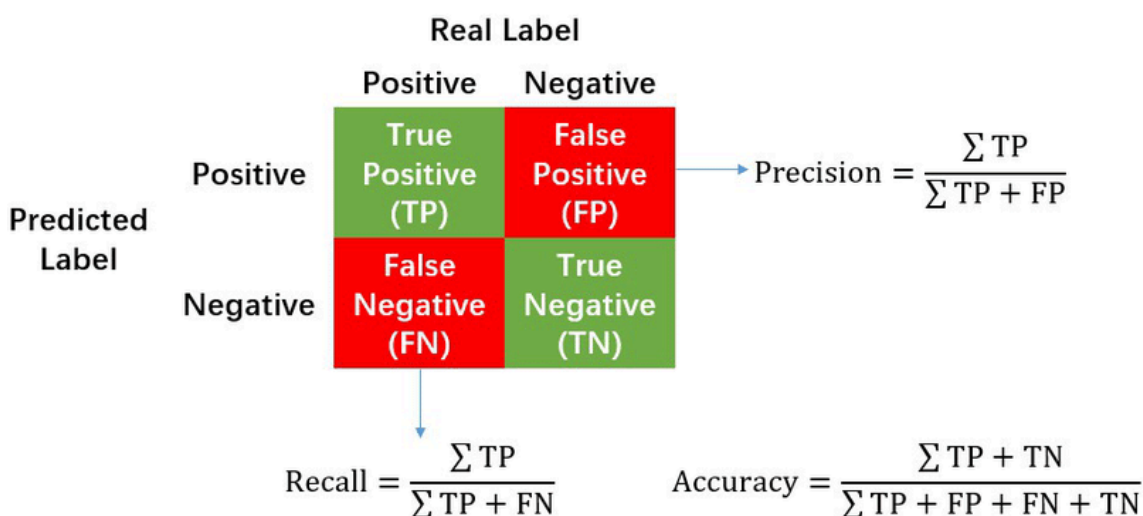
<sup>3</sup> **Sobreajuste (overfitting):** Es un comportamiento en Machine Learning (ML) que se produce cuando el modelo da predicciones precisas para los datos de entrenamiento, pero no para nuevos datos, es decir, aprende de memoria los resultados, sin embargo, no aprende los patrones en realidad para datos no vistos.

$$Accuracy = \frac{Correct\ predictions}{Total\ predictions} * 100$$

Buscando obtener un valor que no solo muestre que el ajuste del modelo es correcto, sino también que no existe una diferencia significativa entre dicho accuracy y el obtenido durante el entrenamiento.

Por último, se utilizará una matriz de confusión para poder identificar si existe tendencia a predecir una clase en específico, el cual es otro problema que se busca evitar, ya que al contar con pocas imágenes de “Vaca de pie”, podría no estar detectando correctamente dicha clase y seguir mostrando un accuracy alto.

Los valores que se muestran en la matriz de confusión son acorde a las clases, en donde se comparará la cantidad de valores predichos correctamente para cada clase. A continuación se muestra un diagrama que ejemplifica los valores obtenidos en una matriz de confusión, en dicho diagrama, los valores que se deberían mostrar más altos son los verdaderos positivos (TP) y los verdaderos negativos (TN), ya que significa que el número de predicciones por cada clase fueron correctas.



**Imagen 1:** Diagrama de construcción de una matriz de confusión.

## Modelo

### Diseño y configuración del tercer modelo

El Modelo 3 se basa en los principios fundamentales de los Modelos 1 y 2, manteniendo las técnicas de preprocesamiento y arquitectura básica, pero

introduciendo mejoras significativas en transformaciones de imágenes, entrenamiento y evaluación.

### Transformaciones de imágenes

Para aumentar la robustez del modelo frente a variaciones en las condiciones de captura, como iluminación y ángulos de cámara, se añadieron nuevas transformaciones al pipeline de preprocesamiento:

- **Transformaciones heredadas:**
  - Redimensionamiento a (950, 450).
  - Giros horizontales y verticales aleatorios con probabilidad del 50%.
  - Rotaciones aleatorias de hasta 30 grados.
  - Zoom out mediante *RandomAffine* con escalado entre 50% y 100%.
- **Transformación nueva Saturación:**
  - Se añadió *ColorJitter* con un factor de saturación de 2.
  - El propósito de añadir esta transformación es para simular variaciones de iluminación y color, especialmente para mejorar la clasificación de las camas por la noche, de modo que los tonos se unifiquen y el modelo busque identificar mejor la forma.

Estas transformaciones buscan incrementar la diversidad del conjunto de datos y mejorar la capacidad del modelo para generalizar en condiciones reales.

### Entrenamiento

- **Pesos por Clase:** En la función de pérdida *CrossEntropyLoss*, se introdujeron pesos específicos para combatir el desbalance de los datos:
  - Clase "cama vacía": peso 3.0.
  - Clase "vaca acostada": peso 1.0.
  - Clase "vaca de pie": peso 9.0.
- **Tasa de Aprendizaje:** Se redujo a 0.0001 para garantizar ajustes finos en los pesos, permitiendo una convergencia más estable.

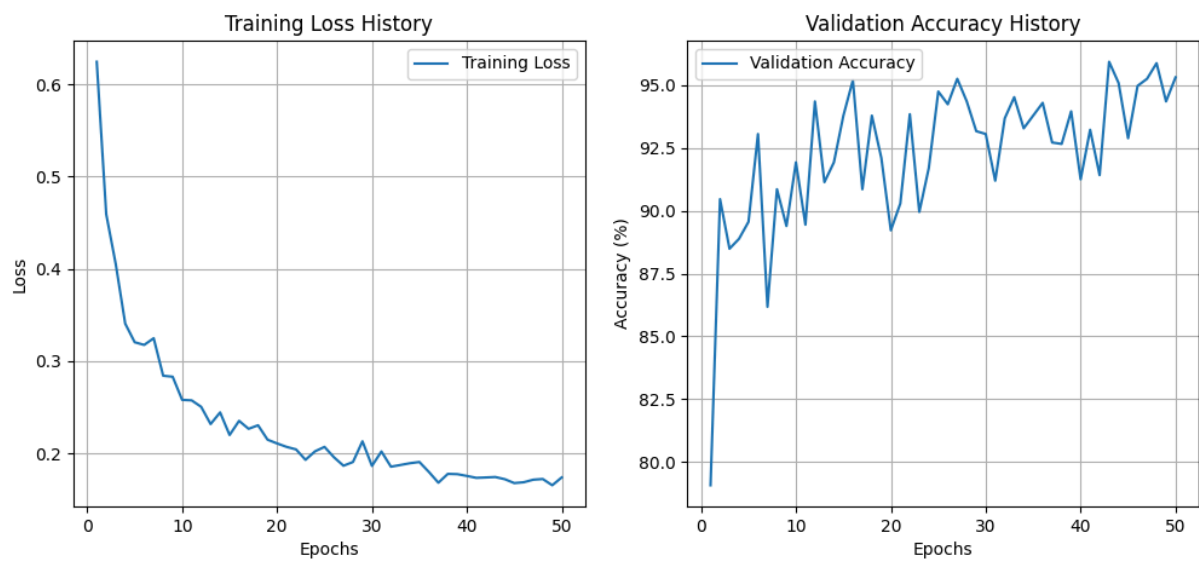
### Métricas y pruebas en el conjunto de Test (Prueba)

Se utilizó lo mismo definido en los Modelos 1 y 2. A continuación, se adjunta a modo de referencia.

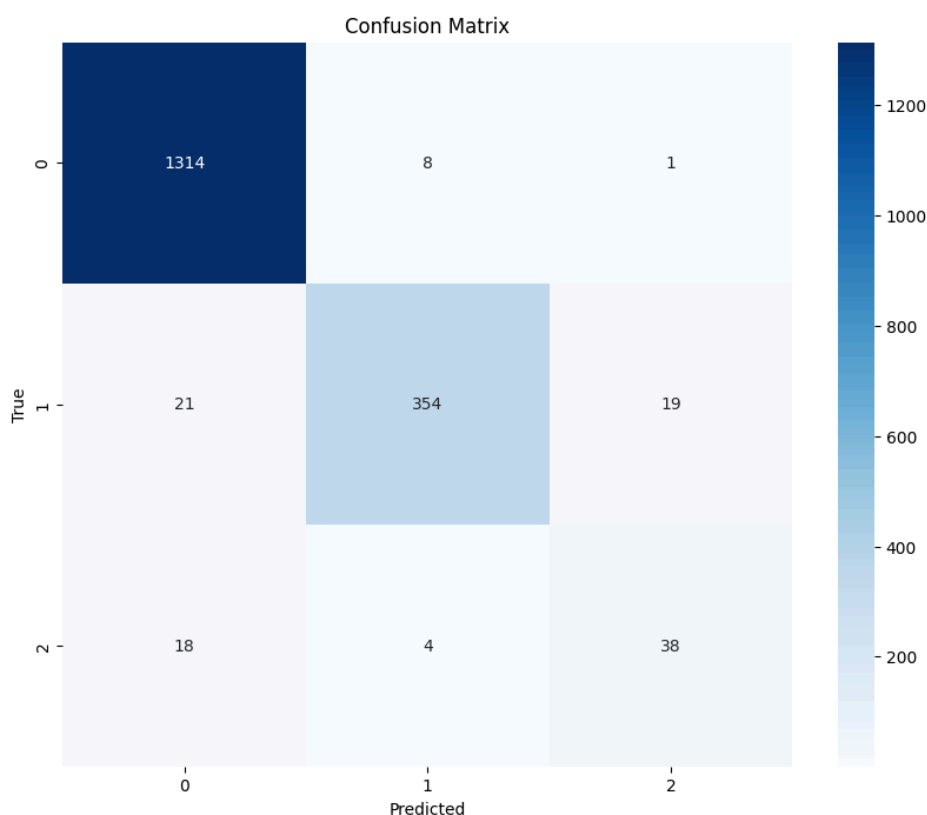
La evaluación final se realiza en el conjunto de prueba, obteniendo el accuracy como métrica principal para medir el rendimiento del modelo en datos no utilizados previamente en el entrenamiento. Adicionalmente, se genera una matriz de confusión para observar si el modelo tiende a preferir alguna clase específica,

particularmente ante la escasez de imágenes de “vaca de pie,” y así detectar problemas de sesgo en la predicción.

### Evaluación del tercer modelo



**Imagen 2:** Gráfico de pérdida de training y de precisión en el conjunto de validación del tercer modelo.



**Imagen 3:** Matriz de confusión del tercer modelo. (0: Cama vacía, 1: Vaca acostada, 2: Vaca de pie)

El modelo 3 fue diseñado para corregir las limitaciones observadas en el Modelo 2, especialmente la inconsistencia en la clasificación de la clase “vaca de pie” y las imágenes capturadas de noche. Tras implementar mejoras adicionales en el preprocesamiento y el entrenamiento, los resultados indican un progreso significativo.

En la gráfica de pérdida (loss) podemos observar que:

- Durante las 50 épocas de entrenamiento, se observa una disminución constante en la pérdida de entrenamiento, indicando que el modelo logró optimizar sus pesos de manera efectiva, minimizando los errores.
- La estabilidad de la curva hacia el final del entrenamiento sugiere que el modelo alcanzó una convergencia adecuada.
- Hay pequeñas fluctuaciones muy mínimas durante las épocas 7 en adelante aproximadamente, sin embargo, sí llega a alcanzar una aparente convergencia.

En la gráfica de precisión (accuracy) podemos observar que:



- El accuracy en validación muestra una mejora continua a lo largo de las épocas, superando el 90% desde la época 11.
- Se alcanzó un accuracy máximo de aproximadamente 95.7%, oscilando mayormente entre el 93% durante las épocas, lo que nos dice que el modelo tiene una generalización adecuada.
- Las oscilaciones menores en el accuracy durante las primeras épocas se estabilizan con el avance del entrenamiento.

En la gráfica de la matriz de confusión del modelo 3 podemos observar que:

- **Clase 0 (Cama vacía):**
  - Tuvo una precisión alta, con 1314 imágenes clasificadas de manera correcta y solo 9 errores en total, 8 de estos siendo clasificados como vaca acostada.
  - Por esta razón, el modelo tiene bastante robustez para la clasificación de las camas vacías.
- **Clase 1 (Vaca acostada):**
  - Tuvo una menor precisión, pues anteriormente lograba clasificar correctamente 390, equivocándose únicamente en 4 imágenes y ahora logró clasificar 354 imágenes correctamente equivocándose en 40. Los errores se distribuyen de manera casi equitativa con 21 para cama vacía y 19 para vaca de pie.
  - El que haya reducido su precisión en la clasificación no significa que este modelo sea peor, pues anteriormente parecía que clasificaba de manera correcta, pero no era así al observar las imágenes clasificadas.
- **Clase 2 (Vaca de pie):**
  - Tuvo una mejor precisión, pues anteriormente lograba clasificar correctamente 32 imágenes, equivocándose en 28 y ahora logró clasificar 38 imágenes correctamente equivocándose en 22. Los errores son mayormente de cama vacía, pues tuvo 18 errores en esta clase, mientras que en vaca acostada se equivocó 4 veces, es decir que el modelo clasificó 4 veces una imagen de tipo “vaca de pie” en la clase de “vaca acostada”.
  - De igual manera, esto no quiere decir que esté mal, debido a las clasificaciones mal hechas anteriormente en el modelo 2. Para ello, más adelante, en el documento se presentará el análisis de datos que es con el que nos basamos para la elección del modelo a utilizar en producción.

**Tabla comparativa entre los 3 modelos**

Aspecto	Modelo 1	Modelo 2	Modelo 3
Transformaciones en imágenes	Redimensionamiento, giros horizontales y verticales, rotación aleatoria.	Transformaciones del Modelo 1 + Zoom out	Transformaciones del Modelo 2 + Saturación (ColorJitter, factor de saturación = 2).
Épocas de entrenamiento	5 épocas	50 épocas	50 épocas
Pesos en función de pérdida	No se utilizaron pesos específicos para las clases.	Pesos ajustados: 4.0 para "vaca acostada" y 8.0 para "vaca de pie" debido al desbalance de clases.	Pesos ajustados: 3.0 para "cama vacía", 1.0 para "vaca acostada" y 9.0 para "vaca de pie".
Learning rate	0.001	0.001	0.0001
Desempeño en la clase "cama vacía"	True Positives: 1300 False Positives: 44 False Negatives: 23	True Positives: 1306 False Positives: 34 False Negatives: 17	True Positives: 1312 False Positives: 40 False Negatives: 11
Desempeño en la clase "vaca acostada"	True Positives: 375 False Positives: 52 False Negatives: 19	True Positives: 371 False Positives: 16 False Negatives: 23	True Positives: 356 False Positives: 13 False Negatives: 38
Desempeño en la clase "vaca de pie"	True Positives: 5 False Positives: 1 False Negatives: 55	True Positives: 30 False Positives: 20 False Negatives: 30	True Positives: 31 False Positives: 25 False Negatives: 29
Matriz de confusión	Mayor precisión en clases mayoritarias ("cama vacía"), pero menor en clases minoritarias.	Mejor distribución, aunque "vaca de pie" sigue siendo la clase más difícil de clasificar.	Distribución más robusta, aunque las clases minoritarias aún enfrentan dificultades.
F1-score (Test)	94%	98%	96%
Áreas de mejora	Manejo del desbalance de datos, aumento de transformaciones y mayor número de épocas de entrenamiento.	Mejorar la detección de la clase "vaca de pie".	Incrementar el dataset para clases minoritarias y mejorar aún más las estrategias de balance.

Accuracy en Test	95%	96%	96%
Accuracy promedio por cama	59.60%	80%	95%

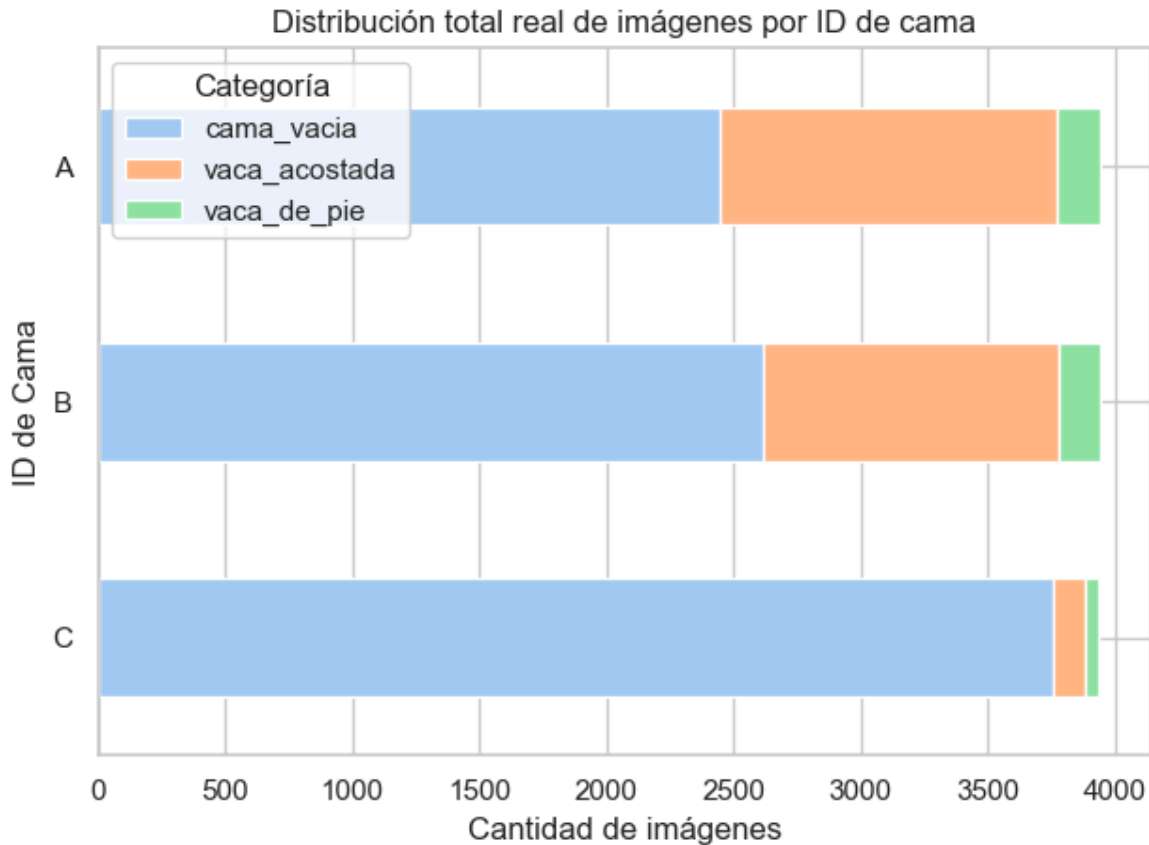
A lo largo de las tres iteraciones de modelado, se han realizado mejoras significativas en el diseño y entrenamiento del modelo, lo que ha permitido abordar los desafíos detectados en cada versión previa. El Modelo 1 presentó una arquitectura básica, pero mostró limitaciones significativas al clasificar clases minoritarias como "vaca acostada" y "vaca de pie." Con el Modelo 2, se introdujeron transformaciones avanzadas como el zoom out y ajustes en los pesos de la función de pérdida, logrando una mejora en la generalización y un accuracy de validation del 96.85%. Sin embargo, persisten problemas de clasificación en la clase "vaca de pie" debido al desbalance de datos.

El Modelo 3 introdujo nuevas transformaciones, como la saturación, y un ajuste más fino en los pesos de pérdida, priorizando la corrección de inconsistencias detectadas en el Modelo 2. Aunque el accuracy en validation fue ligeramente menor, el accuracy promedio por cama aumentó considerablemente, haciendo el modelo más apto para su uso en el despliegue. A pesar de los avances, la clase "vaca de pie" sigue siendo la más desafiante, destacando la necesidad de ampliar el dataset y en casos futuros, se pudiera explorar arquitecturas más complejas para continuar optimizando los resultados.

### Validación de Modelo

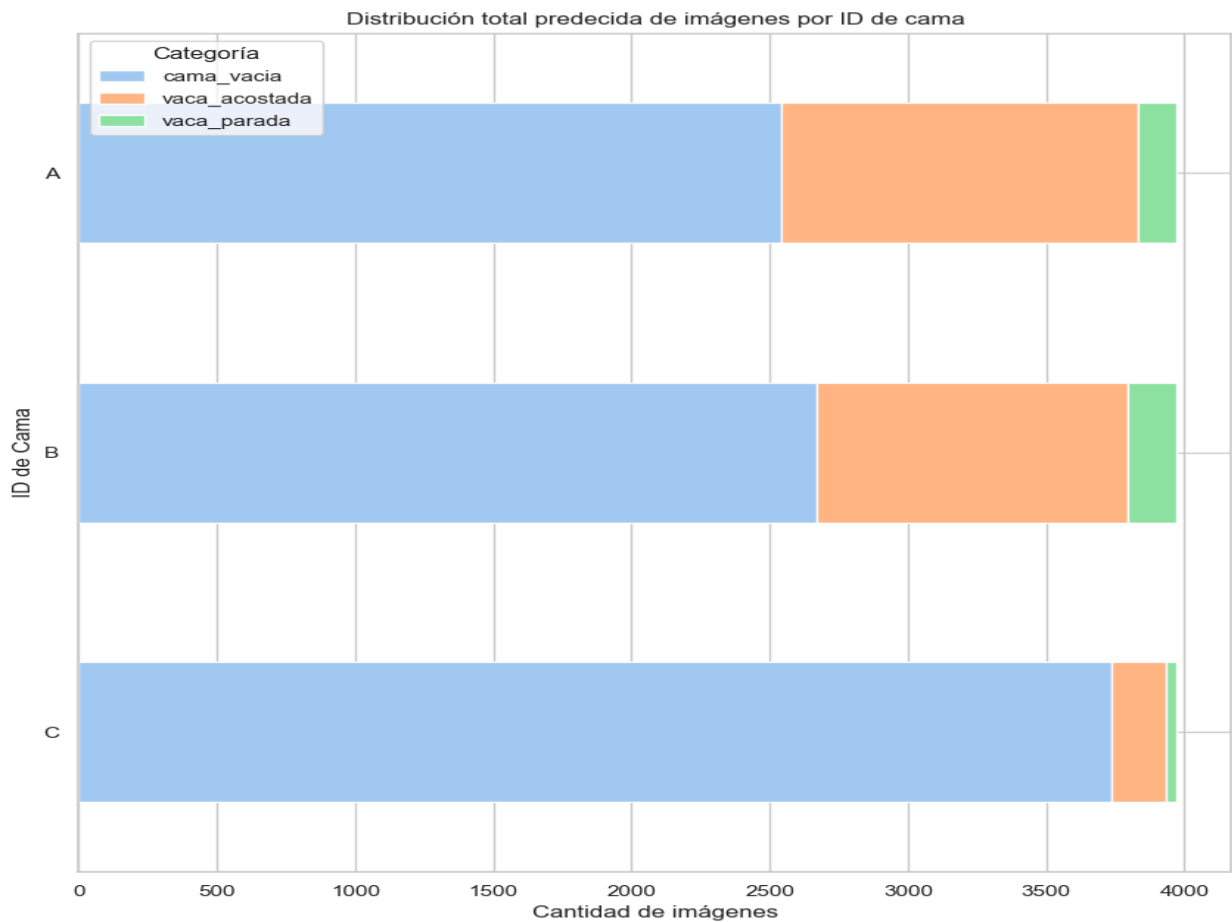
El primer paso del análisis de la fase del Modelo 2 se repitió para verificar el funcionamiento del código de análisis y del modelo. En esta ocasión, las predicciones para el conjunto de entrenamiento son muy similares a los valores reales de la clasificación manual, por lo que se concluyó que el modelo realiza predicciones correctas, y se puede continuar con el análisis.

## Análisis de datos



**Imagen 4:** Distribución de las imágenes de cada cama de acuerdo a las categorías. Dataset de entrenamiento. Clasificación manual.

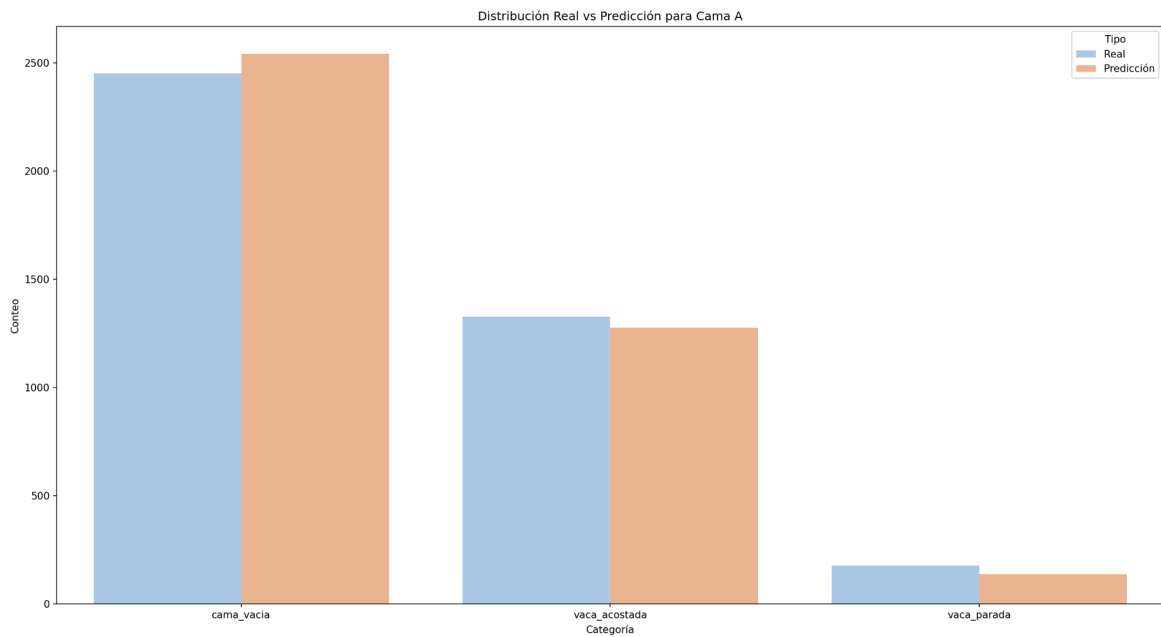
En la imagen 4 se puede observar que la cama C cuenta con un número significativamente mayor de camas vacías, mientras que en la cama A y B los valores se mantienen similares.



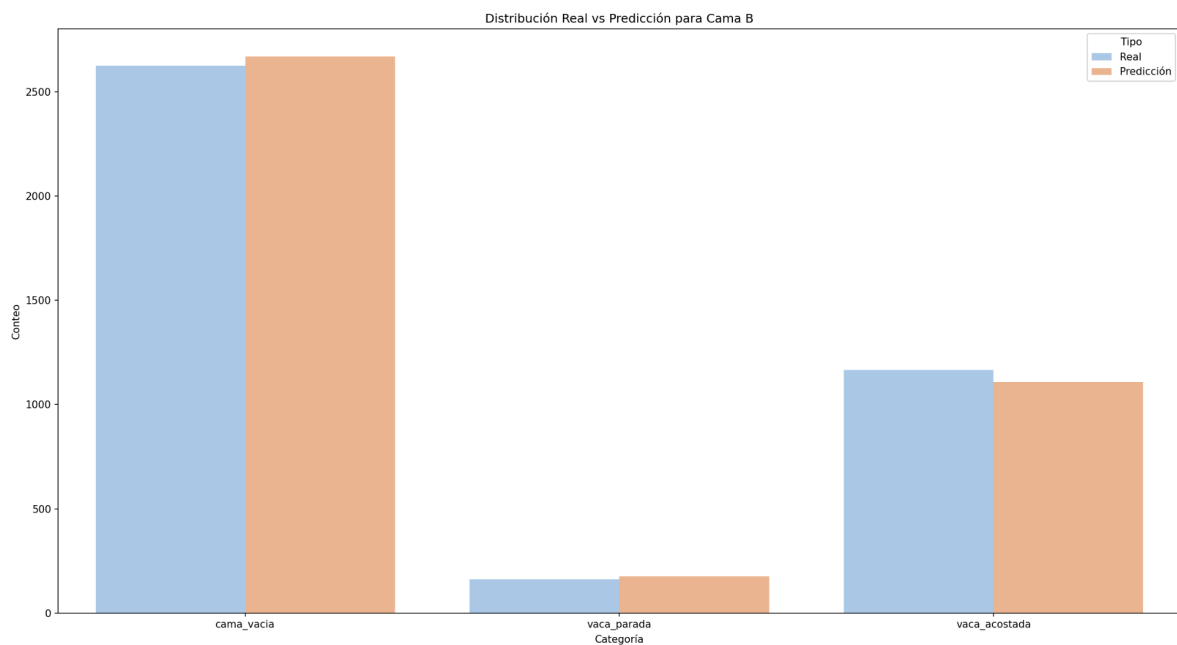
**Imagen 5:** Distribución de las imágenes de cada cama de acuerdo a las categorías. Dataset de entrenamiento. Predicción del modelo.

Como se puede observar en la imagen 5, las predicciones concuerdan en gran medida con los valores reales observados en la imagen 4. Reflejando que la cama C es la menos utilizada.

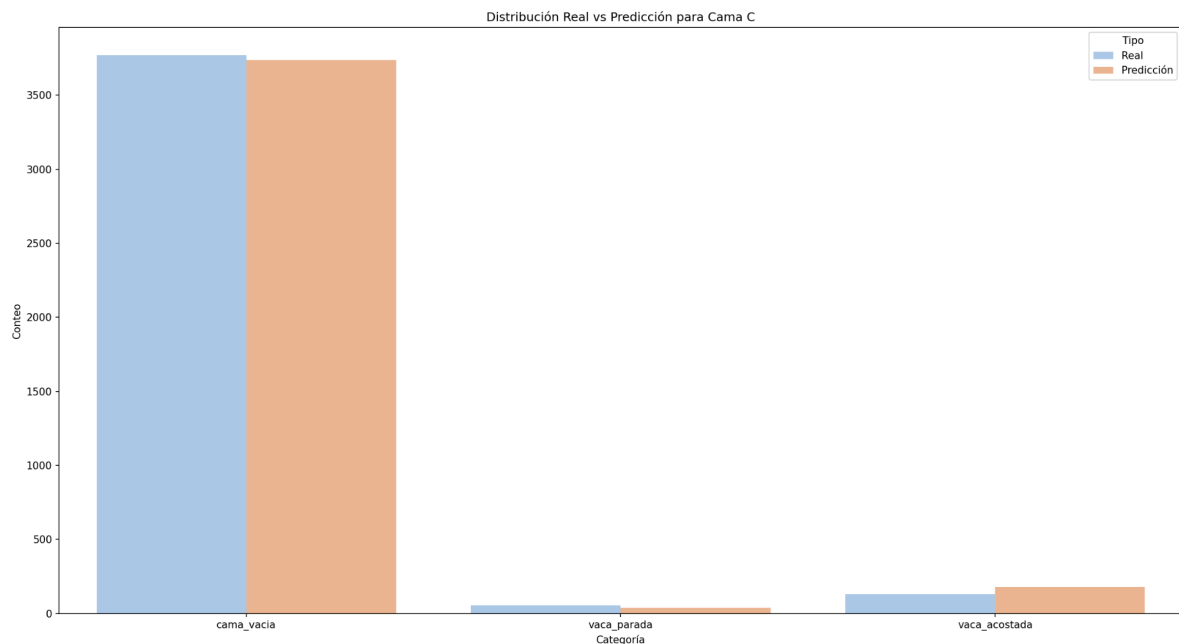
A continuación se muestran las gráficas obtenidas al comparar predicciones contra valores reales y el accuracy obtenido por cada cama.



**Imagen 5.1:** Comparación de predicciones y valores reales en la cama A refleja un 94.23% de accuracy.



**Imagen 5.2:** Comparación de predicciones y valores reales en la cama B refleja un 95.09% de accuracy.

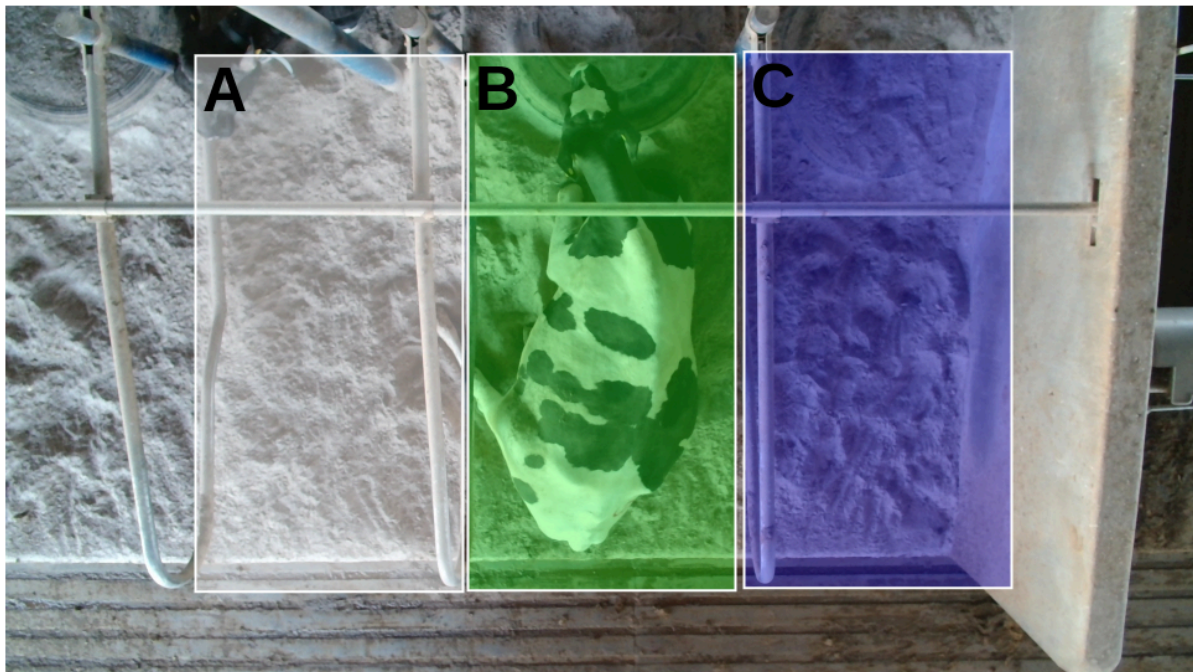


**Imagen 5.3:** Comparación de predicciones y valores reales en la cama C refleja un 98.55% de accuracy.

Esto en promedio nos resulta en un 95% de accuracy entre las 3 camas. Esta métrica nos permite escoger un modelo que se comporte mejor para distintos casos y evitar algunos sesgos que no son tan identificables por el accuracy tradicional y nos asegura que el rendimiento del modelo es similar entre las camas.

### Hallazgos y análisis de datos

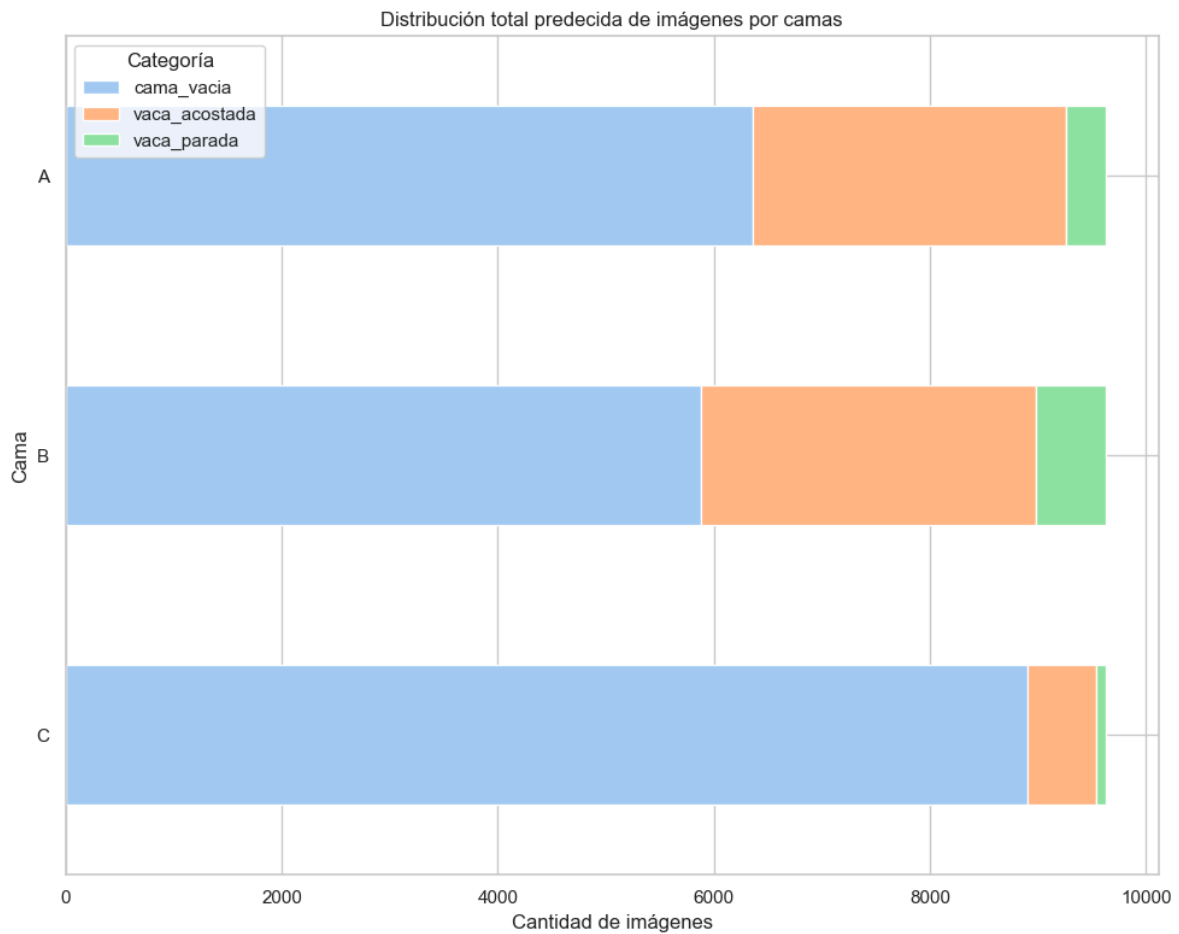
Se realizó un análisis del conjunto de datos completo. Las camas se dividieron usando la siguiente selección:



**Imagen 6:** Selección de las camas para el análisis.

Se descubrió la distribución de las camas individuales. Se puede observar que la distribución de las camas A y B es muy similar. Ambas están vacías la mayoría del tiempo. Sin embargo, la cama C está casi siempre vacía. Esto puede deberse a que la pared en un costado reduce el espacio.

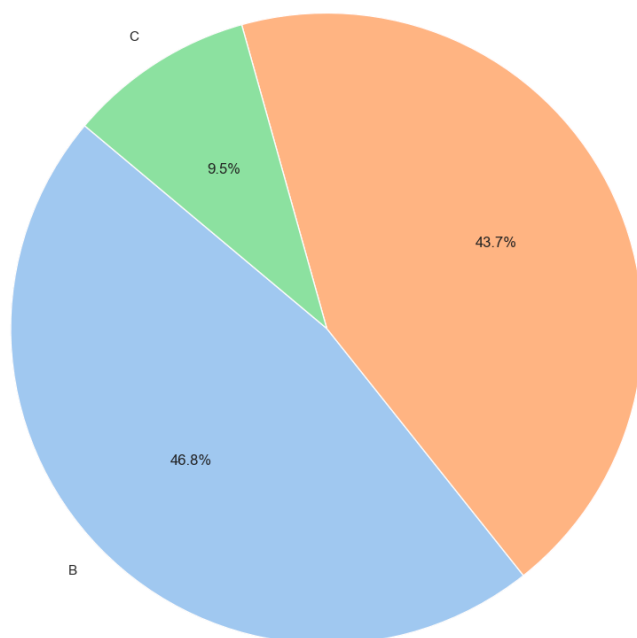




**Imagen 7:** Distribución de las imágenes de cada cama de acuerdo a las categorías. Dataset completo. Predicción del modelo.

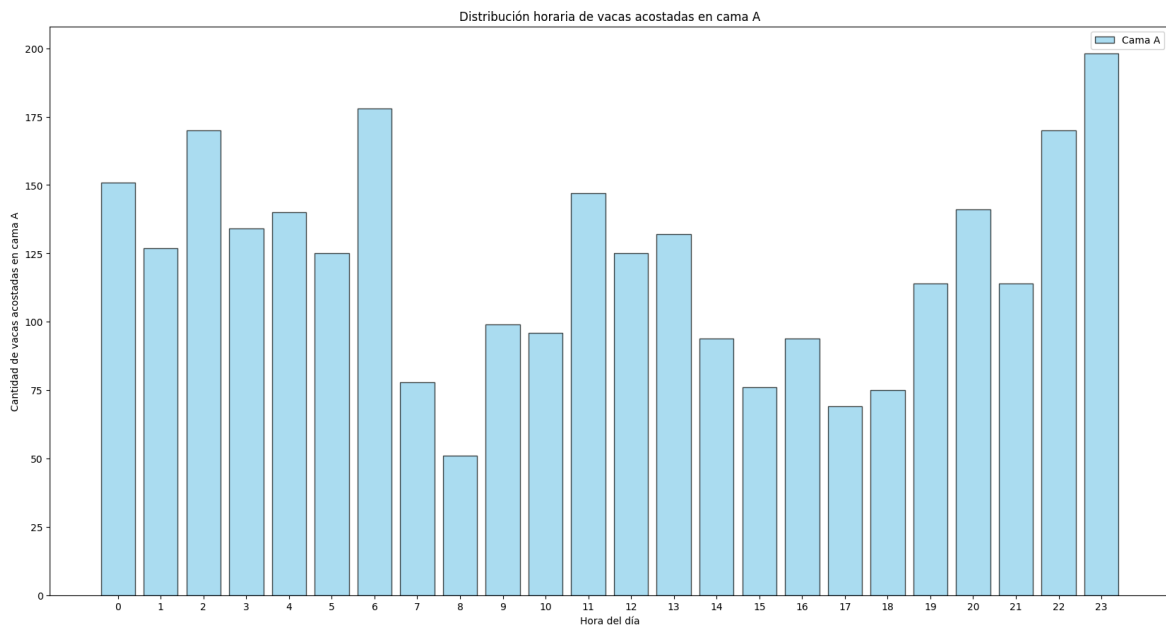
Posteriormente, se analizó cuál era la cama con más uso. De todas las imágenes etiquetadas como vaca acostada por el modelo, se graficó la distribución por cama en porcentaje. Aquí confirmamos que la cama C es la menos usada, solo en un 9.5 % de las veces. Las camas A y B no tienen una diferencia significativa en su uso.

Distribución de vacas acostadas por cama



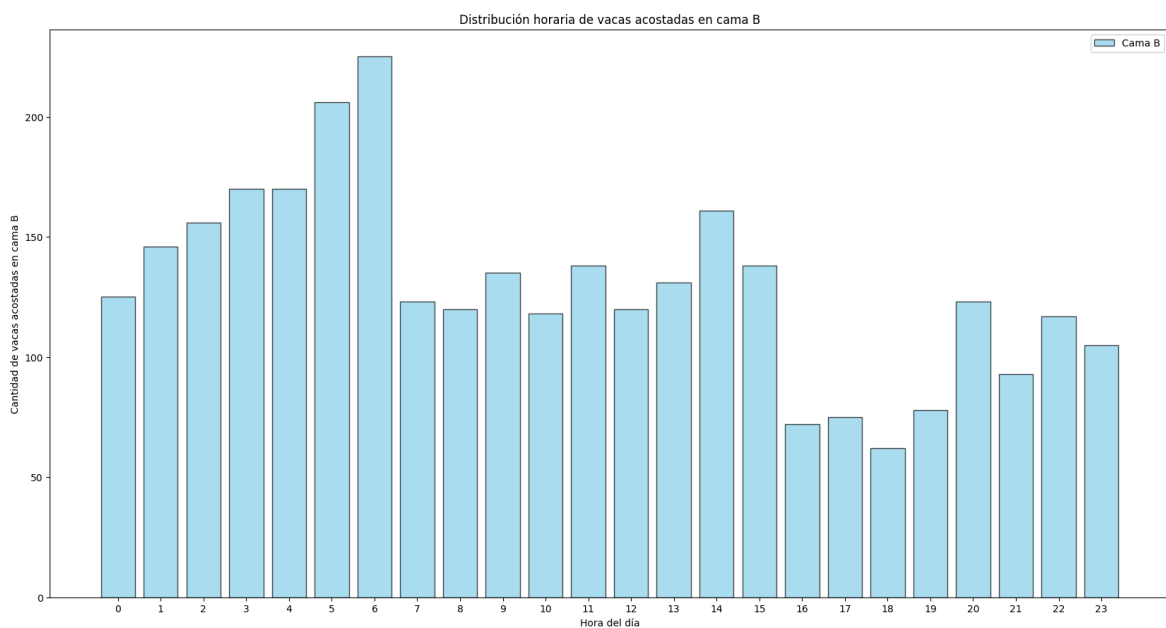
**Imagen 8:** Distribución de las imágenes de cada cama de acuerdo a las categorías. Clasificación manual.

Con la información obtenida, también es posible encontrar un histograma de las vacas acostadas en cada cama distribuido en cada hora del día, así como un histograma de las vacas acostadas en todas las camas combinadas. Esta información es útil para saber en qué momentos del día prefieren acostarse las vacas.



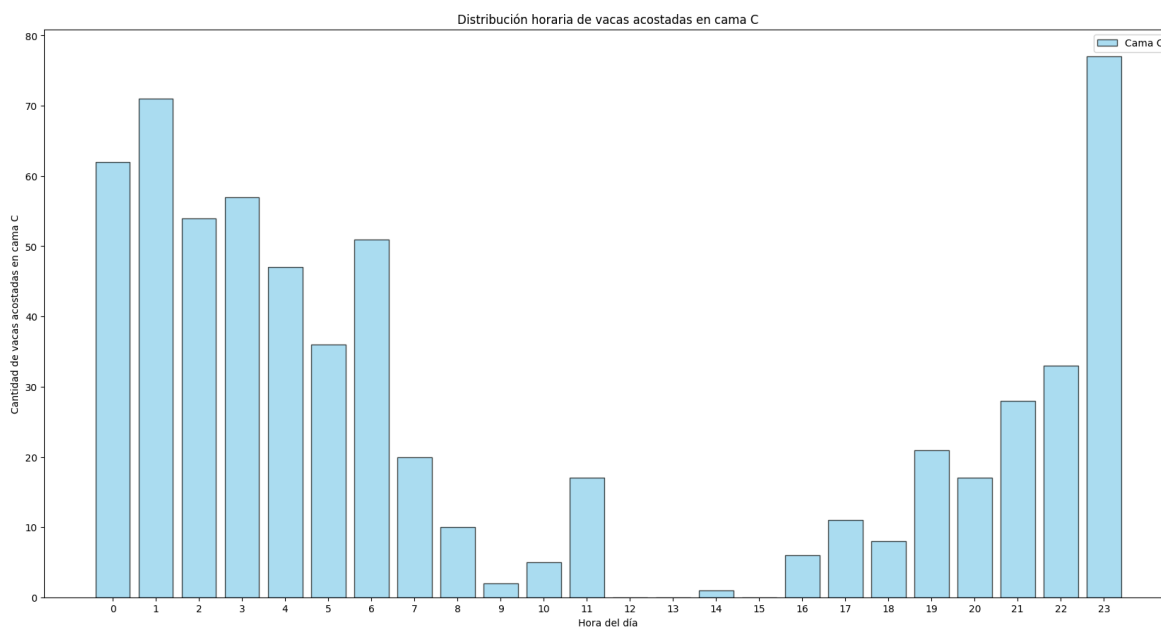
**Imagen 9:** Observación de horas en las que las vacas ocupan más la cama A (23hrs) .

En la imagen 9 se observa que la cama A tiene un mayor número de vacas acostadas a las 23:00 horas, además de presentar una mayor ocupación durante la noche y la madrugada.



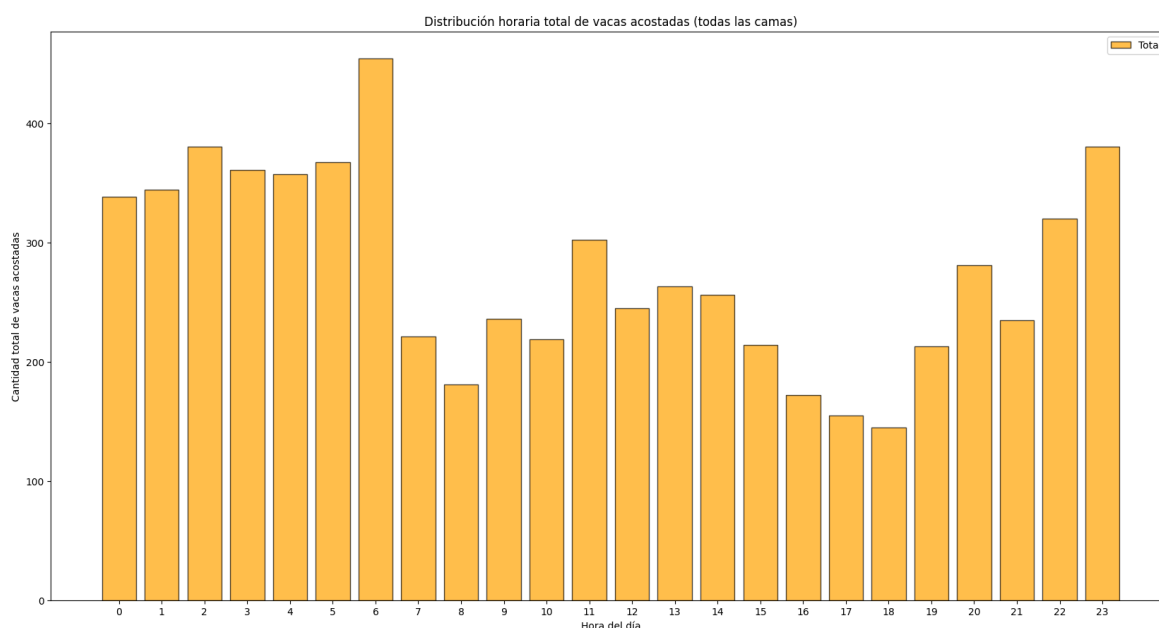
**Imagen 10:** Observación de horas en las que las vacas ocupan más la cama B (6hrs).

En la imagen 10 se observa que la cama B tiene un mayor número de vacas acostadas a las 6:00 horas, además de presentar una mayor ocupación durante la madrugada.



**Imagen 11:** Observación de horas en las que las vacas ocupan más la cama C (23hrs).

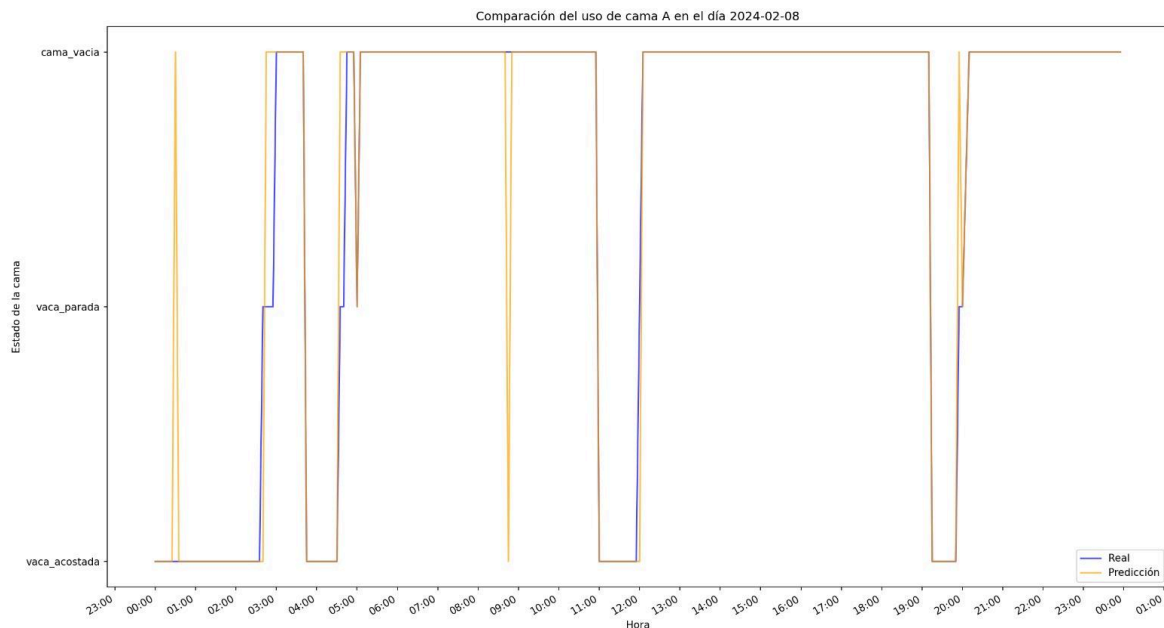
Por último, en la imagen 11 se observa que la cama C casi no presenta vacas acostadas durante el día, sin embargo, la mayor ocupación es a las 23:00 horas.



**Imagen 12:** Observación de horas en las que las vacas ocupan más las camas .

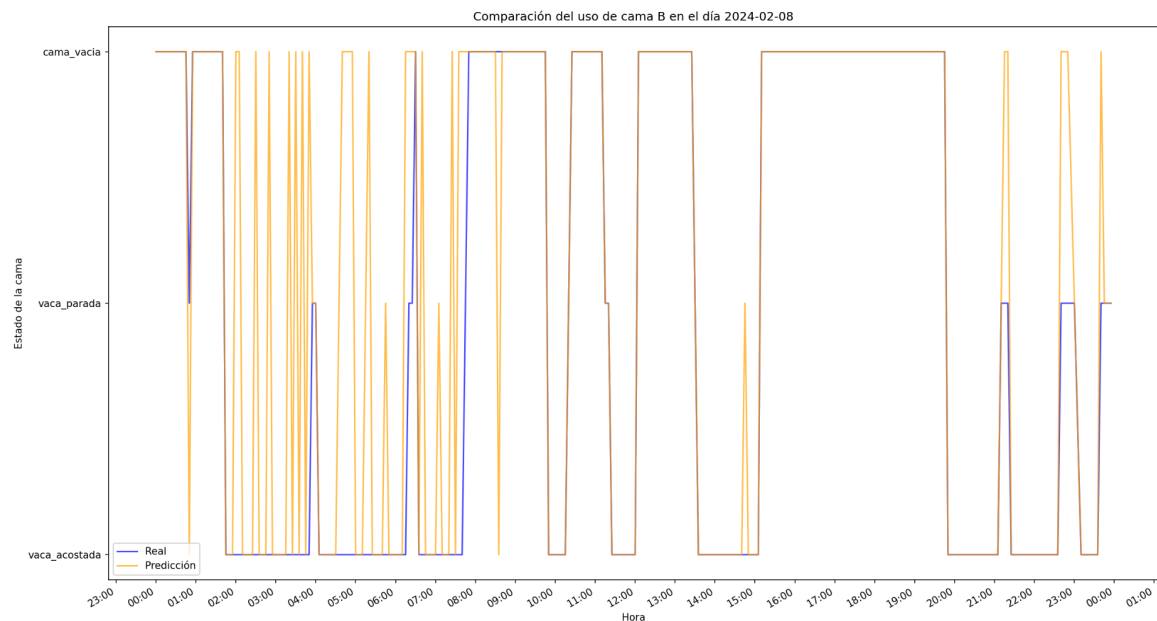
La **imagen 12** es la distribución general, nos indica que en general las camas son ocupadas en la madrugada y en la noche, mientras que presentan un uso menor de las 7 de mañana hasta las 6 de la tarde.

Finalmente, se analizó el día 8 de febrero de 2024 como una muestra, y se graficó el uso de cada cama a lo largo del día. La cama A se utilizó durante las primeras horas del día, y luego se ocupó en periodos cortos. La cama B se utilizó en periodos aún más cortos, pero tuvo actividad durante todo el día, exceptuando el periodo entre las 3 y las 8 pm. La cama C tuvo casi nula actividad en el día, con una vaca acostada a las 11 pm. El análisis de un solo día no es representativo del comportamiento general, pero puede ser útil para analizar comportamientos en momentos específicos.



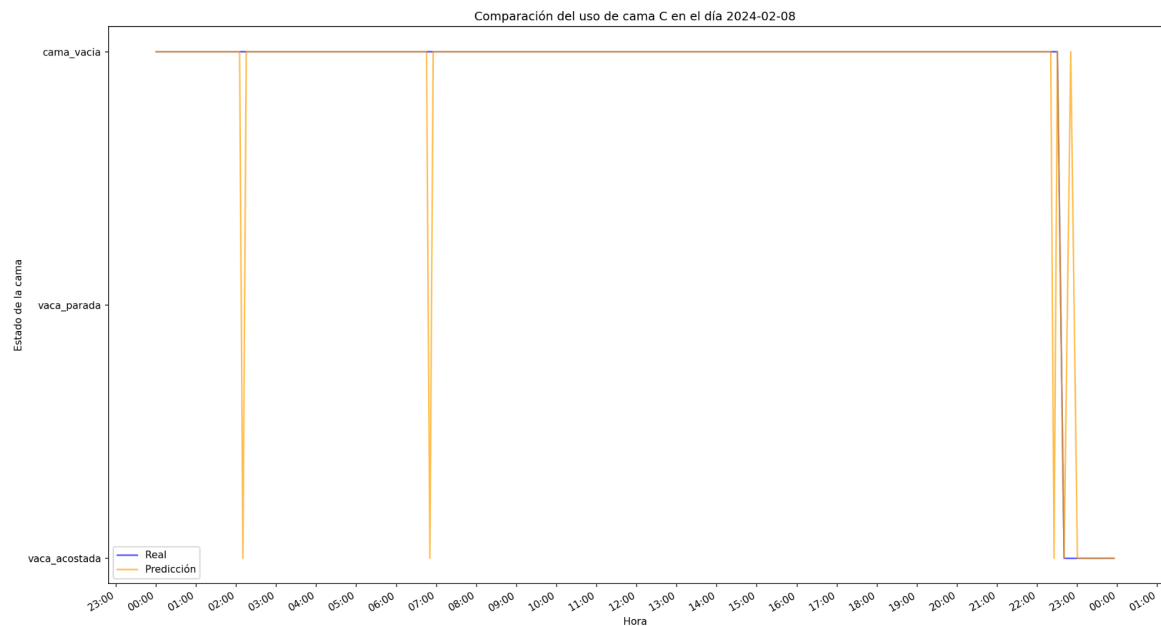
**Imagen 12:** Estado de la cama “A” a lo largo del día 2024-02-08 predicción vs real.

La gráfica nos muestra el uso de la cama A durante el día, donde la línea amarilla serán los valores predichos, la línea azul son los valores reales y la coincidencia es marcada con una línea café. Nos damos cuenta de que la cama A se ocupa principalmente en la mañana y a mediodía.



**Imagen 13:** Estado de la cama B a lo largo del día 2024-02-08 predicción vs real.

La **imagen 13** nos muestra el uso de la cama B durante el día, donde la línea amarilla serán los valores predichos, la línea azul son los valores reales y la coincidencia es marcada con una línea café. Nos damos cuenta de que la cama B se ocupa generalmente todo el día hasta las 3 de la tarde, donde se desocupa hasta las 8 de la tarde.

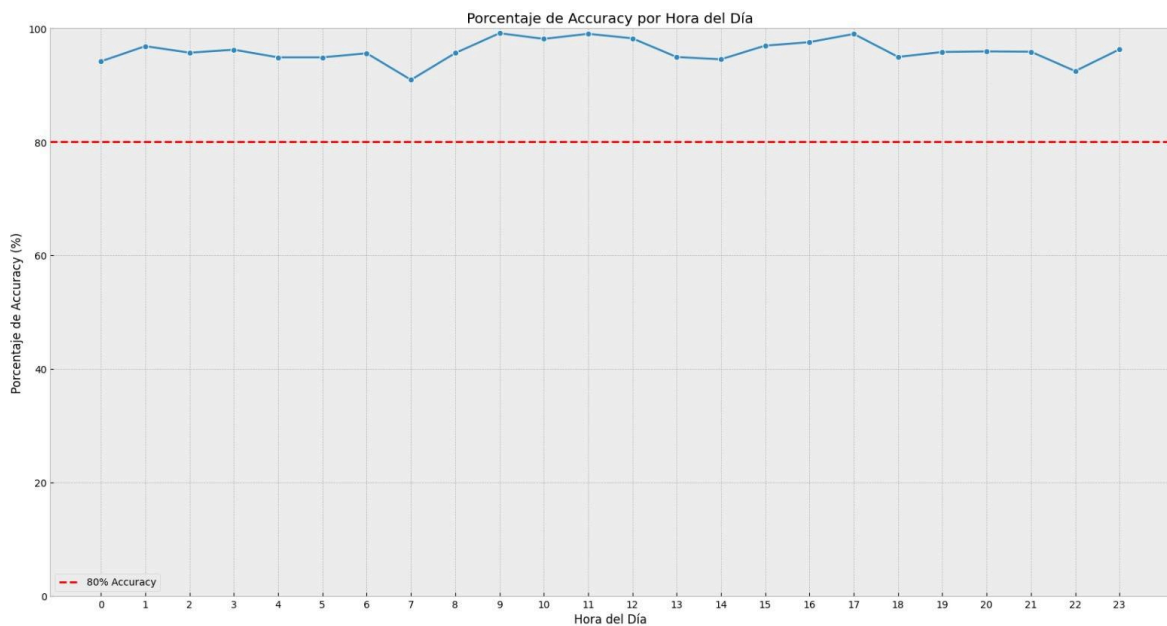


**Imagen 14:** Estado de la cama C a lo largo del día 2024-02-08 predicción vs real.

La **imagen 14** nos muestra el uso de la cama C durante el día, donde la línea amarilla serán los valores predichos, la línea azul son los valores reales y la coincidencia es marcada con una línea café. Nos damos cuenta de que la cama C se ocupa muy poco estando desocupada la mayor parte del día. Esto nos indica que la cama molesta a las vacas.

Estas gráficas de tiempo pueden ser útiles para identificar patrones de comportamiento del grupo de vacas, sin embargo, sería recomendable aplicar un filtro para reducir la frecuencia de los errores y capturar valores más acercados a la realidad.





**Imagen 15:** Accuracy promedio a lo largo del día.

Por último para confirmar que nuestro modelo es funcional durante el 80% del día (19.2 h) graficamos su desempeño promedio a lo largo del día, aunque el desempeño baja alrededor de las 7 de la mañana en general se mantiene arriba del 90% lo cual cumple con nuestro criterio de superar el 80% a lo largo del 80% del día.

## Referencias a los documentos de los otros modelos

- Modelo 1:

[\[04\]-Modeling\\_1\\_TC](#)

- Modelo 2:

[\[04\]-Modeling\\_2\\_TC](#)

## Referencias

GeeksforGeeks. (n.d.). *Convolutional neural network (CNN) in machine learning*. Recuperado de <https://www.geeksforgeeks.org/convolutional-neural-network-cnn-in-machine-learning/>

Srivastava, S. (2023). *Understanding the difference between ReLU and Sigmoid activation functions in deep learning*. Medium. Recuperado de <https://medium.com/@srivastavashivansh8922/understanding-the-difference-between-relu-and-sigmoid-activation-functions-in-deep-learning-33b280fc2071>

Amazon Web Services (AWS). (n.d.). *What is data augmentation?* Recuperado de [https://aws.amazon.com/what-is/data-augmentation/#:~:Data%20augmentation%20is%20the%20process,machine%20learning%20\(ML\)%20models](https://aws.amazon.com/what-is/data-augmentation/#:~:Data%20augmentation%20is%20the%20process,machine%20learning%20(ML)%20models)

Amanatulla, M. (2023). *Vanishing Gradient Problem in Deep Learning: Understanding, Intuition, and Solutions*. Medium. Recuperado de <https://medium.com/@amanatulla1606/vanishing-gradient-problem-in-deep-learning-understanding-intuition-and-solutions-da90ef4ecb54>

Bargainfotos. (n.d.). *Todo sobre los objetivos de ojo de pez*. Recuperado de <https://bargainfotos.com/blog/todo-sobre-los-objetivos-de-ojo-de-pez/>

Amazon Web Services (AWS). (n.d.). *What Is Overfitting?* Recuperado de <https://aws.amazon.com/es/what-is/overfitting/>

Investopedia. (n.d.). *What Is Nonlinearity?* Recuperado de <https://www.investopedia.com/terms/n/nonlinearity.asp>

Sue, N. (2023). *What Is the Softmax Function Used in Deep Learning? Illustrated in an Easy-to-Understand Way*. Medium. Recuperado de

---

[https://medium.com/@sue\\_nlp/what-is-the-softmax-function-used-in-deep-learning-illustrated-in-an-easy-to-understand-way-8b937fe13d49](https://medium.com/@sue_nlp/what-is-the-softmax-function-used-in-deep-learning-illustrated-in-an-easy-to-understand-way-8b937fe13d49)

Analytics Vidhya. (2023). *What Is Adam Optimizer?* Recuperado de <https://www.analyticsvidhya.com/blog/2023/09/what-is-adam-optimizer/>