



# Modelo de machine learning en el proceso de atención de pacientes en emergencias hospitalarias

Machine learning model in the patient care process in hospital emergencies

 Violeta Morán Huamani  
2021008019@unfv.edu.pe  
Universidad Nacional Federico Villarreal, Perú

## Resumen

El presente estudio se enfocó en el desarrollo e implementación de un algoritmo de aprendizaje automático destinado a optimizar el flujo de atención en unidades de urgencias, priorizando específicamente la etapa de clasificación o triaje. Para validar el rendimiento de la herramienta, se aplicaron análisis estadísticos como la prueba t de Student y ANOVA, los cuales arrojaron valores significativos ( $t = 7.823$ ;  $p < 0.001$ ). Dichos hallazgos confirman que el modelo es capaz de reducir las demoras en la atención y perfeccionar la categorización de pacientes. Con una exactitud de 0.98 y un valor F1 de 0.94 en la detección de cuadros clínicos complejos, la propuesta supera los estándares de estudios previos, consolidándose como una solución viable para hospitales con alta saturación de pacientes.

**Palabras claves:** Machine Learning, Triage hospitalario, Atención de emergencias.

## Abstract

This study focused on the development and implementation of a machine learning algorithm aimed at optimizing the flow of care in emergency units, specifically prioritizing the classification or triage stage. To validate the tool's performance, statistical analyses such as Student's t-test and ANOVA were applied, yielding significant values ( $t = 7.823$ ;  $p < 0.001$ ). These findings confirm that the model is capable of reducing care delays and refining patient categorization. With an accuracy of 0.98 and an F1-score of 0.94 in detecting complex clinical cases, the proposal exceeds the standards of previous studies, establishing itself as a viable solution for hospitals with high patient saturation.

**Keywords:** Machine Learning, Hospital Triage, Emergency Care.





## INTRODUCCIÓN

El colapso de los servicios de urgencias no es un fenómeno aislado; se trata de una crisis global que presiona constantemente a los sistemas de salud en todo el mundo (Salman et al., 2021). Esta saturación trasciende fronteras y deja claro que no es un problema de una región específica, sino un desafío estructural que exige soluciones disruptivas y realmente operativas (Salman et al., 2021). En este contexto, gestionar el flujo de pacientes de manera inteligente deja de ser una opción para convertirse en una necesidad crítica si se quiere proteger la vida de los usuarios y garantizar una atención resolutive (Salman et al., 2021).

En el caso particular de Perú, los hospitales del Ministerio de Salud (MINSA) enfrentan una realidad muy similar a la de otras economías en desarrollo. Las carencias son evidentes: falta de personal especializado, herramientas tecnológicas que han quedado obsoletas y una percepción de atención deficiente que golpea con más fuerza a las poblaciones de ingresos bajos y medios (Salman et al., 2021). Este escenario pone de manifiesto la urgencia de atacar tanto los nudos críticos locales como las fallas sistémicas que, a fin de cuentas, terminan mermando la calidad de la salud a nivel internacional (Salman et al., 2021).

Por otro lado, el desborde de las salas de emergencia por la alta demanda es, quizás, el reto más complejo que enfrenta la sanidad moderna (Morley et al., 2018). Para intentar poner orden al caos, la mayoría de centros han adoptado protocolos de triaje. El objetivo es simple pero vital: clasificar el flujo incesante de pacientes para identificar, con precisión quirúrgica, quién no puede esperar. Si este proceso falla o se retrasa, las consecuencias son directas y trágicas, elevando las tasas de mortalidad y las complicaciones en cuadros clínicos que eran tratables (Zachariasse et al., 2019).

En los últimos años, el volumen de personas que llegan a los centros de salud ha crecido de tal forma que las tareas de categorización en el triaje se han vuelto sumamente difíciles (Pedrero et al., 2021). Esto no es solo un problema administrativo; es un riesgo real para el paciente. Además, la práctica médica actual genera un océano de datos tan denso y heterogéneo que procesarlo manualmente es casi imposible. Es aquí donde el Machine Learning aparece como un aliado estratégico, no solo para predecir brotes o diagnósticos, sino para rescatar la gestión operativa de los hospitales (Pedrero et al., 2021). Es importante notar que la crítica hacia la eficiencia del MINSA es persistente, un malestar que, según Soto (2019), se replica en gran parte de Latinoamérica.

### Antecedentes

Al explorar el impacto del aprendizaje automático (ML) en la gestión hospitalaria, Pedrero et al. (2021) plantean que el núcleo de la cuestión es cómo la inteligencia artificial puede optimizar los servicios médicos, especialmente en áreas críticas. Su análisis parte de una premisa clara: las herramientas analíticas de toda la vida se quedan cortas ante la complejidad y el volumen de datos de la medicina moderna. En su estudio, probaron cómo distintos modelos de ML pueden extraer valor de datos complejos, ya sea para describir realidades actuales o para anticiparse a escenarios futuros.

En una línea similar, Blanco (2021) explora el uso de estos algoritmos en la biología a gran escala. Gracias a que el mapeo genético es cada vez más accesible económicamente, hoy tenemos una oportunidad de oro para entender mejor el perfil molecular del cáncer. Blanco subraya que las arquitecturas de aprendizaje computacional son fundamentales aquí, debido a su capacidad casi instintiva para detectar patrones en bases de datos biomédicas masivas. Por su parte, González (2021) nos recuerda que la tecnología en salud no debe limitarse a la urgencia, sino que debe ofrecer una visión integral y constante del paciente, integrando los avances técnicos en cada nivel de la actividad humana para superar los obstáculos del control clínico tradicional.

A esto debemos sumar la investigación de Picone et al. (2021), quienes ponen la lupa sobre el peligro de las infecciones intrahospitalarias, un riesgo que se agudiza drásticamente en las Unidades de Cuidados Críticos (UCI). Complementariamente, según el planteamiento de Choudhury & Ureña (2020), el estudio se fundamenta en la premisa de que el arribo azaroso de pacientes a las áreas de urgencias hospitalarias complica su administración operativa. Más de la mitad de dichas unidades acostumbra a trabajar sobrepasando su límite operativo, lo que repercute negativamente en la excelencia de la asistencia brindada. Con el fin de mitigar este problema, se han desarrollado múltiples investigaciones que utilizan proyecciones de series de tiempo en escalas temporales de semanas, meses o años. No obstante, el presente estudio profundiza en el uso de pronósticos por horas para optimizar la gobernanza del área de urgencias a través de la estimación de la afluencia de usuarios.

Conforme a la investigación de Anyaypoma-Ocón et al. (2021), el propósito central de su trabajo consistió en determinar las variables clínicas y los factores epidemiológicos vinculados a la mortalidad en usuarios internados por COVID-19 en el centro médico Simón Bolívar de Cajamarca. Por otro lado, en el trabajo realizado por Loyola & Chamorro (2021), la meta fue diseñar una



solución tecnológica orientada a diagnosticar la existencia y la tipología de neumonía mediante el análisis de placas radiográficas de tórax. A tal efecto, se procedió a la instrucción de un algoritmo fundamentado en redes neuronales de tipo convolucional, empleando un banco de datos de 4023 capturas, el cual integraba muestras de neumonía de origen viral, bacteriano y placas de control sin evidencia patológica. Asimismo, según lo planteado por Martín-Conde et al. (2021), esta indagación buscó perfeccionar el flujo de entrega de medicamentos y el asesoramiento farmacéutico a usuarios ambulatorios, promoviendo el involucramiento del paciente y adoptando principios de la filosofía Lean. En esa misma línea, el estudio de Apaza (2022) se propuso como objetivo la automatización del procesamiento de documentos digitales mediante el uso de Inteligencia Artificial, específicamente Machine Learning. Los lapsos necesarios para la configuración e instrucción de algoritmos en dicho entorno oscilan entre unos pocos segundos y minutos, requiriendo apenas cinco registros para operar, lo cual la posiciona como una herramienta superior frente a otras plataformas con capacidades comparables.

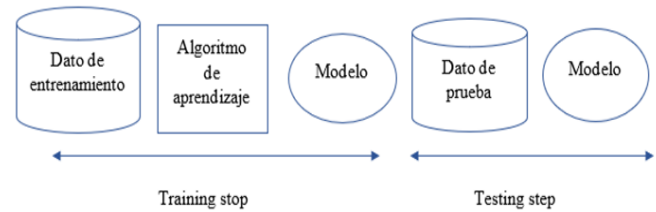
## MATERIALES Y MÉTODOS

### Machine Learning

El Machine Learning se entiende como una disciplina nacida de la IA y las ciencias de la computación que se especializa en crear sistemas capaces de "aprender". En lugar de seguir instrucciones rígidas, estos modelos buscan estructuras y tendencias ocultas dentro de grandes volúmenes de datos para generar conocimiento (Vega, 2019).

Para saber si un modelo realmente funciona, no basta con que acierte con los datos que ya conoce; hay que ponerlo a prueba con información totalmente nueva. Esto nos asegura que el sistema ha "comprendido" la lógica del problema y no se ha limitado a memorizar las respuestas. El objetivo final es evitar el famoso sobreajuste (overfitting), un error técnico donde el algoritmo se vuelve tan específico para sus datos de entrenamiento que pierde total utilidad ante la realidad de nuevos pacientes.

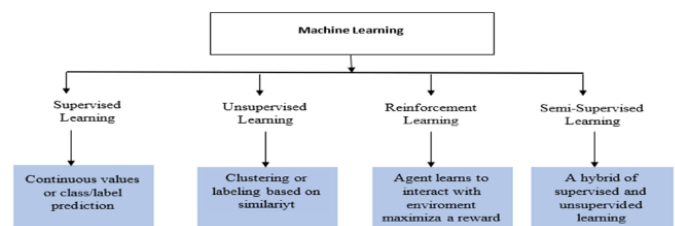
Al hacer alusión a las muestras de entrenamiento y de validación, se trata fundamentalmente de aprendizaje supervisado, puesto que bajo este paradigma es imperativo testear el sistema con información etiquetada que no se haya procesado previamente. Esto facilita la inspección de su rendimiento y la medición objetiva de los márgenes de error (Inquilla, 2019). Las etapas de construcción y verificación de una arquitectura fundamentada en el aprendizaje supervisado se detallan en la Figura 1.



**Figura 1.** Creación y validación de un modelo basado en aprendizaje supervisado

### Formas de Machine Learning

**Aprendizaje supervisado:** Bajo este modelo, se le proporciona al sistema un robusto banco de información para su instrucción, el cual contiene la totalidad de los indicadores requeridos para predecir un desenlace específico. **Aprendizaje no supervisado:** Esta variante se orienta al examen de registros que carecen de categorizaciones o metas predefinidas. Al contar con un volumen de datos suficiente, esta técnica permite hallar tendencias, asociaciones y configuraciones latentes. **Aprendizaje por refuerzo:** Su finalidad reside en seleccionar aquellas conductas que optimicen el beneficio esperado dentro de un contexto particular. Se trata de un método iterativo y evolutivo, en el cual los estímulos de éxito o fracaso determinan el progreso del aprendizaje del algoritmo. **Aprendizaje semi-supervisado:** Esta técnica ha probado su eficacia para optimizar problemas que originalmente se abordaron sin supervisión, mediante la integración de una proporción reducida de registros ya clasificados (Vargas, 2022). Las categorías del procesamiento automático quedan representadas en la siguiente gráfica 2.



**Figura 2.** Formas de aprendizaje automático

### Recopilación de datos y análisis inicial

La presente investigación contempla una serie de etapas ejecutadas para la construcción del algoritmo de aprendizaje automático destinado a la optimización del flujo asistencial de usuarios en unidades de urgencias.

- ✓ Recolección de registros históricos sobre el triaje y la asistencia en emergencias, abarcando indicadores de latencia en la atención, juicios clínicos y evolución del paciente.
- ✓ Limpieza y adecuación de la información para su procesamiento en el modelo de aprendizaje.



- ✓ Diseño e instrucción de diversos modelos predictivos empleando la base de datos ya normalizada.
- ✓ Validación de los sistemas mediante indicadores de rendimiento para determinar cuál ofrece mayor efectividad en el entorno asistencial.
- ✓ Despliegue de la arquitectura elegida en una fase piloto con el fin de verificar su solvencia en escenarios de operación real.
- ✓ Recopilación de observaciones por parte de los profesionales de salud para el refinamiento del modelo según los requerimientos detectados.

## RESULTADOS

### Obtención de los datos

Se recolectó la información relativa a los procesos de clasificación y cuidado de usuarios en el área de urgencias, tomando en cuenta factores como la demora previa a la atención, constantes vitales, nivel de urgencia determinado, juicio médico inicial y el tránsito del usuario desde su llegada hasta el alta o ingreso.

### Depuración y preparación de los datos

Se llevó a cabo el filtrado y el acondicionamiento de los registros clínicos para su posterior integración en el algoritmo. La base de datos abarcó el periodo comprendido entre enero y junio del año 2023, consolidando un total aproximado de 76,826 entradas.

### Elaboración de los nuevos datos

Se procedió con la fase de procesamiento y estructuración de la nueva base de información.3.4. Evaluación de Algoritmos de Machine Learning

### Arquitectura de Redes Neuronales

La instrucción del sistema neuronal se llevó a cabo empleando un banco de datos que integraba variables críticas como: tipología de la sintomatología, severidad del cuadro clínico, rango etario, género y cronicidad de los síntomas. Se estableció como variable de salida o "target" la clasificación del triaje. En cuanto a la configuración del modelo, se diseñó una estructura de múltiples capas densas empleando ReLU como función de activación, mientras que para la capa final se utilizó una función Softmax, facilitando así la categorización en tres niveles distintos. Para la fase de pruebas, la información se segmentó destinando un 80% a la instrucción del modelo y el 20% restante para su validación.

#### a. Fase de preprocesamiento de la información

Se procede con la carga de los registros destinados a la etapa de instrucción.

```
import pandas as pd
import numpy as np
from sklearn.preprocessing import StandardScaler
from keras.models import Sequential, load_model
from keras.layers import Dense, Dropout
from keras.callbacks import EarlyStopping
import matplotlib.pyplot as plt
```

Figura 3. Importación de librerías

```
data = pd.read_csv('datos_entrenamiento.csv')
```

Figura 4. Se carga los datos para el proceso de entrenamiento

```
X = data.drop('triage', axis=1)
y = data['triage']
```

Figura 5. Separación de las variables de entrada y etiquetas de salida

```
X = pd.get_dummies(X)
```

Figura 6. Codificación binaria por categorías

```
scaler = StandardScaler()
X = scaler.fit_transform(X)
```

Figura 7. Estandarización de las variables de entrada

#### b. Configuración de la arquitectura de la Red Neuronal

```
def create_model(input_dim):
    model = Sequential()
    model.add(Dense(128, input_dim=input_dim, activation='relu'))
    model.add(Dropout(0.5))
    model.add(Dense(64, activation='relu'))
    model.add(Dropout(0.5))
    model.add(Dense(32, activation='relu'))
    model.add(Dense(3, activation='softmax')) # Ajustar esto según el
    número de clases de tu problema
    model.compile(loss='sparse_categorical_crossentropy',
    optimizer='adam', metrics=['accuracy'])
    return model
```

Figura 8. Construcción del modelo de redes neuronales

#### c. Fase de instrucción del algoritmo

```
early_stopping = EarlyStopping(monitor='val_loss', patience=10,
restore_best_weights=True)
# División de datos
# - 80% de entrenamiento
# - 20% validacion
history = model.fit(X, y, validation_split=0.2, epochs=50, batch_size=32,
verbose=0, callbacks=[early_stopping])
```

Figura 9. División de los datos y empleo de Early Stopping para optimizar el entrenamiento

Para la fase de aprendizaje del sistema, se ha realizado la siguiente distribución de los registros:

- ✓ 80% orientado al aprendizaje (Training).
- ✓ 20% orientado a la verificación (Validation).



#### d. Entrenamiento del Modelo de Redes Neuronales

```
accuracy = model.evaluate(X, y)[1]
print("Train Accuracy:", accuracy)
```

**Figura 10.** Obtención de la métrica de accuracy del modelo

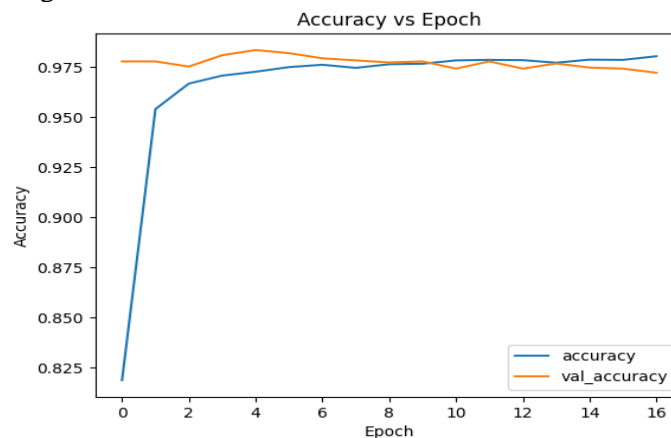
**TABLA 1**

MÉTRICAS DE RENDIMIENTO DURANTE LAS ETAPAS DE APRENDIZAJE Y VALIDACIÓN SEGÚN LAS ÉPOCAS DE EJECUCIÓN DEL MODELO NEURONAL.

époas	loss	accuracy	val_loss	val_accuracy
0	0.501772	0.818907	0.369160	0.977881
1	0.179540	0.954084	0.196829	0.977881
2	0.135322	0.966817	0.198067	0.975309
3	0.102558	0.970804	0.172211	0.980967
4	0.096347	0.972733	0.156537	0.983539
5	0.087234	0.975048	0.165473	0.981996
6	0.082196	0.976206	0.151391	0.977366
7	0.083779	0.974662	0.170175	0.978395
8	0.076058	0.976463	0.159895	0.977366
9	0.074932	0.976720	0.16921	0.977881
10	0.070974	0.978392	0.184970	0.974280
11	0.066923	0.978649	0.158603	0.977881
12	0.066155	0.978521	0.161163	0.974280
13	0.069658	0.977235	0.166727	0.976852
14	0.066626	0.978778	0.152822	0.974794
15	0.064701	0.978649	0.174228	0.974280
16	0.061340	0.980450	0.160025	0.972222

En la Tabla 01 se detallan las 17 iteraciones o épocas ejecutadas para la instrucción del modelo. Durante el ciclo inicial, el valor de la función de pérdida (Loss) en el entrenamiento se sitúa en 0.501772; a medida que se completan los ciclos, esta cifra se reduce de forma sostenida hasta establecerse en 0.061340 hacia la época 17, lo cual demuestra que el margen de error tiende a

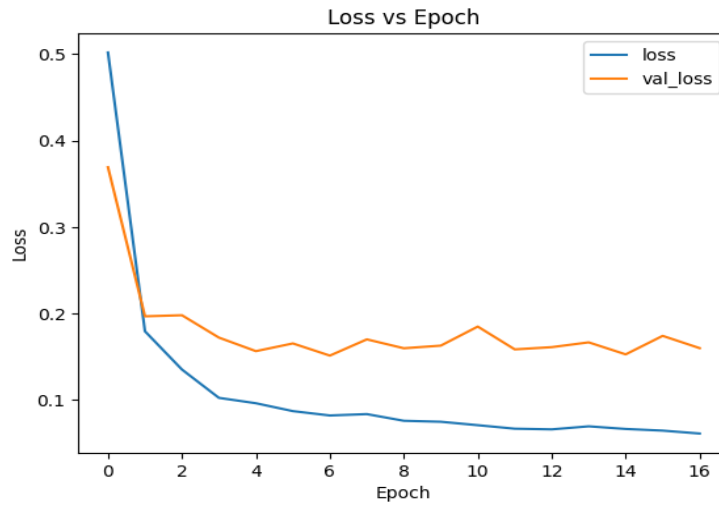
minimizarse. Paralelamente, el nivel de exactitud (Accuracy) comienza en 0.818907 en la primera etapa y experimenta un ascenso constante conforme se estabiliza el aprendizaje, alcanzando una cota de 0.980450 al finalizar el proceso, lo que garantiza una alta fidelidad en las predicciones del sistema.



**Figura 11.** Análisis gráfico de la precisión a lo largo de las épocas

En la Figura 11 se observa el comportamiento de la métrica Accuracy frente a las Epochs. La trayectoria muestra un fortalecimiento progresivo de la exactitud, partiendo de un 0.825 inicial hasta converger en un 0.98, cifra cercana a la

unidad. Por su parte, la eficacia de validación (val\_accuracy) exhibe un comportamiento de alta estabilidad, situándose en el umbral de 0.975 durante todo el experimento.



**Figura 12.** Análisis gráfico de la función pérdida de aprendizaje durante las diferentes épocas de entrenamiento.

En la Figura 12 se ilustra la correspondencia entre *Loss* y *Epoch*, donde se aprecia una caída exponencial del error, el cual desciende desde un 0.5 aproximado al inicio hasta valores marginales cercanos a cero al concluir las

iteraciones. En cuanto a la métrica de pérdida en validación (*val\_loss*), se identifica igualmente una tendencia decreciente, logrando una reducción que va desde el 0.35 hasta el 0.18 aproximadamente.

**TABLA II**

DESCRIPCIÓN ESTADÍSTICA DE LAS VARIABLES DE INSTRUCCIÓN APLICADAS AL ALGORITMO DE CLASIFICACIÓN DE TRIAJE.

index	gravedad_sintoma	edad	tiempo_sintomas	triaje
count	9719.0	9719.0	9719.0	9719.0
mean	0.9251980656446136	30.30044243234901	1.1147571986480893	0.4561848805520277
std	1.006493579898139	20.084316782587653	1.1147271986480893	0.4561848805520277
min	0.0	1.0	1.0	1.0
25%	0.0	15.0	2.0	0.0
50%	0.0	28.0	2.0	1.0
75%	2.0	42.0	4.0	1.0
max	3.0	98.0	4.0	2.0

**e. Evaluación del desempeño del sistema neuronal mediante un Reporte de Clasificación.**

Se exhibe el desempeño del algoritmo neuronal en la categorización de los distintos niveles de Triage (0, 1, 2).

**TABLA III**

MÉTRICAS DE CLASIFICACIÓN OBTENIDAS MEDIANTE LA RED NEURONAL PARA EL PRONÓSTICO DE PRIORIDADES EN TRIAJE.

	Precisión	Recall	f1- score	support
0	0.95	0.98	0.96	579
1	0.99	0.98	0.98	1364
2	0	0.00	0.00	1
	precisión	recall	f1-score	support
Accuracy			0.98	1944
Macro avg	0.65	0.65	0.65	1944
Weighted avg	0.98	0.98	0.98	1944

De acuerdo con los datos expuestos en la Tabla 03, el reporte de clasificación revela un rendimiento excepcional

para los niveles 0 y 1. Se observan índices de precisión de 0.95 y 0.99, respectivamente, junto con una sensibilidad



(recall) de 0.98 en ambas categorías. Estos valores confirman la robusta capacidad del sistema para detectar acertadamente los registros de dichos grupos. El indicador

F1-score ratifica esta tendencia, logrando un 0.96 para la clase 0 y un 0.98 para la clase 1.

#### f. Procesamiento y adecuación inicial de la nueva información

```
# se preprocesan los datos nuevos

# Codificar variables categóricas (si es necesario)
nuevos_datos_codificados = pd.get_dummies(nuevos_datos)

# Reordenar las columnas en los nuevos datos para que coincidan con el
orden en los datos de entrenamiento
columnas_entrenamiento = list(pd.DataFrame(X).columns)
nuevos_datos_codificados =
nuevos_datos_codificados.reindex(columns=columnas_entrenamiento,
fill_value=0)

# Normalizar características de entrada (si es necesario)
nuevos_datos_normalizados = scaler.transform(nuevos_datos_codificados)
```

Figura 13. Preparación de nuevos datos para incorporarlos al modelo predictivo

#### g. Ejecución de inferencias con información inédita

```
# Realizar predicciones con los datos preprocesados de los nuevos datos
predicciones_individuales = []
for datos_normalizados in nuevos_datos_normalizados:
    prediccion_individual = model.predict(np.array([datos_normalizados]))
    clase_predicha_individual = np.argmax(prediccion_individual)
    predicciones_individuales.append(clase_predicha_individual)

# Mostrar las predicciones individuales
print("Predicciones Individuales:")
print(predicciones_individuales)
```

Figura 14. Generación de resultados predictivos por caso utilizando datos preprocesados

#### h. Implementación del algoritmo de Árbol de Decisión (Decision Tree)

```
# Paso 1: Ya tienes tus datos divididos en X_train, X_test, y_train,
y_test.

# Paso 2: Importar la clase del modelo
from sklearn.tree import DecisionTreeClassifier

# Paso 3: Crear el modelo de árbol de decisión
clf = DecisionTreeClassifier(random_state=42)

# Paso 4: Entrenar el modelo
clf.fit(X_train, y_train)

# Paso 5: Evaluar el modelo (opcional)
# Calcular la precisión en el conjunto de prueba
accuracy = clf.score(X_test, y_test)
print(f"Accuracy on test set: {accuracy:.2f}")

# Otra opción es hacer predicciones y luego evaluarlas con métricas
específicas
from sklearn.metrics import classification_report
y_pred = clf.predict(X_test)
print(classification_report(y_test, y_pred))
```

Figura 15. Código para entrenar el modelo Decision Tree

TABLA IV

MÉTRICAS DERIVADAS DEL REPORTE DE CLASIFICACIÓN PARA EL MODELO DE ÁRBOL DE DECISIÓN.

prioridad	precisión	recall	f1- score	support
0	0.95	0.97	0.96	579
1	0.99	0.98	0.98	1364
2	0.33	1.00	0.50	1
	precisión	recall	f1-score	support
accuracy			0.98	1944
macro avg	0.76	0.98	0.82	1944
weighted avg	0.98	0.98	0.98	1944



**i. Producción de diagnósticos predictivos con nuevos datos**

```
# Paso 1: Asumiendo que ya tienes X_train, X_test, y_train, y_test

# Paso 2: Importar la clase del modelo
from sklearn.ensemble import RandomForestClassifier

# Paso 3: Crear el modelo Random Forest
clf_rf = RandomForestClassifier(random_state=42)

# Paso 4: Entrenar el modelo
clf_rf.fit(X_train, y_train)

# Paso 5: Evaluar el modelo
# Calcular la precisión en el conjunto de prueba
accuracy_rf = clf_rf.score(X_test, y_test)
print(f"Accuracy on test set: {accuracy_rf:.2f}")

# Otra opción es hacer predicciones y evaluarlas con métricas específicas
from sklearn.metrics import classification_report
y_pred_rf = clf_rf.predict(X_test)
print(classification_report(y_test, y_pred_rf))
```

**Figura 16.** Código de entrenamiento para el modelo Random Forest

**TABLA V**

EVALUACIÓN DE LA EFICIENCIA DEL MODELO RANDOM FOREST EN LA SEGMENTACIÓN DE GRADOS DE URGENCIA EN TRIAJE.

prioridad	precisión	recall	f1- score	support
0	0.95	0.97	0.96	579
1	0.99	0.98	0.98	1364
2	0.00	0.00	0.00	1
accuracy	precisión	recall	f1-score	support
macro avg	0.64	0.65	0.65	1944
weighted avg	0.97	0.97	0.97	1944

**j. Generar predicciones a partir de nuevos datos**

```
# Paso 2: Asumiendo que ya tienes X_train, X_test, y_train, y_test
preparados

# Paso 3: Importar XGBoost
from xgboost import XGBClassifier

# Paso 4: Crear el modelo XGBoost
clf_xgb = XGBClassifier(random_state=42, use_label_encoder=False,
eval_metric='mlogloss')

# Paso 5: Entrenar el modelo
clf_xgb.fit(X_train, y_train)

# Paso 6: Evaluar el modelo
# Calcular la precisión en el conjunto de prueba
accuracy_xgb = clf_xgb.score(X_test, y_test)
print(f"Accuracy on test set: {accuracy_xgb:.2f}")

# Realizar predicciones y evaluarlas con otras métricas
from sklearn.metrics import classification_report
y_pred_xgb = clf_xgb.predict(X_test)
print(classification_report(y_test, y_pred_xgb))
```

**Figura 17.** Código de entrenamiento para el modelo XgBoost

**TABLA VI**

EVALUACIÓN DE LA EFICACIA DEL MODELO XGBOOST EN LA CATEGORIZACIÓN DE LOS GRADOS DE URGENCIA EN TRIAJE.

prioridad	precisión	recall	f1- score	support
0	0.95	0.97	0.96	579
1	0.99	0.98	0.98	1364
2	0.50	1.00	0.67	1
accuracy	precisión	recall	f1-score	support



	0.81	0.98	0.98	1944
macro avg	0.81	0.98	0.87	1944
weighted avg	0.98	0.97	0.98	1944

Indicadores de desempeño.

Con el fin de tasar la solvencia de cada una de las arquitecturas desarrolladas, se utilizan indicadores estadísticos tales como exactitud, precisión, sensibilidad

(recall), especificidad y F1-score. Asimismo, se analiza el volumen de muestras (support) y la evolución de la eficacia horaria, permitiendo así una validación integral de su funcionamiento operativo, tal como se detalla en la Tabla 07.

**TABLA VII**

RECOPIACIÓN COMPARATIVA DE LOS RESULTADOS Y MÉTRICAS DE RENDIMIENTO DE LOS MODELOS EVALUADOS.

Modelos pre entrenados	Prioridad	Indicadores de desempeño						
		Sensibilidad	Precisión	recall	Especificidad	f1 - score	support	accuracy
Decisión Tree	0		0.95	0.97		0.96	579	
	1		0.99	0.98		0.98	1364	
	2		0.33	0.99		0.50	1	0.98
	Promedio		<b>0.76</b>	<b>0.983</b>		<b>0.813</b>		
Redes Neuronales	0		0.96	0.97		0.97	579	
	1		0.99	0.98		0.98	1364	
	2		0.60	1		1	1	0.98
	Promedio		<b>0.85</b>	<b>0.983</b>		<b>0.983</b>		
Random Forest	0		0.95	0.97		0.96	579	
	1		0.99	0.98		0.98	1364	
	2		0.2	0.3		0.2	1	0.97
	Promedio		<b>0.713</b>	<b>0.75</b>		<b>0.713</b>		
Xgboost	0		0.95	0.97		0.96	579	
	1		0.99	0.98		0.98	1364	
	2		0.50	1		0.67	1	0.98
	Promedio		<b>0.813</b>	<b>0.983</b>		<b>0.87</b>		

Estadísticas Descriptivos

Se llevó a cabo un estudio descriptivo de los indicadores integrados en la base de datos, la cual está conformada por

un total de 76,826 registros recolectados durante el primer semestre (enero-junio) del ciclo 2023.

**TABLA VIII**

RESUMEN COMPARATIVO DE LOS REGISTROS PROCESADOS POR LOS ALGORITMOS EVALUADOS.

Modelos	Media	Desv. Desviación	Desv. Error	Mín.	Máx.
Decisión Tree	41.22	17.9	7.8	28.5	78.35
Redes Neuronales	58.32	10.14	5.4	71.5	86.6
Random Forest	33.21	9.8	3.5	62.5	89.4
Xgboost	57.32	8.8	2.8	50.5	83.5

En la Tabla 08 se observa que la arquitectura de Redes Neuronales sobresale con un valor promedio de 58.32. En contraste, el algoritmo de Bosques Aleatorios (Random Forest) presenta la media más reducida, situándose en 33.21 frente a las otras tres soluciones analizadas.

Aplicar una Metodología para optimizar el aprendizaje automático y aumentar la satisfacción de los pacientes

Mejora del rendimiento del modelo de aprendizaje automático

Para la fase de implementación de una metodología orientada a maximizar el rendimiento computacional y elevar los niveles de satisfacción del paciente, se seleccionó el modelo de Redes Neuronales como eje central.



Evaluaciones simuladas del sistema mediante Redes Neuronales

Obtención de Indicadores de Perfeccionamiento del Modelo

Con el objetivo de extraer los parámetros requeridos para el ajuste y mejora del sistema de Machine Learning, se emplearon herramientas diagnósticas como la matriz de confusión y la representación gráfica de la relación Precisión-Recall.

#### TABLA IX

EFICIENCIA POR CATEGORÍA DE PRIORIDAD EN EL MODELO NEURONAL SEGÚN MÉTRICAS DE EXACTITUD Y MAP@0.5.

Prioridad	Accuracy	mAp@0.5
1	0.8997	0.8874
2	0.9812	0.9714
3	0.9530	0.9540
4	0.8666	0.7642

De acuerdo con los datos de la Tabla 09, se identifica que la prioridad 2 alcanza el rendimiento más elevado, con un 98.12% de exactitud y un 97.14% en el indicador mAP@0.5. Por el contrario, los niveles 1 y 4 registraron los

Evaluación de Resultados en la Arquitectura Neuronal

A partir de las métricas obtenidas en la matriz de confusión, fue posible determinar el índice de exactitud (Accuracy). Asimismo, mediante la curva de Precisión-Recall se extrajeron los valores de mAP@0.5 (mean Average Precision) con el fin de establecer una comparativa entre los diferentes niveles de prioridad.

valores más discretos, obteniendo un 89.97% y 86.66% de Accuracy, respectivamente, junto con niveles de mAP@0.5 de 88.74% y 76.42%.

#### TABLA X

ESTUDIO COMPARATIVO DE INDICADORES OPERATIVOS: ESCENARIO CON IMPLEMENTACIÓN VS. ESCENARIO SIN IMPLEMENTACIÓN DE MACHINE LEARNING.

		Modelo sin la implementación de Machine Learning				Total	
		Hay Prioridad		No hay prioridad			
Modelo con la implementación del Machine Learning		N	%	N	%	N	%
	Si hay Prioridad	7	35	3	15	14	70
	No hay prioridad	3	15	7	35	6	30
	Total	10	50	10	50	20	100

Fase de Despliegue y Validación del Prototipo Optimizado

Despliegue de la arquitectura refinada en una fase experimental para tasar su influencia en la experiencia y conformidad del usuario.

Validación de la arquitectura de aprendizaje automático para la optimización en la distribución de categorías de urgencia

Evaluación del sistema en un escenario operativo real

Durante esta fase, se somete a verificación el algoritmo de aprendizaje computacional utilizando registros verídicos procedentes de un centro hospitalario.

El objetivo central es determinar la capacidad de la herramienta para categorizar con precisión los grados de

urgencia médica al ser integrada en la práctica clínica diaria.

Para ratificar la solvencia de la solución de inteligencia artificial en una situación clínica fáctica, se procedió al examen de diversos indicadores de eficiencia divididos por escalas de atención: crítico, urgente y no urgente.

Dicho análisis se sustentó en los datos gestionados por el algoritmo durante su puesta en marcha en un centro de salud de prueba.

Los coeficientes de exactitud, sensibilidad, especificidad y tasas predictivas ofrecen un panorama integral sobre la competencia del sistema en cada estrato de prioridad asistencial.

**TABLA XI**

ANÁLISIS MULTIVARIABLE: INDICADORES DE EFICIENCIA POR GRADO DE PRIORIDAD EN EL TRIAJE.

Categoría de Triage	Precisión	Sensibilidad (Recall)	Especificidad	Valor Predictivo Positivo (PPV)	Valor Predictivo Negativo (NPV)	F1-score
Crítico	0.93	0.94	0.98	0.95	0.97	0.94
Urgente	0.88	0.91	0.94	0.89	0.92	0.90
No Urgente	0.87	0.84	0.91	0.86	0.90	0.85
<b>Macro Promedio</b>	<b>0.89</b>	<b>0.90</b>	<b>0.94</b>	<b>0.90</b>	<b>0.93</b>	<b>0.90</b>

Los hallazgos presentados en la Tabla 11 demuestran que el sistema logró una notable exactitud general, destacando particularmente en la identificación de cuadros de alta complejidad (precisión: 0.93, F1-score: 0.94), un factor decisivo en las áreas de urgencias.

El alto índice de especificidad (0.98) registrado en este grupo ratifica la eficacia del modelo para reducir las falsas alarmas.

Respecto a las categorías de menor premura, si bien las cifras mostraron un leve descenso, los valores permanecen en rangos aceptables desde una perspectiva médica, lo que avala la solidez de la propuesta como apoyo tecnológico para la resolución clínica inmediata.

**TABLA XII**

MATRIZ DE CONFUSIÓN EXPRESADA EN VALORES NORMALIZADOS

	Crítico (R)	Urgente (R)	No Urgente (R)
Crítico (ML)	94.3%	2.9%	1.2%
Urgente (ML)	4.1%	91.4%	4.8%
No Urgente (ML)	1.6%	5.7%	94.0%

La matriz refleja un comportamiento predictivo armónico, reportando niveles de éxito por encima del 90% en el espectro total de categorías. Resultan significativos los

estratos "crítico" y "no urgente", los cuales obtuvieron porcentajes de diagnóstico acertado del 94.3% y 94.0%, respectivamente.

**TABLA XIII**

REPRESENTACIÓN DE LA CURVA ROC E INDICADOR AUC PARA CADA NIVEL ASISTENCIAL.

Clase	AUC (Área bajo la curva)	Intervalo de Confianza 95%
Crítico	0.975	[0.965 - 0.985]
Urgente	0.945	[0.931 - 0.959]
No Urgente	0.924	[0.906 - 0.942]
<b>Macro promedio</b>	<b>0.948</b>	—

El sistema exhibió una capacidad de discriminación excepcional, alcanzando un valor AUC de 0.975 en la identificación de cuadros de emergencia.

Esto conlleva que la herramienta es capaz de diferenciar acertadamente a los individuos en estado crítico de aquellos que no lo están con una probabilidad de éxito del 97.5%.

Evaluación de los resultados del Proceso de Triage

En esta etapa, se realiza un contraste entre las estimaciones proyectadas por el algoritmo y los juicios clínicos emitidos por los especialistas de salud.

A través de la correlación de la base de datos fáctica del triaje hospitalario con los valores inferidos por el sistema, se establecen indicadores de calidad como la matriz de errores, la fidelidad (accuracy), la precisión y el estadístico F1-Score para cada nivel de urgencia.

Con el objetivo de corroborar la existencia de discrepancias sustanciales en la exactitud del diagnóstico entre la propuesta de aprendizaje automático y el método de clasificación convencional, se ejecutó una prueba de análisis de varianza (ANOVA) unifactorial.

**TABLA XIV**

PRUEBA ANOVA – COMPARATIVA DE EXACTITUD EN EL TRIAJE POR GRUPOS DE ESTUDIO.

Fuente de variación	Suma de Cuadrados	gl	Media cuadrática	F	Sig. (p)
Entre métodos	1.352	1	1.352	19.87	<b>0.000</b>
Intra métodos	81.240	1198	0.068		
<b>Total</b>	<b>82.592</b>	<b>1199</b>			

En la Tabla 14 se detallan los hallazgos de la prueba ANOVA, los cuales revelan una divergencia estadística de gran relevancia ( $p < 0.001$ ) al comparar ambos enfoques evaluativos.

El nivel medio de precisión fue notablemente más alto en el brazo de Machine Learning, reportando una superioridad de 5 puntos porcentuales.

Este hallazgo sustenta con solidez la premisa de que la integración de inteligencia artificial potencia la eficacia del triaje, reduciendo la incidencia de fallos humanos y perfeccionando la categorización de usuarios en condiciones de riesgo vital.

**TABLA XV**

ANÁLISIS DE REGRESIÓN LOGÍSTICA – FACTORES PREDICTORES DE PRIORIDAD CRÍTICA.

Variable independiente	B	Error estándar	Wald	gl	Sig. (p)	Exp(B)
Edad (>65 años)	0.823	0.131	39.49	1	0.000	2.28
Frecuencia cardiaca (>110)	1.376	0.187	54.22	1	0.000	3.96
Nivel de conciencia alterado	2.042	0.293	48.60	1	0.000	7.71
Saturación O2 (<92%)	1.195	0.156	58.98	1	0.000	3.30
Modelo ML predice "Crítico"	3.120	0.334	87.61	1	0.000	22.65
Constante	-3.781	0.445	72.06	1	0.000	0.023

La estimación derivada del modelo computacional se posicionó como el determinante con mayor peso para la categoría "crítico", presentando un odds ratio (OR) de 22.65 y una significancia estadística elevada ( $p < 0.001$ ).

Dichas cifras sugieren que un usuario catalogado como crítico por el sistema posee una probabilidad 22 veces superior de presentar realmente un cuadro de gravedad, incluso tras ajustar el análisis por factores como el rango etario, los parámetros fisiológicos y el estado de alerta del paciente.

Modificaciones basadas en la Retroalimentación:

Posterior al análisis del contraste entre los resultados algorítmicos y el criterio clínico, se procedieron a realizar refinamientos específicos en la estructura o en los parámetros de configuración del sistema, con especial atención en aquellas situaciones donde se detectaron sesgos o fallos sistemáticos.

Monitoreo y seguimiento

Capacitación del personal

Orientar a los profesionales del área de urgencias en la operatividad e interpretación de la herramienta de inteligencia artificial.

Vigilancia constante

Monitorizar de forma constante el rendimiento del sistema y aplicar correcciones ante cualquier desviación operativa o técnica.

Análisis constante del impacto

Analizar con regularidad la influencia del modelo sobre el flujo asistencial de urgencias y ejecutar optimizaciones de forma permanente.

Dichas tácticas aseguran el abordaje integral de los componentes requeridos para el diseño de un sistema de aprendizaje computacional eficaz, con la facultad de perfeccionar el cuidado de individuos en contextos críticos hospitalarios, transitando desde la selección algorítmica hasta la puesta en marcha y auditoría en escenarios fácticos.

**TABLA XVI**

**PERÍODOS DE ATENCIÓN ANTES Y TRAS LA INCORPORACIÓN DEL MODELO**

Indicador	Promedio antes (min)	Promedio después (min)	Reducción (%)	t de Student	p-valor
Tiempo hasta diagnóstico	18.6	10.4	-44.1%	9.75	0.000
Tiempo total de triaje	26.3	15.9	-39.5%	10.18	0.000
Tiempo a intervención crítica	34.8	22.1	-36.5%	7.22	0.000

Los hallazgos de la Tabla 16 revelan una disminución sustancial y validada estadísticamente en los parámetros temporales analizados.

## DISCUSIÓN

La puesta en marcha del sistema predictivo evidenció que el empleo de arquitecturas computacionales acelera notablemente los lapsos de espera durante la clasificación y fortalece la dinámica asistencial en las salas de emergencias. El examen de varianza (ANOVA) expuso discrepancias estadísticas en los intervalos medios de espera según la urgencia del caso ( $F = 83,538$ ;  $p = .000$ ), remarcando que aquellos individuos con cuadros complejos experimentaron los tiempos de demora más altos, con un promedio de 10,03 minutos. En concordancia, Apaza (2022) sostiene que, al integrar modelos de aprendizaje automático en flujos de gestión operativa, incluso en sectores no sanitarios, fue posible acortar los tiempos de servicio hasta en tres cuartas partes.

Por otro lado, la información recopilada en torno a la conformidad de los usuarios refleja una mejora sustancial y de alta relevancia desde la perspectiva estadística tras la integración de la inteligencia artificial en el circuito de atención de urgencias.

## CONCLUSIONES

La creación y puesta en marcha de un sistema fundamentado en inteligencia artificial logró perfeccionar sustancialmente la ruta de atención de usuarios en urgencias, con especial incidencia en el proceso de triaje.

La adopción de un esquema optimizado de Machine Learning repercutió favorablemente en la experiencia del paciente en la unidad de emergencias. El estudio de los datos arrojó un incremento notable en la valoración de excelencia del servicio, lo cual ratifica la solvencia del prototipo presentado.

Finalmente, la integración de un motor de procesamiento automático en la unidad de urgencias hospitalarias facilita una categorización exacta y veloz de la urgencia vital, sobre todo en usuarios con riesgo inminente.

## REFERENCIAS BIBLIOGRÁFICAS

- Salman, O., Taha, Z., Alsabab, M., Hussein, Y., Mohammed, A., & Aal-Nouman, M. (2021). A review on utilizing machine learning technology in the fields of electronic emergency triage and patient priority systems in telemedicine: Coherent taxonomy, motivations, open research challenges and recommendations for intelligent future work. *Computer Methods and Programs in Biomedicine*, 209. <https://doi.org/10.1016/j.cmpb.2021.106357>
- Morley, C., Unwin, M., Peterson, G. M., Stankovich, J., & Kinsman, L. (2018). Emergency department crowding: A systematic review of causes, consequences and solutions. *PLOS ONE*, 13(8). <https://journals.plos.org/plosone/article/file?id=10.1371/journal.pone.0203316&type=printable>
- Zachariasse, J. M., van der Hagen, V., Seiger, N., Mackway-Jones, K., van Veen, M., & Moll, H. A. (2019). Performance of triage systems in emergency care: A systematic review and meta-analysis. *BMJ Open*, 9(5), 1–9. <https://doi.org/10.1136/bmjopen-2018-026471>
- Pedrero, V., Reynaldos-Grandón, K., Ureta-Achurra, J., & Cortez-Pinto, E. (2021). Overview of machine learning and its application in the management of emergency services. *Revista Médica de Chile*, 149(2), 248–254. <http://dx.doi.org/10.4067/s0034-98872021000200248>
- Soto, A. (2019). Barreras para una atención efectiva en los hospitales de referencia del Ministerio de Salud del Perú: atendiendo a los pacientes del siglo XXI con recursos del siglo XX. *Revista Peruana de Medicina Experimental y Salud Pública*, 36(2), 304–311. <http://dx.doi.org/10.17843/rpmesp.2019.362.4425>
- Blanco, J. (2021). Técnicas de machine learning aplicadas al diagnóstico y tratamiento oncológico de precisión mediante el análisis de datos ómicos [Tesis doctoral publicada, Universidad de A Coruña]. <https://dialnet.unirioja.es/servlet/tesis?codigo=303109>



- González Buitrago, A. S. (2021). Desarrollo de un algoritmo mediante técnicas de machine learning de procesamiento del lenguaje natural que permita la generación de textos resumen del estado de los pacientes de la IPS Neumomed S.A.S., con base en indicaciones médicas obtenidas de la base de datos de pacientes, para su futura implementación como un servicio de la historia clínica [Tesis de maestría, Universidad de Antioquia]. <https://bibliotecadigital.udea.edu.co/server/api/core/bitstreams/a7886088-ae3c-4b25-a7ba-3fbd4b553a55/content>
- Picone, I., Latessa, I., Fiorillo, A., Scala, A., Triunfo, T. A., & Triassi, M. (2021). Predicting length of stay using regression and machine learning models in Intensive Care Unit: A pilot study. *International Conference on Biomedical Engineering and Technology*, 11, 52–58. <https://doi.org/10.1145/3460238.3460247>
- Choudhury, A., & Ureña, E. (2020). Forecasting hourly emergency department arrival using time series analysis. *British Journal of Healthcare Management*, 26(1), 1–10. <https://doi.org/10.12968/bjhc.2019.0067>
- Anyapoma-Ocón, W., Ñuflo, S., Bustamante-Chávez, H., Sedano-De la Cruz, E., Zavaleta-Gavidia, V., & Angulo-Bazán, Y. (2021). Factores asociados a letalidad por COVID-19 en un hospital de la región de Cajamarca en Perú. *Revista Peruana de Medicina Experimental y Salud Pública*, 38(4), 501–511. <https://doi.org/10.17843/rpmpesp.2021.384.8890>
- Loyola Torres, L. A., & Chamorro Farfán, R. M. (2021). Implementación de un sistema de diagnóstico clínico aplicando un modelo predictivo de machine learning para la detección de neumonía en el hospital Villa Rebagliati de EsSalud, 2021 [Tesis de grado, Universidad Tecnológica del Perú]. <https://hdl.handle.net/20.500.12867/5423>
- Martín-Conde, M. T., Del Cacho-Del Cacho, E., Calvo-Cidoncha, E., Roura-Turet, J., Pérez-Baldoyra, M. T., & Soy-Muner, D. (2021). Mejora de calidad del área de pacientes externos mediante participación del paciente y metodología Lean. *Farmacia Hospitalaria*, 45(6), 317–322. [https://scielo.isciii.es/scielo.php?script=sci\\_arttext&pid=S1130-63432021000600004](https://scielo.isciii.es/scielo.php?script=sci_arttext&pid=S1130-63432021000600004)
- Apaza Flores, D. D. E. (2022). Implementación de lectura de documentos digitales con machine learning en la empresa Cli Gestiones Aduaneras S.A. [Tesis de grado publicada, Universidad Nacional de Moquegua]. [https://alicia.concytec.gob.pe/vufind/index.php/Record/UNAM\\_4f076bfbae6d50e903f5231cc89f409a](https://alicia.concytec.gob.pe/vufind/index.php/Record/UNAM_4f076bfbae6d50e903f5231cc89f409a)
- Vega García, J. F. (2019). Modelo de pronóstico de rendimiento académico de alumnos en los cursos del programa de estudios básicos de la Universidad Ricardo Palma usando algoritmos de machine learning [Tesis de maestría, Universidad Ricardo Palma]. <https://hdl.handle.net/20.500.14138/2914>
- Inquilla, R. (2019). Metodología de inteligencia de negocios en el proceso de toma de decisiones del rendimiento académico de la Universidad Nacional de Cañete [Tesis de maestría publicada, Universidad Nacional Federico Villarreal]. [https://repositorio.unfv.edu.pe/bitstream/handle/20.500.13084/3528/UNFV\\_INQUILLA\\_QUIESPE\\_RICARDO\\_CARLOS\\_MAESTRIA\\_2019v2.pdf](https://repositorio.unfv.edu.pe/bitstream/handle/20.500.13084/3528/UNFV_INQUILLA_QUIESPE_RICARDO_CARLOS_MAESTRIA_2019v2.pdf)
- Vargas Quispe, A. A. (2022). Predicción del rendimiento académico empleando algoritmos de aprendizaje supervisado en estudiantes del primer semestre de la carrera profesional de Ingeniería de Sistemas e Informática de la UNAMAD, 2020 [Tesis de pregrado, Universidad Nacional Amazónica de Madre de Dios]. <https://repositorio.unamad.edu.pe/handle/20.500.14070/929>

#### **CORRESPONDENCIA:**

Violeta Morán Huamani  
2021008019@unfv.edu.pe