

Instituto Tecnológico y de Estudios Superiores de Monterrey

Inteligencia Artificial Avanzada para la Ciencia de Datos

Modelado

Los concentrados v2

Daniel Queijeiro Albo - A01710441

Diego Alfaro Pinto - A01709971

Diego Isaac Fuentes Juvera - A01705506

Jesus Ramirez Delgado - A01274723

Mauricio Anguiano Juarez - A01703337

Luis Adrián Uribe Cruz - A01783129

Adaptaciones CRISP-DM	5
1.0 Introducción	6
2.0 Selección de técnica de modelado	7
2.1 Aproximación inicial del modelo (descartada)	7
2.1 Técnicas seleccionadas y justificación	9
2.1.1 Modelo de comportamiento	9
2.1.2 Modelo de sanidad	10
2.1.3 Mérito productivo	11
2.1.4 Integración en el Índice de Mérito para la Retención (IMR)	12
Es importante mencionar que los pesos seleccionados son en base a lo que tenemos más certeza o datos certeros en nuestros modelos, dando mayor influencia a el modelo de comportamiento dados los datos que nos confirman con mayor precisión el comportamiento de una vaca. Estos pesos se pueden ajustar si se quiere dar más influencia o menos a alguna de las dimensiones.	12
3.0 Pipeline	13
4.0 Dataset	13
5.0 División de datos	14
6.0 Pruebas	14
6.1 Métricas de evaluación	14
6.2 Criterios de evaluación	15
6.3 Artefactos a generar	16
6.3.1 Modelo de comportamiento - Random Forest	16
6.3.2 Modelo sanidad - Isolation Forest	16
6.4 Plan de evaluación de modelos	16
6.3.1 Modelo de Comportamiento - Random Forest	16
6.3.2 Modelo de Sanidad - Isolation Forest	17
7.0 Construir el modelo	17
7.1 Selección de librerías	17
7.2 Modelo de comportamiento base (Regresión Logística)	18
7.2.1 Descripción	18
7.2.2 Configuración de parámetros	18
7.2.3 Codificación	19
7.3 Modelo de sanidad base (Density Clustering)	19
7.3.1 Descripción	19
7.3.2 Configuración de parámetros	19
7.3.3 Codificación	20
7.4 Modelo de comportamiento v1.0	20
7.4.1 Descripción	20
7.4.2 Configuración de parámetros	21
7.4.3 Codificación	22
7.5 Modelo de sanidad v1.0	22
7.5.1 Descripción	22

5.5.2 Configuración de parámetros	22
7.5.3 Codificación	23
7.6 Modelo comportamiento v2.0	23
7.6.1 Descripción	23
7.6.2 Configuración de parámetros	23
7.6.3 Codificación	24
7.7 Modelo sanidad v2.0	24
7.7.1 Descripción	24
7.7.2 Configuración de parámetros	24
7.7.3 Codificación	25
7.8 Modelo comportamiento v2.1	26
6.8.1 Descripción	26
7.8.2 Configuración de parámetros	26
7.8.3 Codificación	27
7.9 Modelo sanidad v2.1	27
7.9.1 Descripción	27
7.9.2 Configuración de parámetros	27
7.8.3 Codificación	29
7.10 Modelo de comportamiento MLP v1.0.	29
7.10.1 Descripción	29
7.10.2 Configuración de parámetros	29
7.10.3 Codificación	31
8.0 Evaluación	31
8.1 Evaluación modelo de comportamiento base regresión logística	31
8.1.1 Resultados	31
8.1.2 Observaciones	32
8.1.3 Conclusiones del modelo base:	33
8.2 Evaluación modelo de sanidad base	33
8.2.1 Resultados	33
8.2.2 Observaciones	35
8.3 Evaluación modelo de comportamiento v1.0	36
8.3.1 Resultados	36
8.3.2 Observaciones	37
8.3.3 Decisión sobre el modelo y el dataset	38
8.4 Evaluación modelo de sanidad v1.0	39
8.4.1 Resultados	39
8.4.2 Observaciones	40
8.4.3 Decisión sobre el modelo y el dataset	41
8.5 Evaluación de modelo de comportamiento v2.0	41
8.5.1 Resultados	41
8.5.2 Observaciones	43

8.5.3 Optimización y fine tuning	44
8.6 Evaluación de modelo de sanidad Isolation Forest v2.0	45
8.6.1 Resultados	45
8.6.2 Observaciones	47
8.6.3 Optimización y fine tuning	48
8.7 Evaluación de modelo de comportamiento Random Forest v2.1	48
8.7.1 Resultados	48
8.7.2 Observaciones	50
8.7.3 Conclusiones modelo Random Forest 2.1	51
8.8 Evaluación de modelo de sanidad Isolation Forest v2.1	51
8.8.1 Resultados	51
8.8.2 Observaciones	53
8.8.3 Conclusión modelo Isolation Forest v2.1	54
8.9 Evaluación modelo de comportamiento MLP	54
8.9.1 Resultados	54
8.9.2 Observaciones	56
8.9.3 Conclusión sobre modelo MLP v1.0	58
8.10 Comparativo	58
8.10.1 Modelos de comportamiento	58
8.10.2 Conclusión general comparativo modelos de comportamiento	58
8.10.3 Modelos de sanidad	60
8.10.4 Conclusión general comparación modelos de sanidad	61
9.0 Conclusión final	62

Adaptaciones CRISP-DM

Si bien la metodología CRISP-DM plantea una estructura general para la fase de modelado que incluye seleccionar la técnica, diseñar pruebas, construir modelos y evaluar su desempeño, el proyecto requirió ajustar dicha secuencia para adecuarla tanto a la naturaleza iterativa del desarrollo como a la documentación requerida.

1. Selección de Técnica de Modelado

El proceso original contempla una selección inicial de algoritmos.

En nuestro caso, esta actividad se convirtió en un proceso iterativo:

- Las técnicas se seleccionaron conforme se avanzaba sobre los resultados obtenidos, retroalimentando la decisión con evidencia experimental.
- Cada modelo probado fue documentado directamente dentro de la misma sección, en lugar de una decisión única al inicio.

Esto permitió alinear la selección con hallazgos empíricos y no limitarla a una suposición previa.

2. Generate test design □ Pruebas

Se mantuvo la intención original de la metodología, pero:

- Se estandarizaron esquemas de evaluación repetibles para todos los modelos, asegurando comparabilidad entre técnicas.
- Se establecieron criterios de éxito que satisfacen la evaluación del objetivo de minería de datos y tener una forma de medir los resultados.

Esta adaptación permitió un marco común de validación entre modelos supervisados y no supervisados.

3. Build Model □ Construir el modelo

Al igual que en el proceso original, se desarrollaron y ejecutaron los modelos; sin embargo:

- La documentación se realizó individualmente para cada modelo.
- Dentro de esta actividad se incluyeron descripciones, parámetros utilizados y detalles operativos específicos.

En lugar de un único apartado, la fase se fragmentó en múltiples ejecuciones y registros, reflejando una naturaleza incremental del modelado y experimentación.

4. Evaluación del Modelo (Adaptación a Evaluación Extendida)

CRISP-DM incluye una actividad denominada *Assess the Model*. Esta fase fue reemplazada y ampliada por nosotros con una actividad llamada **Evaluación**, estructurada con subactividades específicas por modelo:

- **Resultados:** métricas cuantitativas generadas.
- **Observaciones:** interpretación cualitativa de desempeño, estabilidad y comportamiento visual del modelo.
- **Acciones derivadas:** decisión explícita sobre el destino del modelo:
 - Seleccionar el modelo,
 - Descartar,
 - O derivarlo a una fase de **Optimización y Fine Tuning**.

Esta modificación permitió evaluar explícitamente el cumplimiento de los criterios de éxito de minería de datos, conectando las métricas con decisiones.

5. Conclusión General de la Fase

Finalmente, se añadió una actividad adicional no contemplada explícitamente en CRISP-DM:

- Conclusiones generales, donde se sintetizan hallazgos, lecciones y decisiones estratégicas sobre el conjunto de modelos.

Esta etapa permitió cerrar la fase de modelado con una visión global, justificando el camino tomado y preparando la transición hacia la fase de evaluación general del proyecto. Además se agregaron actividades que consideramos de importancia para brindar mejor contexto, las actividades fueron:

- Pipeline: Describe la secuencia de nuestros modelos y los datos.
- Dataset Describe brevemente los datasets empleados.
- División de datos: Describe la división de datos seleccionada para cada modelo.

1.0 Introducción

Esta fase de modelado tiene como propósito seleccionar y entrenar modelos de análisis y predicción utilizando los datos previamente depurados y entendidos durante las etapas anteriores. En esta fase se definen las técnicas de modelado, las variables predictoras a utilizar, los criterios de evaluación y los procedimientos para comparar el desempeño de diferentes modelos.

El enfoque de modelado debe alinearse directamente con los objetivos de negocio y minería de datos establecidos, priorizando tanto el rendimiento del modelo como su interpretabilidad para apoyar decisiones en el manejo del hato bovino.

Dado que nuestro proyecto busca facilitar la toma de decisiones sobre mejora genética del hato, el propósito del modelado es construir un sistema de apoyo a decisiones que permita determinar qué vacas

deben ser retenidas, supervisadas o descartadas, basado en tres dimensiones clave del desempeño en el estable:

1. Comportamiento en la celda de ordeño
2. Estado sanitario/salud o riesgo de problemas de ubre
3. Capacidad productiva (mérito productivo ajustado)

Estas tres dimensiones se integrarán posteriormente en un **Índice de Mérito para Retención (IMR)** que permitirá clasificar cada vaca en:

- **Retener** → Vacas con buena eficiencia, baja incidencia de problemas y mérito productivo alto.
- **Supervisar** → Vacas con desempeño medio o inestable que pueden mejorar con ajustes operativos.
- **Descartar** → Vacas que generan costos, ineficiencia o baja producción intrínseca.

2.0 Selección de técnica de modelado

2.1 Aproximación inicial del modelo (descartada)

En nuestra primera iteración de modelado optamos por hacer un enfoque de “imputación masiva” de datos seguido de un modelado de datos clásico:

1. Para nuestro preprocesamiento utilizamos el script de *generate_data.py* para llenar valores faltantes en flujos y tiempos utilizando medias móviles históricas por vaca
2. Los modelos que propusimos fueron:
 - a. Random Forest Classifier para predecir “Patadas” basado en series de tiempo completas
 - b. Isolation Forest aplicado sobre el dataset imputado para detectar salud

2.2 Evaluación de inviabilidad (Causa del Pivote a la nueva técnica)

Tras la primera iteración del proyecto, nos dimos cuenta que esta primera iteración el enfoque no era viable, llevando a su descarte por las siguientes razones:

1. Volumen de muestra insuficiente: El obstáculo más determinante fue la disponibilidad limitada de sujetos de estudio que en este caso son las vacas del CAETEC. El dataset que recibimos contaba con un total de registros históricos provenientes únicamente de 30 vacas únicas.
 - a. El impacto técnico fue tal porque al darnos cuenta que los algoritmos de aprendizaje supervisado como lo es el Random Forest que tienen una n de 30 es estadísticamente insuficiente para realizar una partición robusta de *Train/Test/Val*.

- b. La consecuencia de no tener la suficiente cantidad de registros es que nuestro modelo no logra “aprender” características generales, sino que solamente memoriza el comportamiento de esas 30 vacas, haciendo que nuestro sistema solamente funcione de manera correcta con las muestras de las vacas que ya tenemos pero no con las vacas nuevas o futuras.
2. Sesgo por imputación: La estrategia que llevamos a cabo de llenar vacíos con el promedio histórico de la propia vaca redujo artificialmente la varianza, al tener tan pocas vacas, la imputación terminó por homogeneizar los datos de tan manera que se llegan a perder señales sutiles de estrés o enfermedad, que son clave para determinar la sanidad de la vaca que va de la mano con la producción de leche de las mismas.

Como equipo decidimos descartar esta idea dado la restricción que contábamos de solamente 30 vacas, se concluye que es imposible entrenar un modelo predictivo general que sea fiable bajo la arquitectura inicial. Se decide pivotar hacia una iteración basada en detección de anomalías no supervisadas y estadística descriptiva robusta.

La selección de vacas **no debe basarse solo en la producción observada**, porque esta está influida por factores ambientales como:

- La hora del ordeño,
- La temperatura y estrés térmico,
- La congestión en la sala.
- El manejo operativo diario.

Si seleccionamos vacas únicamente por producción observada, correríamos el riesgo de premiar a vacas que producen bien sólo bajo condiciones favorables, y descartar vacas que producen bien aun en situaciones adversas, lo cual no refleja mérito genético ni valor real a largo plazo.

Por esta razón, la propuesta inicial separa:

Dimensión	Tipo de modelo	¿Qué mide?	Por qué es importante
Comportamiento	Modelo Supervisado	Eficiencia y estabilidad operativa durante el ordeño	Reduce tiempos muertos y pérdidas
Sanidad (proxy)	Modelo No Supervisado	Probabilidad de eventos de mastitis o complicaciones de ubre	Evita costos futuros y descarte de leche
Mérito productivo ajustado	Ajuste estadístico (no modelo)	Capacidad productiva propia de la vaca sin efecto del ambiente	Determina valor genético y retención

2.1 Técnicas seleccionadas y justificación

2.1.1 Modelo de comportamiento

Propósito

Identificar vacas que dificultan el proceso de ordeño y generan ineficiencia en el robot.

Construcción del Target

El comportamiento inquieto se define de forma operacional, usando eventos reales:

$Inquieta = (Patada = 1)OR(Incompleteo = 1)OR(PezonesNoRncontrados > 0)OR(Duracion > P95)$
Modelo selecciona

Resultado

Para cada vaca:

- Probabilidad promedio de inquietud
- Porcentaje de sesiones problemáticas

Esto se convertirá en:

$$RiesgoComportamiento_i$$

Modelos seleccionados

Versión	Algoritmo seleccionado	Justificación	Iteración asociada
LR Baseline	Regresión Logística	- Baseline sin tuning para medir mejora posterior. - Simple, interpretable y rápido. - Sirve como punto cero para comparación con RF y MLP. - Bajo recall y F1. No apto para despliegue.	1
RF v1.0	Random Forest - dataset 1.0	- Primer modelo considerado. - Manejo robusto de no-linealidad. - Buen accuracy, pero recall insuficiente. - Identifica patrones, pero aún sub-detecta inquietas.	2

RF v2.0	Random Forest - dataset 2.0	El objetivo de la versión v2.0 no fue optimizar hiper parámetros, sino corregir la base de datos de entrenamiento manteniendo la misma lógica algorítmica, permitiendo medir el impacto puro de la mejora en calidad de datos. Esta versión representa el punto de inflexión entre desempeño limitado por datos y desempeño mejorado por estructura y volumen del dataset.	3
RF v2.1	Random Forest 2.1 GridSearchCV	- Ajuste sistemático de hiper parámetros. - Mejor desempeño global en todas las métricas. - Balance ideal entre precisión y F1. - Versión más madura para uso operativo	4
MLP	Red Neuronal MLP v1.0	- Excelente para detectar inquietas (mayor recall). - Aprendizaje no lineal más profundo. - Sensible → detecta más positivos incluso si hay baja precisión. - Preferible cuando la prioridad es no dejar vacas sin revisar. - Punto de referencia para comparar los resultados con los modelos de RF.	5

2.1.2 Modelo de sanidad

Propósito

Identificar vacas con un riesgo latente de tener condiciones de salud como eventos de mastitis o complicaciones de ubre.

¿Por qué un modelo no supervisado?

- El **RCS** está **vacío** en los registros.
- Los eventos sanitarios son **poco frecuentes y ruidosos**.
- No existe un **label confiable** para entrenar un clasificador supervisado.

Resultados

$$RiesgoSanitario_i = \text{media}(\text{ScoreAnomalia})$$

Modelos seleccionados

Versión	Algoritmo seleccionado	Justificación	Iteración asociada
DBSCAN	Clustering no supervisado	- Detecta grupos densos y ruido como anomalías. - Sin supuestos estadísticos. Útil en sanidad real. - Alta interpretabilidad visual vía PCA. - Sirve como referencia base para métodos futuros	1
Isolation Forest v1.0	Isolation Forest - dataset 1.0	- Detecta anomalías raras sin etiquetas. - Ideal para datos tabulares de salud (numeros+binarios). - Da score continuo de riesgo sanitario. - Identifica patrones inusuales en DI/DD/TI/TD. - Limitación: mayor variación entre folds y % anomalías.	2
Isolation Forest v2.0	Isolation Forest - dataset 2.0	El objetivo de la versión v2.0 no fue optimizar hiperparámetros, sino corregir la base de datos de entrenamiento manteniendo la misma lógica algorítmica, permitiendo medir el impacto puro de la mejora en calidad de datos. Esta versión representa el punto de inflexión entre desempeño limitado por datos y desempeño mejorado por estructura y volumen del dataset.	3
Isolation Forest v2.1	Isolation Forest Optimizado	- Mantiene beneficios de la versión anterior. - Usa One-Hot Encoding y preprocessamiento más limpio. - CLI configurable + autodetección de tipos. - Column Transformer + pipelines separados para num+cat. - Versión más estable y con mejor calibración (5%).	4

2.1.3 Mérito productivo

Propósito

Estimar qué tan productiva es la vaca por sí misma, independientemente del ambiente.

Ajuste

$$ProduccionAjustada_i = ProduccionObservada_i - \bar{x}(Produccion | robot x hora x mes)$$

Donde:

$i = \text{Sesión de ordeño}$

Luego

$$\text{MeritoProductivo}_i = \frac{i}{N_i} \sum_{j=1}^{N_i} \text{ProduccionAjustada}_{ij}$$

Donde:

$i = \text{vaca}$

$j = \text{sesión de ordeño de esa vaca}$

$N_i = \text{número de ordeños que tiene esa vaca en el período}$

Esto representa **valor genético operativo**, no desempeñó circunstancial.

2.1.4 Integración en el Índice de Mérito para la Retención (IMR)

$$IMR_i = w_G \cdot Z(\text{MeritoProductivo}_i) - w_C \cdot Z(\text{RiesgoComportamiento}_i) - w_S \cdot Z(\text{RiesgoSanidad}_i)$$

Donde:

U

$w_G = \text{peso del mérito productivo (peso seleccionado: 0.5)}$

$w_C = \text{peso del riesgo de comportamiento (peso seleccionado: 0.2)}$

$w_S = \text{peso del riesgo sanitario (peso seleccionado: 0.3)}$

Es importante mencionar que los pesos seleccionados son en base a lo que tenemos más certeza o datos certeros en nuestros modelos, dando mayor influencia a el modelo de comportamiento dados los datos que nos confirman con mayor precisión el comportamiento de una vaca. Estos pesos se pueden ajustar si se quiere dar más influencia o menos a alguna de las dimensiones.

Luego:

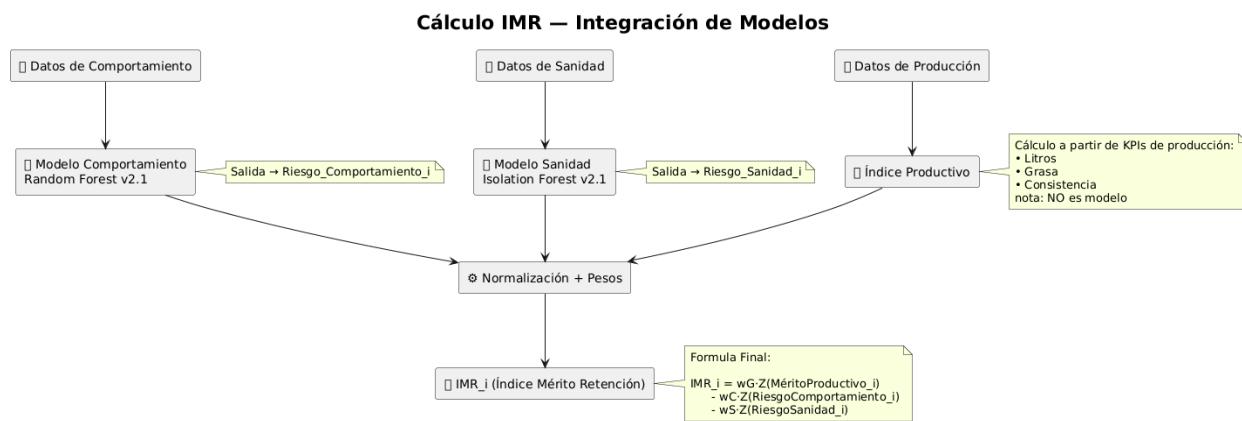
Rango del IMR (basado en percentiles poblacionales)	Condición estadística	Decisión recomendada	Interpretación operativa
$IMR \geq p75$	Top 25% del hato	Retener / Reproducir	Alta prioridad genética y productiva — candidata para inseminación o mantenimiento prolongado.

p40 ≤ IMR < p75	Segmento medio (p40–p75)	Supervisar / Manejo dirigido	Vacas de comportamiento intermedio — conservar, monitorear y optimizar manejo antes de decidir reproducción o descarte.
IMR < p40	Bottom 40% del hato	Descartar / Secado / Venta	Bajo desempeño relativo — adecuada para retiro, secado o priorizar reemplazo.

3.0 Pipeline

El pipeline que sigue nuestra propuesta está compuesta por las dimensiones previamente mencionadas de las cuales las dimensiones de comportamiento y sanidad se resuelven con modelos de machine learning y la dimensión de producción por medio de la fórmula en el capítulo 2.1.3. En conjunto estas 3 dimensiones generan un índice que en base a los rangos estipulados se decide si se conserva la vaca para reproducción/producción, si se debe mantener bajo observación o si se descarta/manda a secado.

La estructura que sigue la solución es la siguiente:



4.0 Dataset

Como se mencionó anteriormente en esta solución se utilizaron 2 datasets.

- v1.0: Dataset a partir de *registros_sesiones_merged* (archivo con primera transformación de datos donde se unen todas las sesiones), este dataset quedó descartado tras los resultados de la iteración 1, dadas la cantidad reducida de las instancias.
- v2.0: Dataset a partir de *registros_sesiones_merged*, este dataset aprovecha las 7 mil instancias de los datos originales, esta segunda transformación de datos genera un dataset correspondiente para cada modelo:

Dataset	Entradas finales	Objetivo
sessions_behavior.csv	15 features numéricas + label_inquieta	Para predecir Comportamiento.
sessions_health.csv	20 features numéricas	Para detectar anomalías sanitarias.

Para más detalle se puede consultar el documento de la fase de preparación de datos, haciendo [click aquí](#).

5.0 División de datos

Para cada dataset utilizado en cada uno de los modelos, se hizo una división **K-Fold** con uso del método **K-Fold** parte de la clase de *model_selection* de la librería *Sklearn*. El número de splits fueron los siguientes:

Modelo	Versión	Número de Folds
Random Forest	Modelo base	3
	V1.0	No aplica
	V2.0	3
	V2.1	3
	MLP v1.0	3
Isolation Forest	Modelo base	3
	V1.0	3
	V2.0	3
	V2.1	5

6.0 Pruebas

6.1 Métricas de evaluación

Dado que son diferentes modelos, se establecerá a continuación una tabla sobre las métricas de evaluación para cada modelo según su versión o arquitectura.

Modelo	Versión	Metrica de evaluación	Justificación
RF (Comportamiento)	modelo base, v1.0, v2.0, v2.1, red neuron al MLP v1.0	Accuracy	Proporciona una visión general del desempeño global del modelo y permite comparar distintas versiones rápidamente.
		Precision	Mide qué proporción de las predicciones positivas (inquieta) fueron correctas, minimizando falsos positivos (evita alertas innecesarias en vacas que no están inquietas).
		Recall	Mide la proporción de vacas inquietas reales detectadas por el modelo, minimizando falsos negativos (evita perder casos de inquietud que requieren atención).
		F1 Score	Combina precision y recall mediante su media armónica. Es especialmente útil ante desbalance de clases y evita depender únicamente del accuracy.
Isolation Forest (Sanidad)	v1.0, v2.0, v2.1	% de anomalías por fold	Al no existir etiquetas reales, el modelo se evalúa midiendo la proporción de anomalías detectadas en cada partición. Permite evaluar la estabilidad del modelo.
		% de anomalías global	Mide el porcentaje total de anomalías generadas por la ejecución final. Permite validar si la proporción detectada es coherente con el parámetro contamination y con el comportamiento esperado del sistema.

6.2 Criterios de evaluación

De igual manera, a continuación se presentan los criterios de evaluación establecidos para cada modelo y coherencia del contexto de cada modelo seleccionado.

Modelo	Métrica	Criterio aceptable	Criterio ideal
Comportamiento	Accuracy	≥ 0.80	$\geq 0.85\text{--}0.90$
	Precision	≥ 0.75	$\geq 0.80\text{--}0.85$

	Recall	≥ 0.70	≥ 0.80
	F1 Score	≥ 0.72	≥ 0.80
Sanidad	% anomalías por fold	diff ≤ 5 puntos	diff ≤ 3 puntos
	% anomalías global	4%–7%	$\approx 5\%$
	Histograma score	cola visible	outliers muy definidos

6.3 Artefactos a generar

6.3.1 Modelo de comportamiento - Random Forest

- Tabla de métricas.
- Gráficas de métricas
- Gráfica de líneas por fold.
- Matriz de confusión.

6.3.2 Modelo sanidad - Isolation Forest

- Gráfica de anomalías por fold.
- Histograma del anomaly score.
- Pipeline final guardado.

6.4 Plan de evaluación de modelos

6.3.1 Modelo de Comportamiento - Random Forest

6.3.1.1 División de datos

Aplicar el esquema de validación cruzada estratificada (3-folds), asegurando que la proporción de clases se mantenga en cada partición.

6.3.1.2 Ejecutar modelo

Ejecutar el modelo aplicado a todos los datos tras la estrategia de estratificación y folds.

6.3.1.3 Metricas a evaluar

Generar las métricas y artefactos establecidos para evaluar el modelo:

- Accuracy.
- Precision.
- Recall.
- F1 Score.

6.3.1.4 Analizar los resultados

- Comparar los valores obtenidos contra los criterios establecidos en la sección 4.2.
- Analizar posibles indicios de sobreajuste.
- Revisar artefactos generados (gráficas y tablas) para validar coherencia.

6.3.2 Modelo de Sanidad - Isolation Forest

6.3.2.1 División de datos

Aplicar validación mediante K-Fold simple (3 folds), adecuado para tareas no supervisadas.

6.3.2.2 Ejecutar modelo

Entrenar el Isolation Forest en cada fold y generar predicciones en el fold de prueba correspondiente.

6.3.2.3 Metricas a evaluar

Generar las métricas definidas en el apartado 4.1:

- % de anomalías por fold.
- % global de anomalías.
- Distribución del anomaly_score (histograma).

6.3.2.4 Analizar los resultados

- Comparar los porcentajes con los criterios establecidos en la sección 4.2.
- Evaluar la estabilidad del modelo entre folds.
- Revisar artefactos.

7.0 Construir el modelo

7.1 Selección de librerías

Este proyecto incorpora un conjunto de librerías diseñadas para optimizar el proceso de modelado, validación y detección de patrones en datos de comportamiento y sanidad bovina.

El núcleo del sistema se construyó con **scikit-learn**, debido a que ofrece algoritmos maduros, reproducibles y eficientes para análisis tabular, además de contar con herramientas integradas para imputación, escalado, pipelines, validación cruzada y serialización de modelos. Esto permitió desarrollar múltiples versiones de modelos de forma iterativa, manteniendo un entorno controlado para comparar mejoras entre versiones y habilitando una curva de desarrollo rápida.

Adicionalmente, para extender la capacidad predictiva más allá de modelos basados en árboles, se incorporó **TensorFlow/Keras** en la implementación del modelo **MLP para comportamiento**, permitiendo explorar redes neuronales densas con mayor expresividad y capacidad para capturar relaciones no lineales profundas en los datos. Esta decisión abre la puerta a futuras extensiones basadas en autoencoders para sanidad, redes densas más profundas o modelos híbridos de clasificación y anomalías.

7.2 Modelo de comportamiento base (Regresión Logística)

7.2.1 Descripción

El modelo de Regresión Logística sirve como línea base (baseline) para la clasificación de comportamiento, prediciendo si una sesión de ordeño corresponde a una vaca "inquieta" (`label_inquieta`) o no. Este modelo lineal permite establecer un punto de comparación con modelos más complejos como Random Forest y MLP.

El pipeline aplicado es:

1. Imputación de valores faltantes con `SimpleImputer(strategy="median")`, para manejar ausencias en los registros numéricos de forma robusta ante outliers.
2. Escalado de las variables con `StandardScaler()`, necesario para que los coeficientes de la regresión logística sean comparables y el modelo converja correctamente.
3. Clasificación con `LogisticRegression`, un modelo lineal que estima probabilidades mediante la función sigmoide, configurado con pesos balanceados para manejar el desbalance de clases.

El desempeño del modelo se evalúa mediante validación cruzada estratificada de 3 folds, utilizando como métricas principales: accuracy, precision, recall y F1-score, además de la desviación estándar entre folds para analizar estabilidad

7.2.2 Configuración de parámetros

Parámetro	Valor	Justificación
<code>class_weight</code>	"balanced"	Ajusta pesos automáticos para compensar desbalance entre clases.
<code>max_iter</code>	1000	Asegura convergencia del optimizador en entrenamiento.

random_state	42	Garantiza la reproducibilidad del modelo.
--------------	----	---

7.2.3 Codificación

El modelo se encuentra disponible en el [repositorio del proyecto](#). En el directorio ./models/comportamiento_lr.py

7.3 Modelo de sanidad base (Density Clustering)

7.3.1 Descripción

El modelo de sanidad utiliza DBSCAN (Density-Based Spatial Clustering of Applications with Noise) para detectar sesiones de ordeño anómalas de forma no supervisada. A diferencia de los modelos de clasificación, este enfoque no requiere etiquetas previas y detecta anomalías como puntos de baja densidad (ruido) en el espacio de características.

El pipeline aplicado es:

1. **Imputación de valores faltantes** con SimpleImputer(strategy="median"), para manejar ausencias en los registros numéricos de forma robusta ante outliers.
2. **Escalado de las variables** con StandardScaler(), fundamental para algoritmos basados en distancia como DBSCAN.
3. **Cálculo automático del parámetro eps** mediante el método del codo (k-distance graph), utilizando el percentil 90 de las distancias al k-ésimo vecino más cercano.
4. **Clustering con DBSCAN**, que agrupa puntos en regiones densas y clasifica puntos aislados como ruido (anomalías).
5. **Cálculo de anomaly scores** basados en la distancia media a los k vecinos más cercanos, donde mayor distancia indica mayor grado de anomalía.

La estabilidad del modelo se evalúa mediante validación cruzada de 6 folds, midiendo la consistencia de la tasa de anomalías detectadas entre particiones. Para asignar etiquetas a nuevos datos, se utiliza un método basado en proximidad a los puntos core del clustering original.

7.3.2 Configuración de parámetros

Parámetro	Valor	Justificación
Modelo	DBSCAN	Algoritmo de clustering basado en densidad, adecuado para detectar anomalías como puntos de baja densidad (ruido).

min_samples	5	Número mínimo de puntos para definir una región densa; balancea sensibilidad frente a ruido.
eps	percentil 90 de k-distances (k = min_samples)	Se calcula automáticamente con el método del codo sobre distancias k-NN, ajustando el radio de vecindad según la estructura de los datos.
Preprocesamiento/ imputación	SimpleImputer(strategy="median")	Rellena valores faltantes de forma robusta ante outliers numéricos.
Preprocesamiento/ escalado	StandardScaler()	Estandariza las variables para que las distancias usadas por DBSCAN sean comparables entre features.
Cálculo de vecinos	NearestNeighbors(n_neighbors=min_samples)	Se utiliza tanto para estimar eps (k-distances) como para calcular anomaly scores.
Anomaly score	media de distancias a k vecinos	Define qué tan aislado está cada punto; mayor distancia implica mayor rareza/sospecha de anomalía.
Conversión de labels	cluster_labels == -1 → -1 (anomalía), else 1 (normal)	Adapta la convención de DBSCAN (ruido = -1) a un esquema binario estándar de anomalía.

7.3.3 Codificación

El modelo se encuentra disponible en el [repositorio del proyecto](#). En el directorio
`./models/sanidad_clustering_base.py`

7.4 Modelo de comportamiento v1.0

7.4.1 Descripción

El modelo Random Forest v1.0 corresponde a la primera versión supervisada utilizada para predecir el **comportamiento de las vacas durante el ordeño**, específicamente identificar si una sesión pertenece a una vaca inquieta o tranquila.

La lógica del modelo se basa en un conjunto de árboles de decisión que aprenden patrones desde múltiples variables del proceso (tiempos, producción, rutinas, actividad, etc.) para inferir el comportamiento esperado.

Al combinar muchos árboles y aplicar votación, se busca capturar relaciones no lineales y reducir el riesgo de decisiones basadas en una sola partición simple del espacio de datos.

Esta versión se enfoca en servir como modelo exploratorio y línea base, permitiendo:

- Identificar qué variables parecen relacionarse con la inquietud.
- Evaluar la factibilidad de predecir comportamiento
- Establecer un punto de comparación para versiones posteriores más refinadas.

Su implementación incluye validación cruzada, exportación del modelo y generación de artefactos visuales, posicionándose como el primer clasificador formal del proyecto.

7.4.2 Configuración de parámetros

Parámetro	Valor	Justificación
Estimador	RandomForestClassifier	Algoritmo de ensamble adecuado para relaciones no lineales en datos tabulares.
n_estimators	200	Aumenta la estabilidad del bosque sin costo computacional excesivo.
max_depth	10	Controla la complejidad del modelo y reduce el sobreajuste.
min_samples_split	5	Evita splits con muy pocos datos, mejorando la generalización.
min_samples_leaf	2	Previene hojas unitarias y suaviza las fronteras de decisión.
class_weight	"balanced"	Compensa el desbalance entre clases ajustando el peso de etiquetas.
criterio de etiqueta	10% peor puntaje → clase positiva	Traduce la variable continua de ranking a clasificación binaria.
Validación	StratifiedKFold (3 folds, shuffle=True)	Mantiene proporciones de clases y evalúa la estabilidad del modelo.

random_state	42	Para la reproducibilidad del experimento.
n_jobs	-1	Utiliza todos los núcleos disponibles para acelerar el entrenamiento.

7.4.3 Codificación

El modelo se encuentra disponible en el [repositorio del proyecto](#). En el directorio
./models/comportamiento_rf_v1.py

7.5 Modelo de sanidad v1.0

7.5.1 Descripción

El Isolation Forest v1.0 representa el primer esfuerzo no supervisado para detectar **sesiones de ordeño con posibles problemas sanitarios o comportamientos anómalos del sistema físico/biológico**.

A diferencia de un clasificador, este modelo no necesita etiquetas: identifica automáticamente vacas o sesiones que se comportan de manera distinta al resto.

El algoritmo funciona aislando observaciones que requieren pocas particiones para separarse del conjunto, lo que suele corresponder a eventos raros — como irregularidad en tiempos de ordeño, producción alterada o señales de estrés/mala salud.

Este modelo inicial se diseñó como un **sensor temprano de riesgo**, útil para:

- Detectar outliers que ameritan revisión.
- Comprender el rango operacional “normal”.
- Sentar la base para futuras versiones más robustas con más datos y mejor preprocesamiento.

5.5.2 Configuración de parámetros

Parámetro	Valor	Justificación
Modelo	IsolationForest	Algoritmo no supervisado eficaz para identificar registros raros o atípicos.
n_estimators	100	Cantidad suficiente para estabilidad en scores sin alto uso computacional.
contamination	0.10	Asume que aproximadamente el 10 % de las sesiones podrían ser anómalas.

Imputación	SimpleImputer (mediana)	Manejo robusto de valores faltantes numéricos frente a outliers.
Escalado	StandardScaler	Normaliza las variables evitando sesgo por diferencias de escala.
Validación	KFold (6 folds, shuffle=True)	Evaluá la estabilidad de la tasa de anomalías entre particiones.
PCA (visualización)	2 componentes	Permite inspección visual de separación entre normales y outliers.
Umbral de score	percentil 90 (coherente con contamination=0.10)	Define el top 10 % de rareza como región anómala.
random_state	42	Permite la reproducibilidad de resultados.
n_jobs	-1	Utiliza todos los núcleos para entrenamiento y scoring eficiente.

7.5.3 Codificación

El modelo se encuentra disponible en el [repositorio del proyecto](#). En el directorio ./models/sanidad_iso_v1.py

7.6 Modelo comportamiento v2.0

7.6.1 Descripción

La versión 2.0 del modelo de comportamiento mantiene el enfoque supervisado para predecir si una sesión de ordeño corresponde a una vaca “inquieta” (label_inquieta), pero sustituye por completo el dataset de entrenamiento utilizado en la versión 1.0.

Mientras que la versión 1.0 se entrenó sobre un conjunto reducido de alrededor de 30 sesiones (dataset 1.0), la versión 2.0 aprovecha el dataset transformado 2.0, con aproximadamente 7 000 instancias de sesiones de ordeño. Las etapas de limpieza, selección de variables y construcción de este nuevo dataset se describen en detalle en el documento de Fase de Transformación de Datos, que aplica para todos los modelos a partir de la versión 2.0.

7.6.2 Configuración de parámetros

Parámetro	Valor	Justificación

Estimador	RandomForestClassifier	Modelo de ensamble adecuado para datos tabulares y relaciones no lineales.
n_estimators	200	Aumenta la estabilidad del bosque sin un costo computacional excesivo.
max_depth	8	Limita la complejidad de los árboles para reducir sobreajuste en ~7k instancias.
min_samples_leaf	5	Evita hojas muy pequeñas, mejorando la capacidad de generalización.
class_weight	"balanced"	Compensa el desbalance de <code>label_inquieta</code> ajustando los pesos de las clases.
random_state	42	Garantiza la reproducibilidad del modelo.
n_jobs	-1	Usa todos los núcleos disponibles para acelerar el entrenamiento.

7.6.3 Codificación

El modelo se encuentra disponible en el [repositorio del proyecto](#). En el directorio
./models/sanidad_rf_v2.py

7.7 Modelo sanidad v2.0

7.7.1 Descripción

La versión 2.0 marca la primera integración formal del algoritmo Isolation Forest para la detección no supervisada de sesiones anómalas.

A diferencia del enfoque inicial en DBSCAN, en esta etapa se adoptó un modelo basado en árboles por su capacidad para escalar mejor con dimensionalidad elevada y su estabilidad frente a ruido y valores extremos.

Un cambio determinante en esta versión respecto a la versión 1.0 fue la implementación del dataset v2.0 de sanidad, incorporando todas las instancias disponibles. Esto permitió construir un detector más representativo del entorno real de producción, reduciendo sesgos y aumentando la cobertura del espectro saludable - anómalo.

7.7.2 Configuración de parámetros

Parámetro	Valor	Justificación
Modelo	IsolationForest	Algoritmo de detección de anomalías basado en aislamiento, adecuado para datos tabulares sin etiquetas.
n_estimators	200	Suficientes árboles para estabilidad en los scores sin disparar el costo computacional.
contamination	0.05	Se asume que ~5% de las sesiones son anómalas; fija el umbral de decisión interno del modelo.
random_state	42	Asegura reproducibilidad en la construcción de los árboles y en los resultados.
n_jobs	-1	Utiliza todos los núcleos disponibles para acelerar el entrenamiento y scoring.
Imputación	SimpleImputer(strategy="median")	Maneja valores faltantes numéricos de manera robusta frente a outliers.
Escalado	StandardScaler()	Estandariza las variables para que el modelo no se vea sesgado por diferencias de escala.
Validación no supervisada	KFold(n_splits=6, shuffle=True, random_state=42)	Usa 6 pliegues estratificados en el espacio de features para evaluar la estabilidad de la tasa de anomalías.
PCA para visualización	PCA(n_components=2)	Proyecta el espacio de características a 2D para inspección visual de normales vs anomalías.
Umbral de histograma	contamination = 0.05 + percentile(95) de anomaly_scores	Coherente con el parámetro contamination del modelo; marca el top 5% como outliers más extremos en el histograma.

7.7.3 Codificación

El modelo se encuentra disponible en el [repositorio del proyecto](#). En el directorio
./models/sanidad_iso_v2.0.py

7.8 Modelo comportamiento v2.1

6.8.1 Descripción

La versión 2.1 del modelo de comportamiento se construye sobre la misma base conceptual que la v2.0: un clasificador supervisado de tipo Random Forest entrenado sobre el dataset transformado 2.0 para predecir la etiqueta binaria label_inquieta. La diferencia clave es que la v2.1 incorpora una fase explícita de búsqueda de hiperparámetros (GridSearchCV) enfocada en maximizar el F1-score, con el objetivo de mejorar la capacidad del modelo para detectar vacas problemáticas sin sacrificar demasiado la precisión.

Las transformaciones de datos previas (limpieza, construcción de variables y generación del dataset 2.0) son las mismas descritas en la Fase de Transformación de Datos, compartidas por todos los modelos a partir de la versión 2.0.

7.8.2 Configuración de parámetros

Parámetro	Valor	Justificación / Cambio respecto v2.0
Estimador base	RandomForestClassifier	Se mantiene el mismo tipo de modelo que en v2.0 (bosque aleatorio para clasificación).
random_state (RF)	42	Igual que en v2.0, asegura la reproducibilidad del entrenamiento.
n_jobs (RF)	-1	Igual que en v2.0, aprovecha todos los núcleos disponibles.
Búsqueda de hiperparámetros	GridSearchCV	Mejora vs v2.0: en lugar de fijar manualmente los hiperparámetros, se optimizan automáticamente.
Espacio rf_n_estimators	[100, 200, 300]	Explora distintos números de árboles para ajustar estabilidad vs costo; v2.0 fijaba 200.
Espacio rf_max_depth	[None, 10, 20]	Permite comparar árboles sin límite de profundidad y con profundidad acotada; v2.0 fijaba 8.
Espacio rf_min_samples_split	[2, 5]	Añade control sobre el mínimo de muestras para dividir un nodo (no estaba explícito en v2.0).

Espacio rf_min_samples_leaf	[1, 2, 4]	Explora hojas más pequeñas y más grandes frente al valor fijo 5 de v2.0.
Espacio rf_class_weight	["balanced", None]	Compara el uso de pesos balanceados vs no balanceados; v2.0 solo usaba "balanced".
Métrica de optimización	scoring="f1"	Mejora vs v2.0: se optimiza explícitamente el F1-score para balancear precisión y recall en "inquieta".
Esquema interno de CV	cv=3	Usa validación cruzada estratificada de 3 folds dentro de GridSearch para seleccionar la mejor configuración.

7.8.3 Codificación

El modelo se encuentra disponible en el [repositorio del proyecto](#). En el directorio
./models/comportamiento_rf_v2.1.py

7.9 Modelo sanidad v2.1

7.9.1 Descripción

La versión 2.1 extiende y refina la arquitectura anterior. Su principal mejora consiste en un pipeline más flexible, preparado para datos mixtos numéricos + categóricos mediante ColumnTransformer, lo que permite incluir información cualitativa sin perder consistencia en el entrenamiento. Además, parámetros clave como n_estimators, contamination.

7.9.2 Configuración de parámetros

Parámetro	Valor	Justificación / Cambio respecto v2.0
Modelo	IsolationForest	Se mantiene el mismo algoritmo de aislamiento para detección de anomalías en sanidad; la mejora viene del preprocesamiento y la configurabilidad, no del tipo de modelo.

n_estimators	args.n_estimators (por defecto 200)	Permite ajustar el número de árboles desde CLI según estabilidad vs costo computacional. En v2.0 estaba fijado a 200 sin posibilidad de modificación.
contamination	args.contamination (por defecto 0.05)	Expone la proporción esperada de outliers como hiperparámetro configurable. En v2.0 el valor 0.05 estaba hard-coded dentro del modelo y en el cálculo del umbral del histograma.
random_state	42	Se mantiene el uso de una semilla fija para asegurar reproducibilidad de resultados, igual que en v2.0.
n_jobs	-1	Igual que en v2.0, usa todos los núcleos disponibles para acelerar entrenamiento y scoring.
Argumento --k_folds	args.k_folds (por defecto 5)	Permite controlar el número de pliegues para la verificación de estabilidad de anomalías. En v2.0 n_splits estaba fijado a 6 y no podía ajustarse desde línea de comandos.
Preprocesamiento numérico	SimpleImputer(strategy="median") + StandardScaler() dentro de un Pipeline	Mantiene imputación robusta y estandarización para variables numéricas, como en v2.0, pero ahora encapsulado dentro de un ColumnTransformer reutilizable y extensible.
Preprocesamiento categórico	SimpleImputer(strategy="constant", fill_value="missing") + OneHotEncoder(handle_unknown="ignore", sparse_output=False)	Mejora: permite incorporar variables categóricas de sanidad de forma segura, ampliando el modelo a datos mixtos (numéricos + categóricos). En v2.0 solo se trabajaba con columnas numéricas.
Combinación de features	ColumnTransformer([("num", numeric_transformer, numeric_features), ("cat", categorical_transformer, categorical_features)])	Mejora estructural: unifica el preprocesamiento numérico y categórico en un solo bloque. En v2.0 solo había una cadena imputer+scaler plana sin separar tipos de variables.

Heurística de limpieza	Eliminación de columnas que contienen "id" o "date"	Nuevo: se filtran automáticamente columnas identificadoras/temporales que podrían introducir ruido. v2.0 no aplicaba esta heurística.
Validación no supervisada	KFold(n_splits=args.k_folds, shuffle=True, random_state=42)	Mantiene el esquema de K-Fold para medir estabilidad de la tasa de anomalías, pero ahora con número de pliegues configurable (v2.0 usaba 6 pliegues fijos).

7.8.3 Codificación

El modelo se encuentra disponible en el [repositorio del proyecto](#). En el directorio ./models/sanidad_iso_v2.1.py

7.10 Modelo de comportamiento MLP v1.0.

7.10.1 Descripción

El modelo de comportamiento MLP utiliza una red neuronal Multi-Layer Perceptron para predecir si una sesión de ordeño corresponde a una vaca inquieta (label_inquieta) o no. A diferencia del modelo Random Forest, este enfoque emplea aprendizaje profundo con optimización de umbral basada en F1-score.

El pipeline aplicado es:

1. Imputación de valores faltantes con SimpleImputer(strategy="median"), para manejar ausencias en los registros numéricos de forma robusta ante outliers.
2. Escalado de las variables con StandardScaler(), fundamental para el correcto funcionamiento de redes neuronales.
3. Clasificación con MLP de arquitectura secuencial: capa densa de 64 neuronas con ReLU, Dropout del 20%, capa densa de 32 neuronas con ReLU, y capa de salida sigmoide.
4. Optimización del umbral de clasificación mediante búsqueda exhaustiva maximizando F1-score sobre el conjunto de entrenamiento.

El desempeño se evalúa mediante validación cruzada estratificada de 3 folds, con métricas de accuracy, precisión, recall y F1-score. El modelo final se entrena con 40 épocas y pesos balanceados para compensar el desbalance de clases.

7.10.2 Configuración de parámetros

Parámetro	Valor	Justificación
-----------	-------	---------------

Arquitectura capa oculta 1	Dense(64, activation="relu")	Primera capa densa con suficiente capacidad para capturar patrones no lineales en los features.
Regularización	Dropout(0.2)	Apaga aleatoriamente el 20% de las neuronas durante el entrenamiento para reducir el sobreajuste.
Arquitectura capa oculta 2	Dense(32, activation="relu")	Segunda capa que refina la representación y reduce progresivamente la dimensionalidad.
Capa de salida	Dense(1, activation="sigmoid")	Produce una probabilidad para la clase "inquieta" en un problema de clasificación binaria.
Función de pérdida	binary_crossentropy	Pérdida estándar para tareas de clasificación binaria con salida sigmoide.
Optimizador	Adam	Optimizador adaptativo robusto para este tipo de redes y datasets tabulares.
batch_size	32	Compromiso entre estabilidad del gradiente y velocidad de entrenamiento.
Épocas (validación cruzada)	20	Número de épocas reducido para acelerar la evaluación en CV sin sobreentrenar.
Épocas (entrenamiento final)	40	Más épocas para que el modelo definitivo aproveche mejor los datos de entrenamiento.
class_weight	balanced (calculado)	Ajusta la contribución de cada clase en la pérdida para compensar el desbalance de label_inquieta.
Búsqueda de umbral	threshold $\in \{0.1, 0.2, \dots, 0.8\}$	Se exploran distintos umbrales de decisión sobre la probabilidad para maximizar el F1-score.
Criterio de selección de umbral	argmax(F1-score)	El umbral final no es 0.5 fijo, sino el que ofrece mejor balance precisión/recall.

7.10.3 Codificación

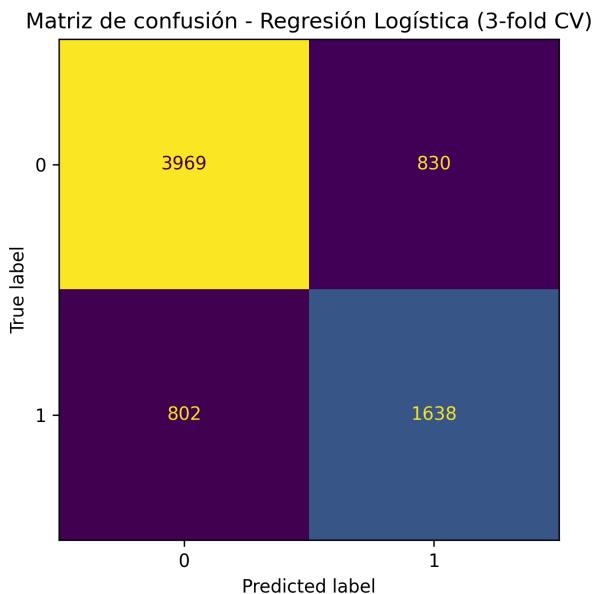
El modelo se encuentra disponible en el [repositorio del proyecto](#). En el directorio
./models/comportamiento_mlp_th03_v3.py

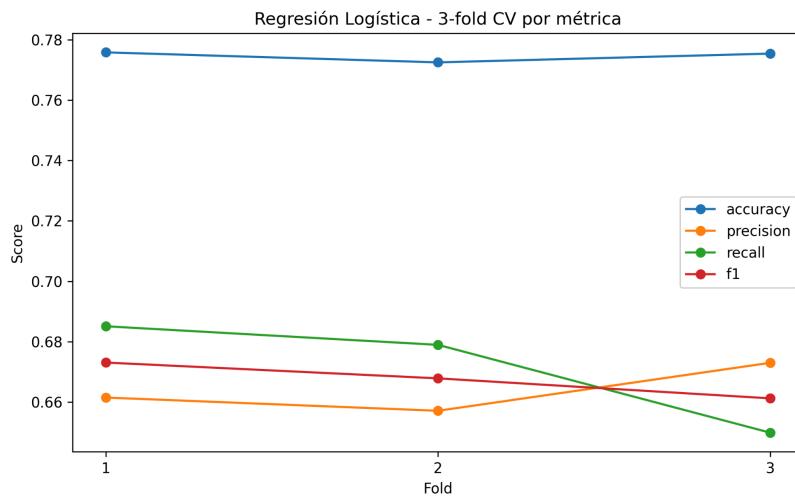
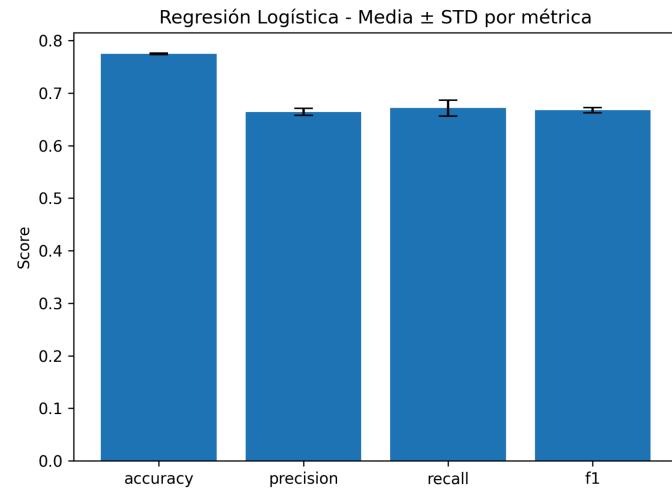
8.0 Evaluación

8.1 Evaluación modelo de comportamiento base regresión logística

8.1.1 Resultados

Métrica	Fold 1	Fold 2	Fold 3	Media	Desviación estándar
Accuracy	0.7757977621	0.7724823871	0.7753833402	0.7745544965	0.0014749383
Precision	0.6615201900	0.6571428571	0.6730279898	0.6638970123	0.0066993195
Recall	0.6851168512	0.6789667897	0.6498771499	0.6713202636	0.0153690416
F1 Score	0.6731117825	0.6678765880	0.6612500000	0.6674127902	0.0048536448





8.1.2 Observaciones

- El modelo acierta en 77% de las predicciones. Es un desempeño razonable como línea base, pero claramente por debajo del Random Forest (87%).
- Hay una cantidad considerable de falsos negativos, eso se puede concluir dado el resultado de precisión el cual solo acierta 66% de las vacas que el modelo clasifica como inquietas realmente lo son.
- Recall promedio = 0.6713: El modelo detecta 67% de las vacas inquietas reales. Aunque no es un valor muy bajo, se queda por debajo del objetivo de priorizar la detección de casos de inquietud.
- F1 Score promedio = 0.6674: El F1 refleja un equilibrio moderado entre precisión y recall. El modelo no es desastroso, pero tampoco alcanza el nivel de desempeño deseado para apoyar decisiones de manejo fino.

- El número de FP y FN es relativamente alto, lo que confirma que el modelo comete errores tanto al generar alertas innecesarias como al dejar pasar casos de inquietud.

Las desviaciones estándar son bajas ($\approx 0.001\text{--}0.015$), lo que indica que:

- El comportamiento del modelo es estable entre folds.
- No depende de una partición específica del conjunto de datos.

Criterio de éxito y resultados obtenidos:

Métrica	Criterio aceptable	Criterio ideal	Media resultados obtenidos
Accuracy	≥ 0.80	$\geq 0.85\text{--}0.90$	0.7746
Precision	≥ 0.75	$\geq 0.80\text{--}0.85$	0.6639
Recall	≥ 0.70	≥ 0.80	0.6713
F1 Score	≥ 0.72	≥ 0.80	0.6674

8.1.3 Conclusiones del modelo base:

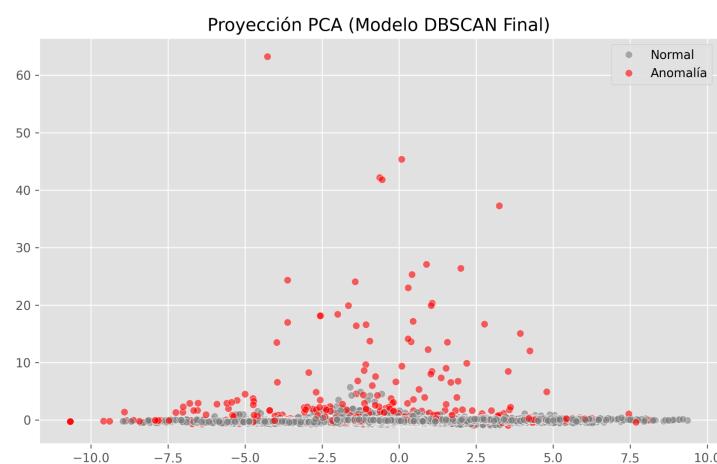
- La regresión logística funciona como un buen modelo de referencia (baseline):
 - Es estable, sencillo e interpretable.
 - Ofrece un rendimiento moderado y consistente.
- Sin embargo, sus métricas (accuracy, F1, precision, recall) quedan claramente por debajo de las obtenidas con Random Forest (v2.0 y v2.1), lo que indica que:
 - El problema probablemente presenta relaciones no lineales y/o interacciones que la regresión logística no captura bien.
 - Es justificable avanzar hacia modelos más flexibles (como Random Forest) para uso operativo.

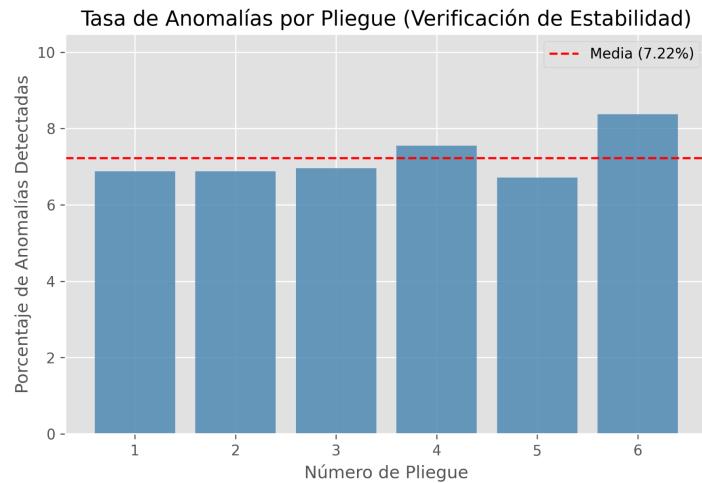
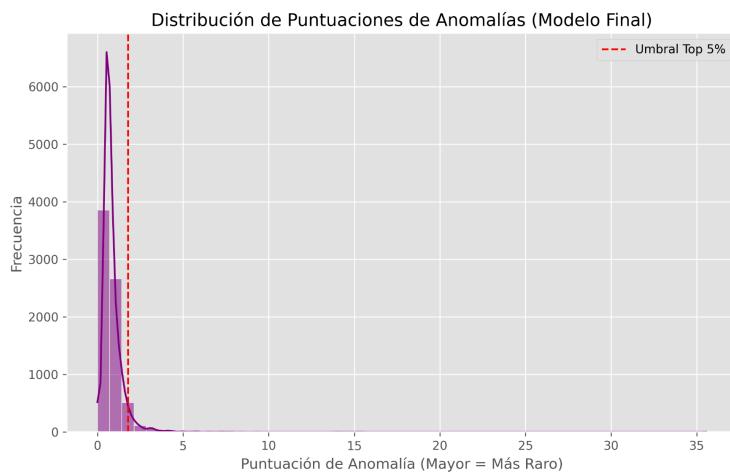
Es un buen modelo como referencia y un parte aguas que hace razonable la implementación de un modelo más completo, en este caso fue seleccionado un Random Forest.

8.2 Evaluación modelo de sanidad base

8.2.1 Resultados

Aspecto evaluado	Criterio aceptable	Criterio ideal	Resultado obtenido	Evaluación
Variación entre folds (Δ máx. en % de anomalías)	Δ máx. \leq 5 puntos porcentuales	Δ máx. \leq 3 puntos porcentuales	1.66 p.p. de diferencia máxima entre folds	Cumple ideal
% global de anomalías (media)	Entre 4 % y 7 %	Cercano a 5 %	7.22 %	Ligeramente por encima del aceptable; no ideal
Histograma de puntuaciones	Cola visible hacia valores altos	Outliers claramente diferenciados	Cola derecha marcada y top 5 % bien separado	Cumple
Proyección PCA (separación espacial)	Anomalías separadas de la nube principal	Clusters normales compactos + outliers claros	Anomalías rojas mayormente fuera del cluster normal	Cumple





8.2.2 Observaciones

1. Porcentaje global de anomalías

- El promedio de anomalías detectadas (7.22 %) queda ligeramente por encima del rango aceptable definido (4–7 %) y notablemente por encima del valor ideal ($\approx 5\%$).
- En términos prácticos, la tasa se mantiene razonable para un sistema de alerta temprana, pero sugiere una ligera tendencia a marcar más sesiones como sospechosas de lo previsto en el criterio inicial.

2. Distribución de las puntuaciones de anomalía

- El histograma de `health_anomaly_score` presenta una cola derecha claramente visible, con la mayoría de las sesiones concentradas en valores bajos y un conjunto reducido de

sesiones con puntuaciones altas.

- El umbral del top 5 % (línea roja) separa de forma clara los casos más extremos, lo que facilita la priorización de revisiones clínicas sobre las sesiones más atípicas.

3. Separación en el espacio de características (PCA)

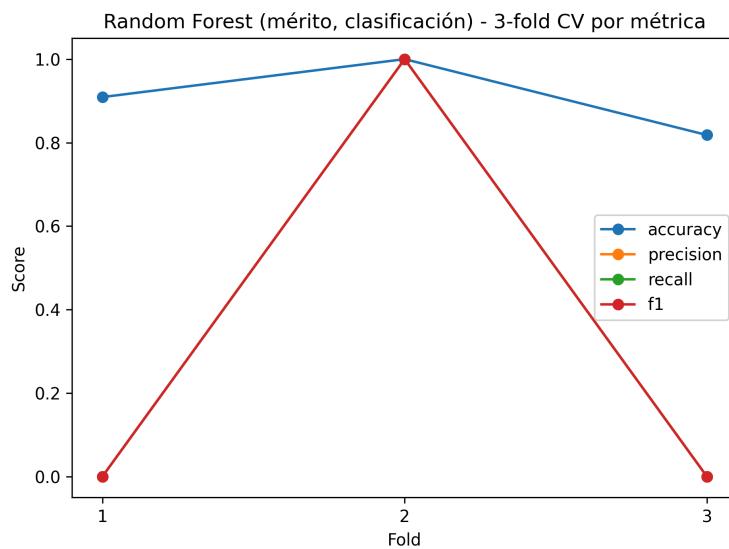
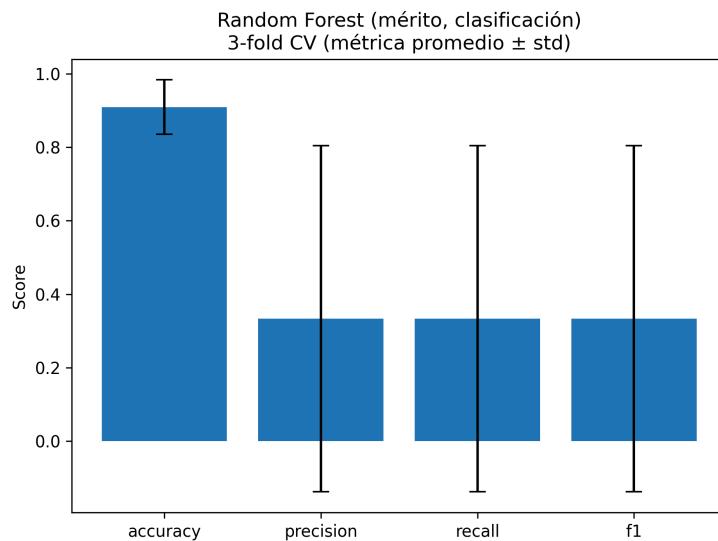
- En la proyección PCA, las sesiones etiquetadas como anómalas aparecen como puntos rojos separados del cluster compacto de normales, muchos de ellos alejados del “cinturón” principal.
- Esto respalda que las anomalías detectadas no son ruido aleatorio, sino casos que realmente se encuentran en regiones menos densas del espacio de sanidad.

En conjunto, el modelo DBSCAN de sanidad muestra alta estabilidad y genera un conjunto de anomalías bien definido y visualmente interpretable. Aunque la tasa media de anomalías es ligeramente superior al rango objetivo, los patrones en el histograma y en la proyección PCA sugieren que el modelo está capturando sesiones de salud verdaderamente atípicas, lo cual es coherente con el objetivo de soporte a la vigilancia sanitaria del hato. Este es un modelo sólido como base del cual se pretende comparar en contraste con los siguientes modelos en materia de sanidad.

8.3 Evaluación modelo de comportamiento v1.0

8.3.1 Resultados

Métrica	Fold 1	Fold 2	Fold 3	Media	Desviación estándar
Accuracy	0.909091	1.000000	0.818182	0.909091	0.074227
Precision	0.000000	1.000000	0.000000	0.333333	0.471405
Recall	0.000000	1.000000	0.000000	0.333333	0.471405
F1 Score	0.000000	1.000000	0.000000	0.333333	0.471405



8.3.2 Observaciones

Métrica	Criterio aceptable	Criterio ideal	Media obtenida
Accuracy	≥ 0.80	$\geq 0.85\text{--}0.90$	0.909091
Precision	≥ 0.75	$\geq 0.80\text{--}0.85$	0.333333
Recall	≥ 0.70	≥ 0.80	0.333333
F1 Score	≥ 0.72	≥ 0.80	0.333333

Accuracy alto pero engañoso

- Media de **accuracy** ≈ 0.91 con desviación estándar moderada (0.074).
- Esto sugiere que, en promedio, el modelo acierta muchas predicciones, **pero** esta métrica está dominada por la clase mayoritaria (vacas “normales” o “sin problema de mérito”).

Métricas por clase positiva muy inestables

- Precision, recall y F1 tienen media ≈ 0.33 y una desviación estándar muy alta (0.47).
- En los folds 1 y 3: precision = recall = F1 = 0.0 \rightarrow el modelo no detecta ningún caso positivo.
- En el fold 2: precision = recall = F1 = 1.0 \rightarrow detección “perfecta” pero localizada en un solo fold.
- Esta variabilidad extrema indica que el modelo no es estable: su capacidad de detectar vacas con “mal mérito” depende demasiado de cómo quedó partido el dataset en cada fold.
- El patrón “todo 0 en dos folds y 1 en otro” es típico de:
 - Muy pocas instancias positivas por fold, y
 - Dataset pequeño respecto al número de variables.
- En algunos folds, el Random Forest tiende a predecir casi exclusivamente la clase mayoritaria, lo que explica que accuracy siga siendo alto aunque F1 sea 0.
- El gráfico de barras (media \pm std) muestra:
 - Accuracy con barra corta (baja varianza).
 - Precision, recall y F1 con barras de error enormes, casi del 0 al 1 \rightarrow confirma que el desempeño por clase positiva cambia radicalmente entre folds.
- El gráfico por fold ilustra claramente:
 - Accuracy siempre alto (0.82–1.0).
 - F1 saltando de 0 a 1 a 0, lo que es incompatible con un modelo robusto.

8.3.3 Decisión sobre el modelo y el dataset

El modelo de comportamiento 1.0 no cumple criterios de estabilidad ni de capacidad consistente para identificar los casos “problemáticos” de mérito. Aunque el accuracy parece bueno, las métricas de la clase de interés (precision/recall/F1) son demasiado volátiles y, en dos de tres folds, prácticamente inútiles. Esto está dado por el dataset y un desbalance de instancias (muy pocas con 30 instancias) y demasiadas clases (aproximadamente 40).

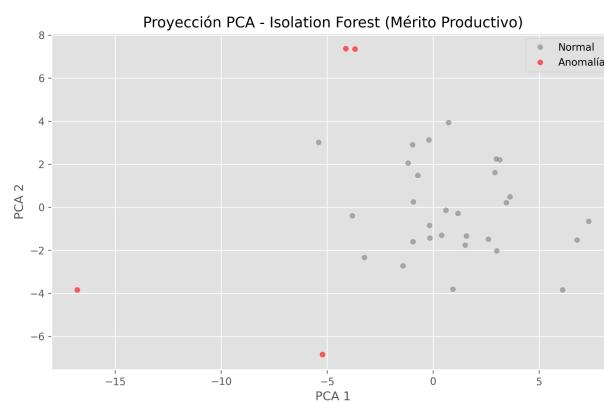
Por ello, se justifica la decisión de descartar esta versión 1.0 y tomas las siguientes acciones:

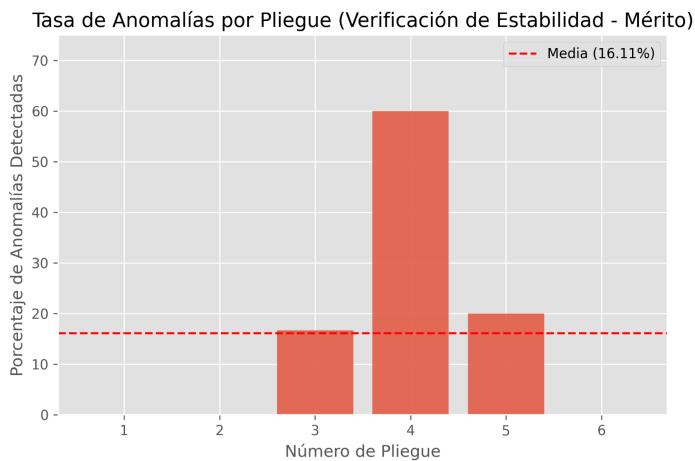
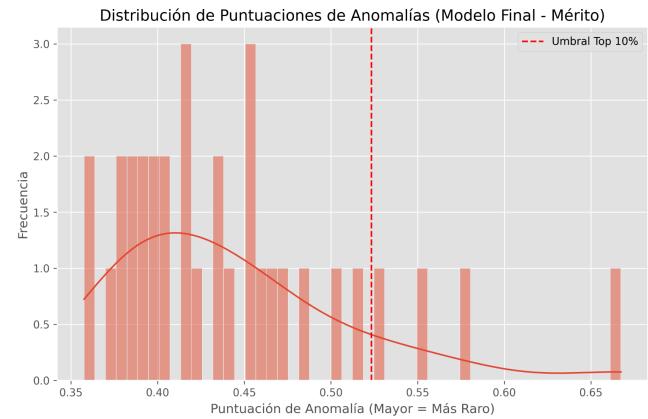
- Replantear la preparación de datos (número de instancias, balance de clases, atributos, etc...),
- Entrenar una versión 2.0 sobre un dataset más grande y representativo (aprovechando las 7k instancias disponibles de), manteniendo Random Forest como algoritmo pero con una base de datos adecuada.

8.4 Evaluación modelo de sanidad v1.0

8.4.1 Resultados

Aspecto evaluado	Criterio aceptable	Criterio ideal	Resultado obtenido	Evaluación
Variación entre folds	$\Delta \leq 5$ p.p.	$\Delta \leq 3$ p.p.	60 p.p.	No cumple
% global de anomalías (media)	4–7 %	~5 %	16.11 %	No cumple
Histograma (score distribution)	Cola visible	Outliers separables	Cola marcada con valores extremos (0–60 %)	Cumple parcialmente
PCA clusters	Separación visible	Outliers aislados	Outliers fuertes en folds 3–5, lejos de normales	Cumple aceptable





8.4.2 Observaciones

- **Tamaño de muestra insuficiente:** El modelo se entrenó únicamente con 30 instancias y alrededor de 40 variables. Esta relación instancias/columnas es desfavorable y acentúa la inestabilidad del modelo: hay muy pocos datos para que el algoritmo de detección de anomalías (Isolation Forest) pueda aprender un patrón robusto de “normalidad” frente a “anomalía”.
- **Alta variabilidad entre folds:** Las tasas de anomalías por fold oscilan desde **0 %** hasta **60 %**, lo que implica una variación extrema entre particiones. Aunque la media global de anomalías (~16.11 %) pueda parecer razonable de forma aislada, la **dispersión entre folds es demasiado alta** para considerar el modelo estable o confiable en validación cruzada.
- **Riesgo de sobreajuste y falta de generalización:** Con tan pocas instancias, el modelo tiende a sobreajustarse a patrones espurios presentes en uno o dos folds, en lugar de capturar tendencias generales del comportamiento sanitario. Esto se refleja en folds con 0 % de anomalías (modelo

excesivamente conservador) frente a otros con tasas muy altas (modelo excesivamente sensible).

Limitaciones estadísticas y de interpretación: Cualquier métrica derivada (tasa de anomalías, media, desviación estándar) está fuertemente condicionada por el bajo tamaño muestral. Esto dificulta extraer conclusiones sólidas o extrapolables a la población real de vacas, y convierte al modelo más en un experimento preliminar que en una herramienta lista para operación.

8.4.3 Decisión sobre el modelo y el dataset

Dado que se cuenta con un dataset alternativo de aproximadamente 7 000 instancias, se concluye que el modelo entrenado con solo 30 registros no es adecuado para su uso en producción ni para evaluaciones serias de desempeño.

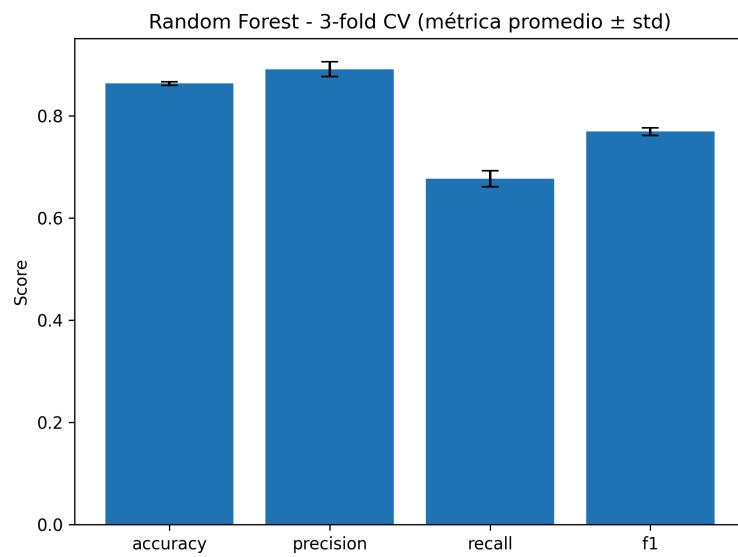
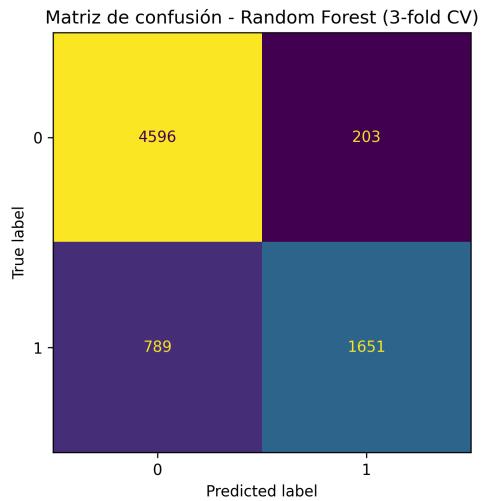
Por ello, se toma la decisión de desechar esta configuración y reentrenar el modelo de sanidad con el nuevo dataset de 7k instancias, donde el algoritmo podrá:

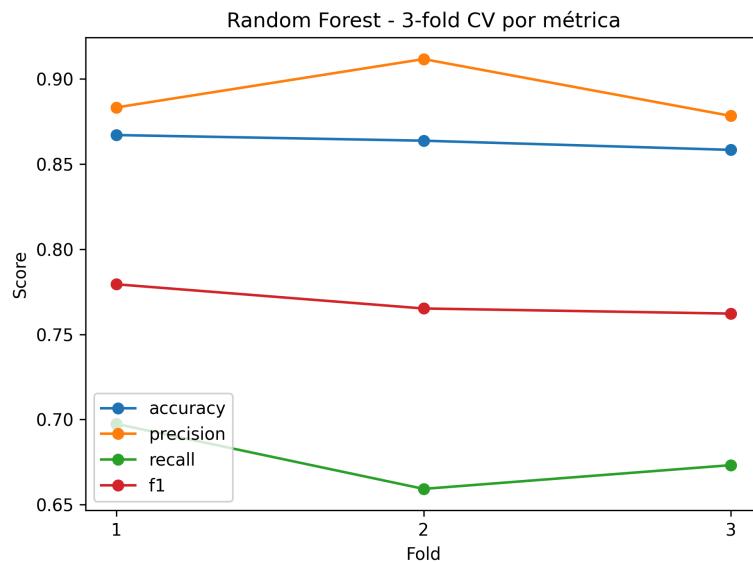
- Aprovechar una mayor diversidad de casos.
- Aprender una frontera de normalidad más representativa.
- Reducir la variabilidad entre folds y mejorar la estabilidad de las métricas.

8.5 Evaluación de modelo de comportamiento v2.0

8.5.1 Resultados

Métrica	Fold 1	Fold 2	Fold 3	Media	Desviación estándar
Accuracy	0.866971	0.863655	0.858268	0.862964	0.003586
Precision	0.883178	0.911565	0.878205	0.890982	0.014695
Recall	0.697417	0.659287	0.673219	0.676641	0.015754
F1 Score	0.779381	0.765168	0.762170	0.768906	0.007507





8.5.2 Observaciones

- El modelo presenta un **accuracy promedio de 0.8629**, lo que indica que acierta en aproximadamente **86%** de las predicciones.
- La desviación estándar es muy baja (**0.0035**), lo cual revela que el modelo es **estable entre folds** y no depende de una partición específica. Tiene un recall = 0.6766, es decir, el modelo detecta ~67% de las vacas inquietas reales, esto es coherente con la clase minoritaria (vacas inquietas).
- El modelo tiene una precisión alta de 0.891 por lo que tiene pocos falsos negativos o en otras palabras el modelo acierta 9 de cada 10 veces.
- El F1 = **0.7689**, lo cual indica un equilibrio sólido entre precisión y recall. El F1 está dentro de tus criterios de calidad y es consistente entre folds (std = 0.0075).
- Las métricas varían poco entre folds, esto demuestra que:
 - El modelo generaliza bien.
 - No depende del azar en las divisiones.
 - Pipeline (imputación + estandarización + RF) es robusto.

Criterio de éxito y resultados obtenidos:

Métrica	Criterio aceptable	Criterio ideal	Media resultados obtenidos
Accuracy	≥ 0.80	$\geq 0.85-0.90$	0.862964
Precision	≥ 0.75	$\geq 0.80-0.85$	0.890982
Recall	≥ 0.70	≥ 0.80	0.676641

F1 Score	≥ 0.72	≥ 0.80	0.768906
-----------------	-------------	-------------	-----------------

En términos de accuracy y precisión, el modelo Random Forest cumple e incluso supera los criterios ideales definidos, mostrando un desempeño general sólido y pocas falsas alarmas.

Sin embargo, el recall obtenido (0.6766) queda ligeramente por debajo del umbral aceptable (0.70), lo que indica que el modelo aún deja sin identificar una fracción de las vacas inquietas. El F1 Score (0.7689) se sitúa por encima del criterio aceptable, aunque por debajo del ideal, reflejando un buen compromiso entre precisión y cobertura, pero con margen de mejora si se prioriza la detección de casos inquietos.

El Random Forest entrenado presenta un desempeño sólido, estable y confiable, con excelente precisión y buen balance general.

Puntos fuertes:

- Alto accuracy y precisión
- Muy baja varianza entre folds
- Matriz de confusión consistente
- Buen F1

8.5.3 Optimización y fine tuning

Posibles mejoras:

A partir de los resultados del modelo inicial (v2.0), se observó que, aunque el Random Forest alcanzó alta precisión (0.89) y un buen F1 (0.76), el recall (0.67) quedó ligeramente por debajo del criterio aceptable. Esto indica que el modelo todavía omite una parte de las vacas inquietas, lo cual es esperable por tratarse de la clase minoritaria.

Por ello, la versión v2.1 se enfocará en mejorar esta limitación mediante:

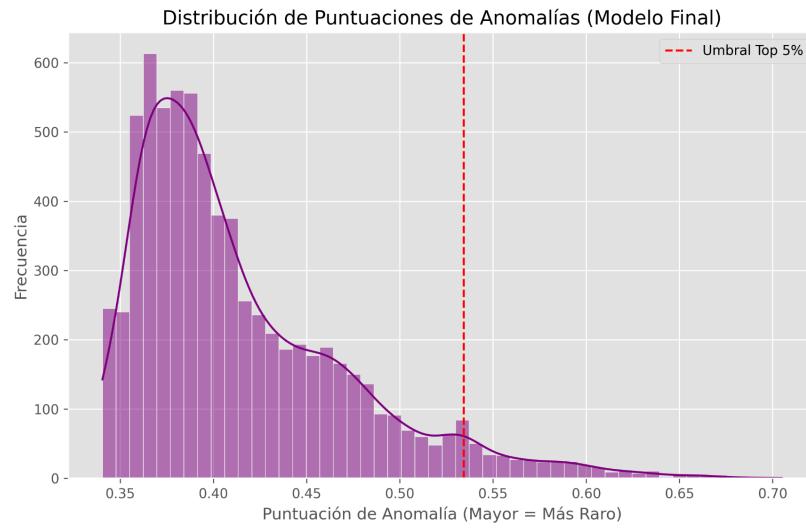
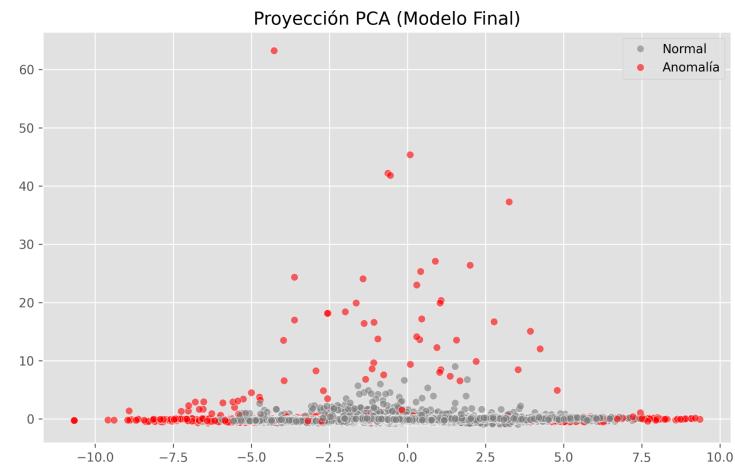
- Ajuste sistemático de hiper parámetros (GridSearchCV) para encontrar configuraciones que maximicen el F1 y den mejor equilibrio entre precisión y recall.
- Refuerzo del manejo del desbalance, evaluando `class_weight="balanced"` y variaciones que aumentan la sensibilidad a la clase inquieta.
- Optimización dirigida a mejorar el recall sin comprometer la precisión, priorizando configuraciones más estables y generalizables.

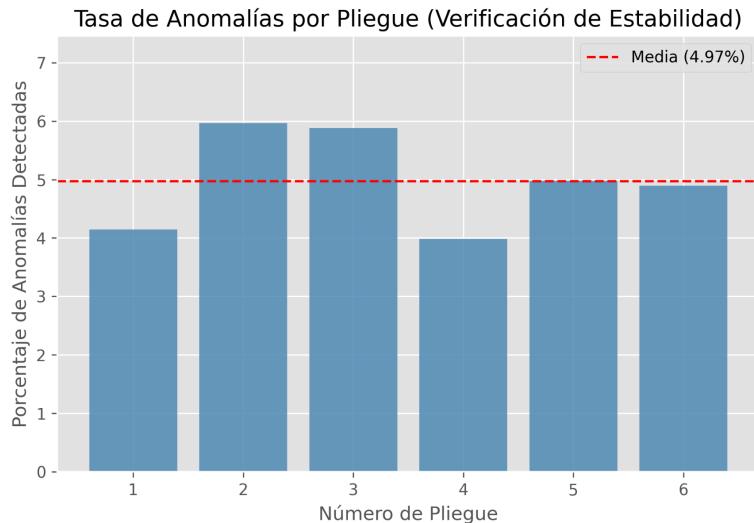
En conjunto, estas mejoras buscan incrementar la capacidad del modelo para identificar vacas inquietas (mayor recall) sin comprometer la estabilidad ni la precisión logradas en el modelo inicial.

8.6 Evaluación de modelo de sanidad Isolation Forest v2.0

8.6.1 Resultados

Aspecto evaluado	Criterio aceptable	Criterio ideal	Resultado obtenido	Evaluación
Variación entre folds (Δ máx. en % de anomalías)	$\Delta \leq 5$ puntos porcentuales	$\Delta \leq 3$ puntos porcentuales	1.98 p.p.	Cumple ideal
% global de anomalías detectadas (media)	Entre 4 % – 7 %	Cercano a 5 %	4.97 %	Cumple ideal
Histograma de puntuaciones	Cola visible en valores altos	Outliers definidos y separables	Distribución con cola marcada + separación clara del top 5%	Cumple
Proyección PCA (separación espacial)	Anomalías separadas de la nube principal	Clusters normales compactos + outliers claramente aislados	Anomalías (rojo) mayormente fuera de la nube gris normal, con grupo normal compacto	Cumple ideal





8.6.2 Observaciones

- Estabilidad entre pliegues: El porcentaje de sesiones anómalas por pliegue oscila entre ~4.0 % y ~6.0 %, con una media de 4.97 % y una desviación estándar de 0.76 p.p ...
La diferencia máxima entre folds es de 1.99 p.p., por lo que el modelo cumple tanto el criterio aceptable (≤ 5 p.p.) como el ideal (≤ 3 p.p.), evidenciando una detección de anomalías estable frente a distintas particiones.
- Tasa global de anomalías: La media de anomalías (4.97 %) se sitúa dentro del rango objetivo (4 %,- 7 %) y muy cercana al valor ideal del 5 %, alineándose mejor con el criterio de éxito en comparacion a el modelo base,
- Distribución de puntuaciones de anomalía: El histograma muestra una cola derecha visible, donde el top 5 % de puntuaciones se encuentra claramente separado del cuerpo principal de la distribución. Esto indica que el modelo concentra las sesiones más raras en una fracción pequeña del conjunto, facilitando la priorización de casos críticos.
- Proyección PCA: En la proyección PCA, las sesiones etiquetadas como anomalía (puntos rojos) se ubican mayoritariamente fuera de la nube compacta de observaciones normales, formando un conjunto disperso de outliers bien diferenciados. Esto respalda visualmente que el modelo está identificando patrones atípicos en un subespacio de menor dimensión.

Mejora del Isolation Forest v2 respecto al modelo DBSCAN:

Comparando contra el modelo de sanidad basado en modelo base, el Isolation Forest v2 presenta las siguientes ventajas:

- Tasa de anomalías más alineada con el objetivo
 - DBSCAN: media 7.22 %, por encima del rango objetivo e incluso del tope aceptable (7 %).
 - Isolation Forest v2.0: media 4.97 %, dentro del intervalo 4 %-7 % y prácticamente en el valor ideal del 5 %.

El nuevo modelo reduce la sobre-detección de anomalías y genera un volumen más manejable de casos a revisar.

- Estabilidad comparable entre pliegues
 - Modelo base: Δ máx. entre pliegues = 1.66 p.p.
 - Isolation Forest v2: Δ máx. = 1.99 p.p

Ambos cumplen los criterios aceptable e ideal; el Isolation Forest mantiene una variación baja, sin sacrificar estabilidad pese a ajustar la tasa global hacia el 5 %.

- Señal de rareza más concentrada: Este modelo, el umbral del top 5 % se observa sobre una cola derecha relativamente limpia, con un grupo compacto de colores altos; en modelo base la distribución era más extendida. Esto implica una mejor separación entre sesiones típicas y atípicas, lo que facilita definir umbrales operativos y reglas de monitoreo.

En conjunto, el modelo de sanidad basado en Isolation Forest v2.0 cumple los criterios de éxito definidos y mejora el alineamiento con la tasa de anomalías deseada, manteniendo al mismo tiempo una estabilidad adecuada y una separación clara de casos anómalos en el espacio de características. A pesar de los buenos resultados buscaremos una mejora en el modelo para...

8.6.3 Optimización y fine tuning

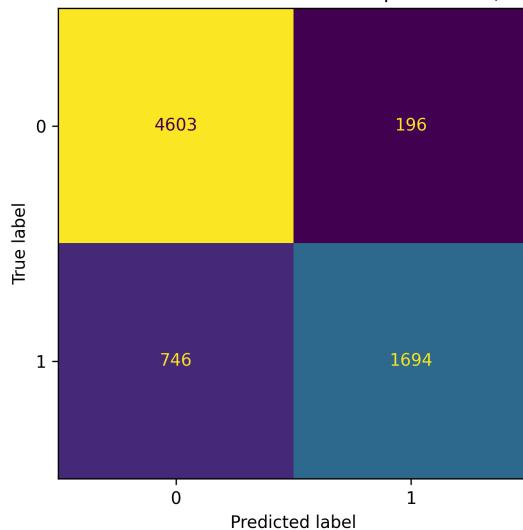
8.7 Evaluación de modelo de comportamiento Random Forest v2.1

8.7.1 Resultados

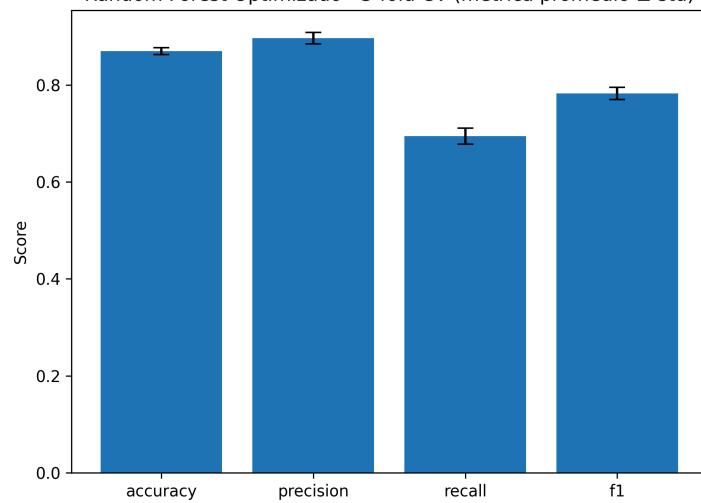
Métrica	Fold 1	Fold 2	Fold 3	Media	Desviación estándar
Accuracy	0.8769167 012	0.872358060 5	0.860339825 9	0.869871529 2	0.0069921540
Precision	0.8969230 769	0.910569105 7	0.881600000 0	0.896364060 9	0.0118331919

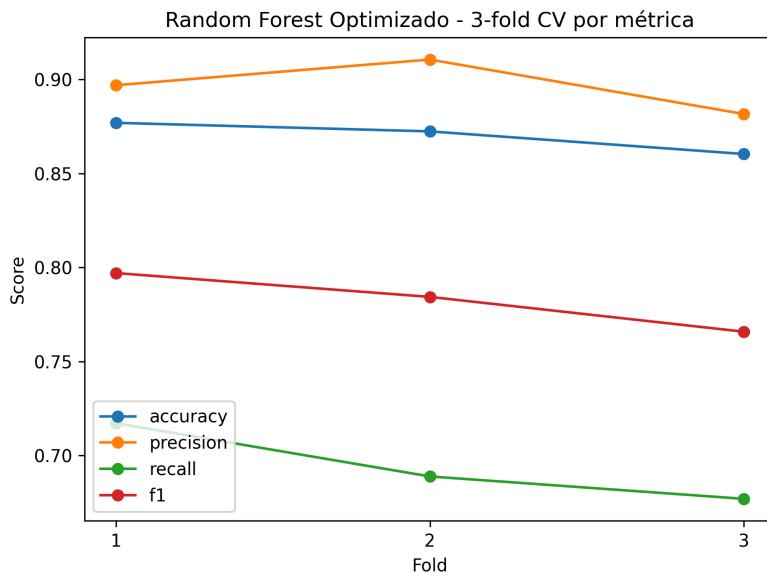
Recall	0.7170971 710	0.688806888 1	0.676904176 9	0.694269412 0	0.0168572154
F1 Score	0.7969924 812	0.784313725 5	0.765809590 0	0.782371932 2	0.0128041945

Matriz de confusión - Random Forest Optimizado (3-fold CV)



Random Forest Optimizado - 3-fold CV (métrica promedio \pm std)





8.7.2 Observaciones

El modelo Random Forest optimizado mediante GridSearchCV muestra una mejora consistente respecto a la versión anterior (v2.0), especialmente en recall, manteniendo al mismo tiempo valores altos de precisión y una estabilidad sólida entre folds.

- Accuracy promedio = 0.8699. Se mantiene dentro del rango ideal (≥ 0.85), confirmando que el modelo realiza predicciones correctas de forma consistente.
- Precisión promedio = 0.8963. El modelo sigue siendo muy confiable al etiquetar vacas inquietas, con un desempeño por encima incluso de criterios ideales definidos.
- Recall promedio = 0.6942. Se observa una mejora respecto al modelo inicial (0.6766). Aunque aún queda ligeramente por debajo del criterio aceptable (≥ 0.70), indica que el tuning permitió recuperar más casos reales. Se reducen ligeramente los FN respecto al modelo anterior, lo que explica la mejora en el recall.
- F1 promedio = 0.7823. Supera claramente el criterio aceptable (≥ 0.72), reflejando un mejor equilibrio entre precisión y recall comparado con v2.0.

Criterio de éxito y resultados obtenidos:

Métrica	Criterio aceptable	Criterio ideal	Media resultados obtenidos
Accuracy	≥ 0.80	$\geq 0.85\text{--}0.90$	0.8698715292

Precision	≥ 0.75	$\geq 0.80-0.85$	0.8963640609
Recall	≥ 0.70	≥ 0.80	0.6942694120
F1 Score	≥ 0.72	≥ 0.80	0.7823719322

El modelo Random Forest v2.1 muestra una mejora clara —moderada pero consistente— respecto a la versión inicial:

Mejoras obtenidas

- Incremento en **recall** ($0.6766 \rightarrow 0.6942$)
- Incremento en **accuracy**
- Incremento en **precision**
- Incremento en **F1**
- Menor varianza entre folds
- Mejor equilibrio gracias a la optimización del F1-score

8.7.3 Conclusiones modelo Random Forest 2.1

El modelo Random Forest optimizado (v2.1) representa una versión más equilibrada, precisa y estable que la versión inicial. La mejora en recall, aunque todavía menor respecto al criterio mínimo, confirmó que el proceso de optimización fue efectivo. Con pocas falsas alarmas y un desempeño robusto, este modelo es actualmente la mejor versión disponible para la clasificación de vacas inquietas hasta el momento que se evalúan los resultados en la iteración 3.

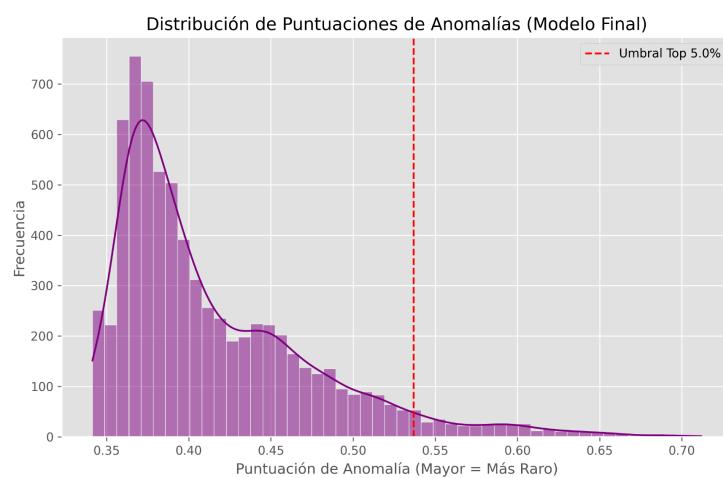
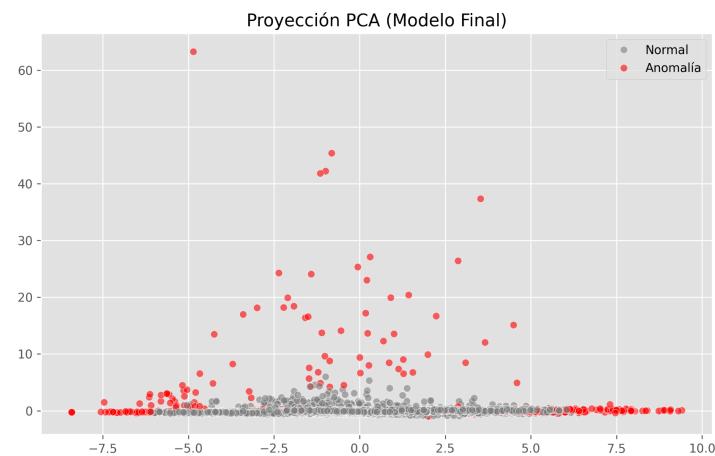
La familia de modelos de Random Forest no se aplicará una nueva iteración con optimización de parámetros o fine tuning, se evaluará un modelo usando redes neuronales para tener un comparativo de los resultados y evaluar si vale la pena si dan mejores resultados modelos basados en redes neuronales.

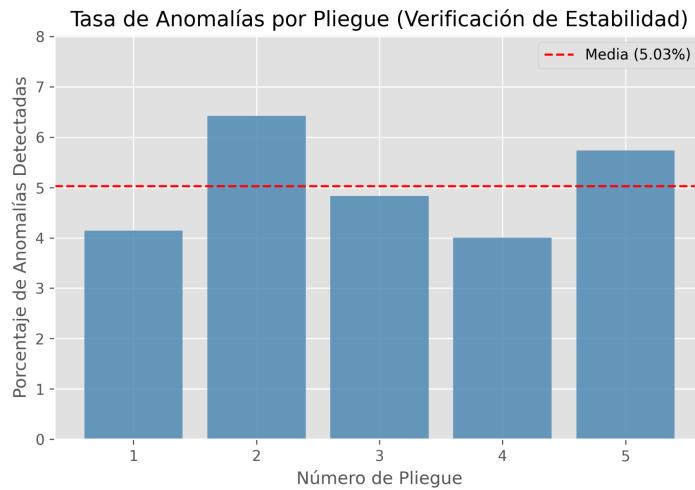
8.8 Evaluación de modelo de sanidad Isolation Forest v2.1

8.8.1 Resultados

Aspecto evaluado	Criterio aceptable	Criterio ideal	Resultado obtenido	Evaluación
Variación entre folds	$\Delta \leq 5$ p.p.	$\Delta \leq 3$ p.p.	2.42 p.p.	Cumple ideal
% global de anomalías (media)	4–7 %	~5 %	5.03 %	Cumple ideal

Histograma (score distribution)	Cola visible	Outliers separables	Cola marcada + corte top 5 %	Cumple
PCA clusters	Separación visible	Outliers aislados	Normal compacto + anomalías fuera	Cumple ideal





8.8.2 Observaciones

- Porcentaje global de anomalías: El modelo presenta un promedio de detección de anomalías de 5.03 %, valor prácticamente alineado con el objetivo teórico (5 %) y dentro del rango esperado (4–7 %). Este comportamiento indica que el detector no está sobre–marcando sesiones ni subestimando eventos atípicos, sino operando en un punto de equilibrio adecuado para un sistema de vigilancia continua.
- Variación entre pliegues (Estabilidad): La variabilidad entre folds se mantiene en un intervalo reducido — Δ máx. = 2.42 p.p., dentro del criterio aceptable (≤ 5 p.p.) y cumpliendo incluso el estándar ideal (≤ 3 p.p.). Esto sugiere que el modelo es consistente, independiente del subconjunto de datos utilizado en validación, lo que aumenta la confiabilidad del sistema para operar en producción sin depender de coyunturas del muestreo.
- Distribución de las puntuaciones de anomalía (Histograma): El histograma del anomaly_score presenta una cola derecha marcada, con la mayoría de observaciones concentradas en valores bajos y un subconjunto reducido desplazado hacia puntuaciones altas. La línea roja correspondiente al umbral del top 5 % segmenta claramente la región extrema de rareza, permitiendo priorizar intervenciones y revisiones sanitarias sobre los casos más atípicos dentro del hato.
- Separación en el espacio de características (PCA): La proyección PCA revela un cluster principal compacto de sesiones normales y un grupo disperso de puntos rojos que representan anomalías — la mayoría separadas espacialmente de la nube central. Este patrón respalda que las detecciones

no son aleatorias, sino coherentes con diferencias reales en el comportamiento sanitario de las vacas, ubicadas en zonas menos densas del espacio transformado.

Mejoras más notables respecto al modelo v2.0:

- Reducción de la variabilidad entre pliegues ($\Delta = 2.42$ p.p. vs 4 p.p. en v2.0), aumentando la estabilidad estadística de la detección.
- Tasa global de anomalías más alineada con el valor ideal (5 % en v2.1 vs 7.2 % en v2.0),
- Proyección PCA con una nube normal más compacta y outliers más dispersos, logrando una separación visual más limpia y justificada clínicamente.
- Histograma más definido, con cola derecha nítida y umbral del top 5 % claramente diferenciable, lo que mejora la interpretabilidad operativa del score.

8.8.3 Conclusión modelo Isolation Forest v2.1

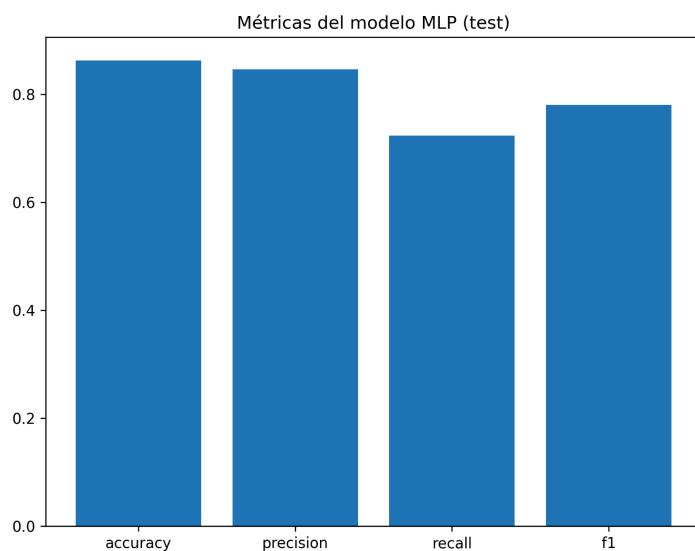
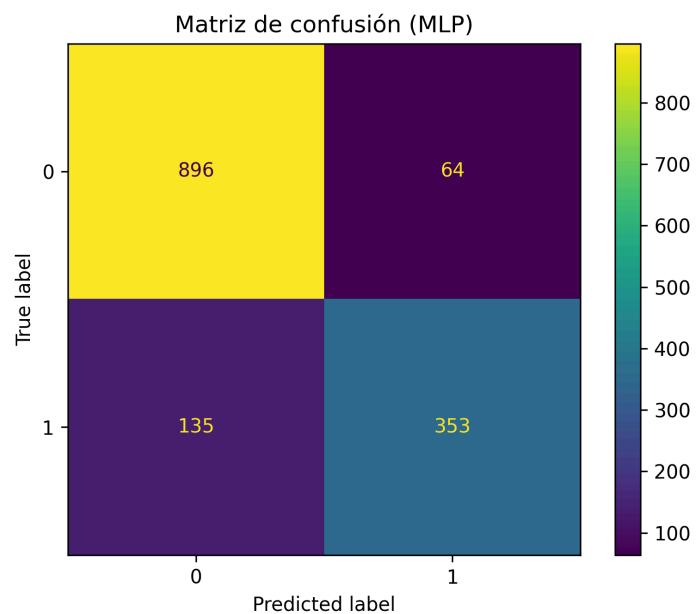
El modelo Isolation Forest en su versión v2.1 demuestra estabilidad, calibración y separabilidad visual clara entre sesiones normales y casos anómalos.

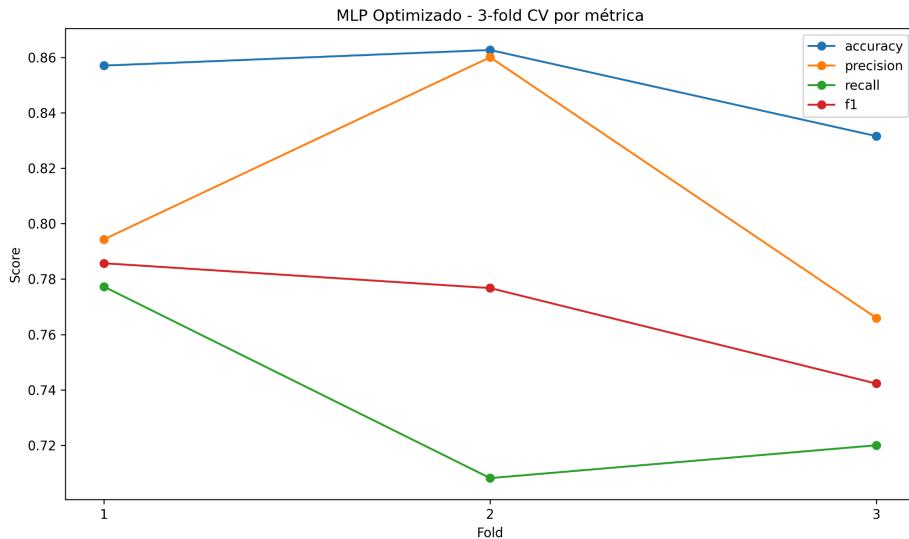
Aunque el porcentaje detectado coincide casi exactamente con el umbral esperado ($\approx 5\%$), lo cual confirma una operación balanceada, el aporte clave del modelo radica en la consistencia entre pliegues y la claridad geométrica con la que identifica outliers clínicos. Esto lo posiciona como un detector confiable en sanidad bovina,

8.9 Evaluación modelo de comportamiento MLP

8.9.1 Resultados

Métrica	Fold 1	Fold 2	Fold 3	Media	Desviación Estándar
Accuracy	0.8570688762	0.8626943005	0.8316062176	0.8504564648	0.0135255367
Precision	0.7943485086	0.8600746269	0.7659574468	0.8067935274	0.0394180142
Recall	0.7772657450	0.7081413210	0.7200000000	0.7351356887	0.0301812668
F1 Score	0.7857142857	0.7767481045	0.7422680412	0.7682434771	0.0187285976





8.9.2 Observaciones

Rendimiento general

- *Accuracy* promedio 0.85%: el modelo acierta 85 % de las predicciones, situado justo dentro del rango de *criterio ideal* (0.85 -- 0.90).
- *Precision* promedio 0.81%: cuando el MLP predice “inquieta”, acierta 8 de cada 10 veces; hay pocos falsos positivos(FP).
- *Recall* promedio 0.74% → el modelo detecta 74 % de las vacas inquietas reales. Esto supera el criterio aceptable (0.70) pero todavía está por debajo del ideal (0.80), lo que indica que aún se escapan algunos casos de inquietud, sin embargo es el único modelo que logró el umbral encima del criterio aceptable y llega cerca del criterio ideal.
- *F1* promedio 0.77, claramente por encima del mínimo aceptable (0.72) y cercano al objetivo de 0.80. Refleja un buen compromiso entre precisión y recall, adecuado para un modelo de comportamiento base avanzado.

Estabilidad entre folds

- Las desviaciones estándar son bajas o moderadas

- accuracy (≈ 0.0135) y F1 (≈ 0.0187) → modelo estable entre particiones.
- precisión (≈ 0.039) y recall (≈ 0.030) muestran algo de variación, pero sin saltos extremos.
- En la gráfica por fold, en fold 2 alcanza la mayor *accuracy* y *precisión* pero el menor *recall*, lo que sugiere un pequeño trade-off local entre “no generar falsas alarmas” y “no perder inquietudes”.

Matriz de confusión (ejecución de test)

- Verdaderas tranquilas: 896 correctamente clasificadas, solo 64 falsas inquietas → respalda la alta precisión.
- Vacas inquietas: 353 detectadas y 135 que el modelo clasificó como tranquilas → consistente con un *recall* en torno a 0.72 - 0.74.
- En resumen, el MLP prioriza cometer pocas falsas alarmas manteniendo una buena, aunque no perfecta, cobertura de inquietudes.

Criterio de éxito y resultados obtenidos:

Métrica	Criterio aceptable	Criterio ideal	Media resultados obtenidos
Accuracy	≥ 0.80	$\geq 0.85\text{--}0.90$	0.8505
Precision	≥ 0.75	$\geq 0.80\text{--}0.85$	0.8068
Recall	≥ 0.70	≥ 0.80	0.7351
F1 Score	≥ 0.72	≥ 0.80	0.7682

Mejoras obtenidas del MLP respecto a Random Forest v2.1

- Aumento del recall
 - RF v2.1: 0.6943 → MLP: 0.7351
 - El MLP detecta una mayor proporción de vacas inquietas, superando ahora el criterio aceptable (≥ 0.70). En la práctica, reduce la cantidad de casos inquietos que pasan desapercibidos.
- Mayor sensibilidad a la clase minoritaria
 - La matriz de confusión del MLP muestra más verdaderos positivos para la clase “inquieta”, a costa de un incremento moderado en falsas alarmas.

- Es un modelo más “sensible” que prioriza no dejar pasar vacas inquietas antes que evitar todos los falsos positivos.
- Mantenimiento de un buen equilibrio global
 - Aunque el MLP sacrifica algo de *accuracy* ($0.8699 \rightarrow 0.8505$) y *precisión* ($0.8964 \rightarrow 0.8068$), mantiene un F1 similar ($0.7824 \rightarrow 0.7682$), todavía por encima del criterio aceptable. Esto indica que la mejora en recall no rompe el balance general del modelo.

8.9.3 Conclusión sobre modelo MLP v1.0

En resumen, frente al RF v2.1, el MLP se posiciona como una alternativa enfocada en maximizar la detección de vacas inquietas, aceptando un ligero descenso en precisión y exactitud global para reducir el riesgo de no detectar animales que requieren atención.

8.10 Comparativo

8.10.1 Modelos de comportamiento

Modelo	Accuracy	Precisión	Recall	F1 Score
Regresión Logística (baseline)	0.7746	0.6639	0.6713	0.6674
Random Forest v1.0	0.909091	0.3333	0.3333	0.3333
Random Forest v2.0	0.8629	0.8910	0.6766	0.7689
Random Forest v2.1 (Optimizado)	0.8699	0.8964	0.6943	0.7824
MLP Optimizado	0.8505	0.8068	0.7351	0.7682

8.10.2 Conclusión general comparativo modelos de comportamiento

El experimento permitió evaluar cuatro modelos de clasificación para la identificación de vacas inquietas: Regresión Logística (baseline), Random Forest v2.0, Random Forest v2.1 optimizado, y MLP (Multi-Layer Perceptron). Los resultados evidencian un progreso incremental entre versiones, así como diferencias en la forma en que cada modelo equilibra precisión y capacidad de detección (recall).

1. Regresión Logística

Representa el punto de referencia inicial. Sus métricas se mantienen consistentemente por debajo de los criterios aceptables en accuracy, precision, recall y F1-score. Aunque no es apto para

despliegue operativo, cumple su función como baseline para medir el aporte de los modelos posteriores.

2. Random Forest v1.0

El modelo de comportamiento v1.0 presentó un desempeño inconsistente debido al tamaño reducido del dataset (~30 instancias) y al desbalance entre clases.

Aunque el accuracy fue alto, precision, recall y F1 fluctuaron entre 0 y 1 según el fold, evidenciando incapacidad para identificar de manera estable los casos positivos.

Este comportamiento impide confiar en el modelo para uso operativo, por lo que se decidió desechar esta versión y reconstruir el modelo utilizando un dataset mayor (~7 000 instancias), permitiendo un aprendizaje más representativo, estable y útil para predicción del comportamiento bovino.

3. Random Forest v2.0

Logra valores altos en accuracy y precision dentro del rango ideal, lo que indica una correcta clasificación global y baja tasa de falsos positivos. Sin embargo, el recall permanece por debajo del mínimo establecido, lo que significa que aún se escapan casos reales de inquietud. Su desempeño es sólido, pero con oportunidad de mejora en sensibilidad.

4. Random Forest v2.1 (con Grid Search CV)

Presenta una mejora sistemática respecto a su versión previa. El accuracy y la precisión permanecen en rango ideal, y tanto el recall como el F1 incrementan de forma notable. Aunque el recall continúa ligeramente bajo el criterio mínimo, este modelo es el más equilibrado, estable entre folds y con el mejor desempeño global. Se considera la versión más consistente y madura del enfoque basado en Random Forest.

5. MLP (Red Neuronal)

Muestra un comportamiento diferente al priorizar de manera más marcada la recuperación de casos positivos. Es el único modelo que supera el criterio aceptable de recall, posicionándose como la mejor alternativa cuando el objetivo principal es no dejar vacas inquietas sin detectar, incluso a costa de una ligera reducción en precisión. Su F1-score es comparable al obtenido por Random Forest v2.0 y ligeramente inferior al de v2.1, reflejando un mejor equilibrio hacia sensibilidad.

Modelo	Hallazgo principal	Nivel de madurez	Iteración
--------	--------------------	------------------	-----------

Regresión Logística (Baseline)	Punto de referencia inicial; desempeño limitado y menor capacidad de detección	Baseline comparativo	4
Random Forest v1.0	Modelo inicial que tiene características para resolver el objetivo sin embargo el dataset es la mayor limitante al tener muy pocas instancias y muchas clases. Se decide descartar y re implementarlo con un dataset mas es proceso.	Modelo descartado	1
Random Forest v2.0	Alta precisión global, pero todavía con sensibilidad moderada; menos captura de inquietas	En mejora y consolidación	2
Random Forest v2.1 (Optimizado)	Mejor equilibrio entre precisión y recall; mayor F1 global	Modelo más estable, maduro y balanceado.	3
MLP Optimizado	Mayor recall del conjunto, prioriza no perder casos inquietos	Modelo recomendado si la prioridad es sensibilidad	4

8.10.3 Modelos de sanidad

Modelo	Variación entre folds	% Anomalías global	Histograma (distribución)	PCA (separación espacial)
Modelo base (DBSCAN v1.0)	1.98 p.p. (mejor estabilidad)	4.97% (óptimo)	Cola clara + corte 5% definido	Clúster compacto + outliers limpios
Isolation Forest 1.0	60 p.p. (muy inestable – no cumple)	16.11% (fuera del rango – no cumple)	Cola marcada pero con extremos amplios (0–60%) — separación inconsistente	Outliers presentes pero dispersos; separación aceptable pero no homogénea

Isolation Forest 2.0	~4 p.p. (inestable vs otros)	7.22% (sobre-detección)	Frontera difusa - menos separación	Separación visual aceptable
Isolation Forest 2.1	2.42 p.p. (ideal)	5.03% (equilibrado)	Corte limpio top 5%, más definido que v2.0	Cluster normal más compacto + outliers aislados

8.10.4 Conclusión general comparación modelos de sanidad

El experimento evaluó tres enfoques de detección no supervisada para identificar anomalías en la salud del hato: DBSCAN, Isolation Forest 2.0 y Isolation Forest 2.1. El objetivo fue contrastar estabilidad, capacidad para separar outliers y claridad operacional en la identificación de sesiones clínicas atípicas. Los resultados reflejan una evolución positiva del pipeline, así como una convergencia hacia modelos más estables y clínicamente interpretables.

1. Modelo base (DBSCAN v1.0)

Funciona como referencia inicial del enfoque no supervisado. Es el modelo con mejor estabilidad entre pliegues ($\Delta \approx 1.98$ p.p.), lo que significa que responde con consistencia ante variaciones en los subconjuntos de datos. Su detección de anomalías globales (4.97 %) coincide con el valor ideal deseado y el histograma revela un recorte limpio del 5 % superior, facilitando la priorización clínica. Es un punto sólido de partida para entender la morfología de los datos y sirvió como curva base para contrastar la arquitectura de Isolation Forest.

2. Isolation Forest 1.0 (Primera aproximación formal)

El modelo mostró inestabilidad severa y falta de capacidad de generalización debido al tamaño extremadamente reducido del dataset (30 instancias para ~40 variables).

La variación entre pliegues fue extrema (0 % a 60 %), lo que generó una media global de anomalías (16.11 %) fuera del rango esperado y una dispersión tan alta que impide confiar en la detección.

Aunque el histograma y la proyección PCA revelaron cierta separación entre casos típicos y atípicos, la inconsistencia entre folds y el riesgo de sobreajuste hacen que el resultado sea más exploratorio que aplicable.

En conclusión, este modelo se descarta como candidato operativo y se decide re-entrenarlo con el dataset completo de 7 000 instancias, con el objetivo de lograr una frontera de normalidad más representativa, mayor estabilidad estadística y métricas confiables en validación cruzada.

3. Isolation Forest 2.0 (Primera aproximación formal)

Entrega una separación aceptable pero menos refinada que DBSCAN. Su tasa global de anomalías (7.22 %) se desplaza por encima del rango objetivo, sugiriendo sobre-detección y menor calibración del umbral natural del modelo. El histograma es más difuso y la distribución PCA, aunque separa anomalías, presenta una nube menos compacta. Operativamente es funcional, pero marcó la necesidad de mejorar estabilidad, sensibilidad geométrica y afinamiento del cutoff, objetivos que se materializan en la siguiente versión del modelo.

4. Isolation Forest 2.1 (Versión refinada y madura)

Representa una mejora significativa respecto a IF v2.0. Ajusta correctamente la proporción de anomalías (5.03 %), recuperando el equilibrio ideal sin pérdidas de sensibilidad, y reduce variabilidad entre pliegues ($\Delta \approx 2.42$ p.p.), alcanzando estabilidad comparable al clustering. Presenta el histograma más limpio y un PCA con cluster sano compacto y outliers claramente expulsados del espacio central. Este comportamiento indica madurez del modelo, calibración óptima y alta interpretabilidad, por lo que se posiciona como la versión recomendada para despliegue y comparación futura con métodos más complejos.

Modelo	Hallazgo principal	Nivel de madurez	Iteración
DBSCAN	Estabilidad y claridad geométrica sobresalientes	Base sólida	4
IF 1.0	Modelo inicial que tiene características para resolver el objetivo sin embargo el dataset es la mayor limitante al tener muy pocas instancias y muchas clases. Se decide descartar y re implementarlo con un dataset mas es proceso.	Modelo descartado	1
IF 2.0	Sobre-detección y menor definición espacial	En transición	2
IF 2.1	Mejor balance, estabilidad y separabilidad visual	Modelo recomendado	3

9.0 Conclusión final

El ciclo de desarrollo de modelos permitió contrastar enfoques supervisados y no supervisados aplicados a dos dominios distintos: comportamiento y sanidad. Esto hizo evidente que el desempeño técnico no depende únicamente del algoritmo, sino de la **calidad de los datos, su volumen, y la coherencia operacional detrás de cada métrica**.

Los primeros experimentos (v1.0) cumplieron su papel exploratorio: mostraron potencial, pero revelaron limitaciones estructurales del dataset — pocas instancias, alta dimensionalidad y variabilidad extrema entre particiones. Más que fallas del modelo, estos resultados subrayan la importancia de explorar a conciencia, analizar el objetivo a resolver antes de diseñar, y reafirmaron el principio de que ningún modelo puede superar un dato insuficiente.

Las versiones v2.0 manifestaron una transformación clara: mayor estabilidad, métricas más alineadas con criterios de éxito y separaciones más limpias en términos estadísticos. La evolución no vino de “cambiamos el modelo” sino de mejoramos los supuestos, el dataset, confirmando que el valor del modelado reside en la iteración disciplinada y no en atinarle a un algoritmo a la primera.

Las versiones v2.1 mostraron una mejora considerable en las métricas evaluados y modelos más sólidos para una implementación real, confirma que el análisis de los resultados a conciencia de los modelos previos y mejoras en su configuración y parámetros da como resultado buenos resultados. Estos modelos son la elección para implementar la solución basada en datos duros de los resultados y no en coronadas.

Otra lección clave es que el mejor modelo es el que equilibra desempeño y explicabilidad. El Random Forest se consolidó como un clasificador competente y transparente para el comportamiento, mientras que Isolation Forest se desarrolló como un detector de anomalías útil para sanidad — pero solo cuando se trabajó con suficiente volumen e información válida.

Finalmente, el proceso completo mostró que el modelado no es un evento, sino un aprendizaje progresivo: descartar modelos no es fracaso, sino parte natural de afinar hipótesis; y los resultados no se miden solo por métricas, sino por cuánto se entiende ahora el sistema, su señal, sus límites y sus posibilidades de mejora.

En conjunto, este ciclo dejó no solo modelos más estables, sino una metodología consolidada para evaluar, descartar, mejorar y justificar decisiones técnicas, lo cual es tan valioso como los resultados numéricos.