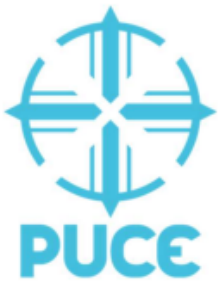
PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR

FACULTAD DE INGENIERÍA



TEMA:

Desarrollo de un Modelo Clasificador para el Descarte de Ganado Vacuno en una Hacienda Ganadera

AUTOR:

Ing. Liseth Andrea Medina Perugachi

TRABAJO PREVIO A LA OBTENCIÓN DEL TÍTULO DE MAGISTER EN SISTEMAS DE INFORMACIÓN MENCIÓN DATA SCIENCE

TUTOR:

Henry Nelson Roa Marin, Ph.D.

Quito, marzo 2025

RESUMEN

El descarte vacuno constituye un proceso esencial en la gestión ganadera, ya que permite identificar y retirar del hato aquellos animales que no cumplen con ciertos criterios productivos y reproductivos. En este contexto, el presente trabajo tiene como finalidad desarrollar un modelo clasificador que optimice este proceso, utilizando técnicas avanzadas de aprendizaje automático.

Para lograr este objetivo, se recopiló un conjunto de datos que incluye diversas características relacionadas con la productividad y la salud de los animales, tales como la producción de leche, el historial de enfermedades, y los parámetros fisiológicos. A partir de estos datos, se implementó un enfoque de aprendizaje supervisado, comparando tres modelos de clasificación: Random Forest, Máquinas de Soporte Vectorial (SVM) y Regresión Logística.

Se utiliza la metodología CRISP-ML, que permite desarrollar y evaluar los modelos de clasificación, adicionalmente se aplican métricas para evaluar la calidad y rendimiento del modelo como accuracy, F1-score, recall y ROC-AUC.

Este trabajo contribuye al avance de la tecnología en el sector agropecuario y abre nuevas posibilidades para la aplicación de técnicas de aprendizaje automático en la gestión ganadera.

ÍNDICE

Contenido

1. Introducción 9

1.1 Antecedentes 10

1.2 Planteamiento del Problema 10

1.3 Justificación 11

1.4 Objetivos 12

1.4.1 Objetivo General 12

1.4.2 Objetivos Específicos 12

1.5 Alcance 13

2. Marco Teórico 14

2.1. Producción Lechera 14

2.1.1 Ciclo de Lactancia 15

2.1.2 Factores que influyen en la producción lechera 16

2.2. Descarte Vacuno 17

2.2.1 Causas del Descarte 19

2.2.2 Impacto del Descarte Vacuno 20

2.3 Machine Learning 20

2.3.1 Aprendizaje Supervisado 21

2.3.2 Modelos de Clasificación 21

2.3.3 Métricas de Clasificación 23

2.4 Metodología CRISP- ML 24

3. Metodología 27

3.1 Comprensión de Datos y Negocio 27

3.1.2 Comprensión de Datos 29

3.2 Ingeniería de Datos 34

3.2.1 Selección de datos 34

3.2.2 Limpieza de datos 47

3.2.3 Identificación y tratamiento de valores nulos 49

3.2.4 Identificación de Outliers 51

3.2.5 Estandarización de datos 57

3.3 Ingeniería de modelos de aprendizaje automático 57

3.3.1 Definir la medida de calidad del modelo 57
3.3.2 Selección del algoritmo de ML 58
3.3.3 Construcción y Evaluación de los modelos 59
3.3.4 Entrenamiento del modelo 61
3.3.5 Selección del modelo 62
3.4 Despliegue 66
3.5 Monitoreo y mantenimiento 69
3.5.1 Monitoreo del Desempeño del Modelo 69
3.5.2 Mantenimiento del Modelo 70
4. Resultados 71
5. Conclusiones y Recomendaciones 74
5.1 Conclusiones 74
5.2 Recomendaciones 75
6. Bibliografía. 77

INDICE DE FIGURAS

Figura 1. Principales provincias productoras de leche 15
Figura 2. Ciclo de lactancia de una vaca lechera 16
Figura 3. Días de producción vs kg/leche día 17
Figura 4. Ciclo productivo de una vaca lechera 18
Figura 5. Proceso de ciclo de vida de CRISP-ML 25
Figura 6. Tipo de datos HistoricoEXCE.csv 30
Figura 7. Tipo de datos Cabezasexce.csv 32
Figura 8. Tipo de datos Clinicoxml.csv 32
Figura 9. Tipo de datos Individual.csv 33
Figura 10. Tipo de datos Reproducción.csv 33
Figura 11. Tipo de datos TipoBaja.csv 34
Figura 12. Carga de Datos historicoEXCE.csv 34
Figura 13. Selección de características historicoEXCE.csv 35
Figura 14. Selección solo hembras histórico descarte 36
Figura 15. Cálculo edad histórico 36
Figura 16. Selección ganado histórico mayor a dos años 37
Figura 17. Carga dataset Tipo de Baja 37
Figura 18. Filtro tipo de baja 38
Figura 19. Carga dataset Enfermedades 38
Figura 20. Frecuencia de enfermedades diagnosticadas por número de arete 39
Figura 21. Left outer join enfermedades 39
Figura 22. Carga dataset Días de lactancia 40
Figura 23. Días de lactancia agrupados por número de arete 40
Figura 24. Left outer join días de lactancia 41
Figura 25. Carga dataset Producción lechera 41
Figura 26. Producción lechera agrupada por número de arete 42
Figura 27. Left outer join producción lechera 42
Figura 28. Inclusión campo descarte “SI” 42
Figura 29. Nueva selección de características dataset ganado descartado 43
Figura 30. Carga dataset Ganado no descartado 43
Figura 31. Selección de características dataset ganado no descartado 44
Figura 32. Cálculo edad dataset ganado no descartado 44
Figura 33. Selección solo hembras dataset ganado no descartado 45
Figura 34. Selección de ganado mayor a dos años dataset ganado no descartado 45
Figura 35. Left outer join enfermedades dataset ganado no descartado 46
Figura 36. Left outer join producción lechera dataset ganado no descartado 46
Figura 37. Inclusión campo descarte “NO” 46
Figura 38. Selección nuevas características ganado no descartado 47
Figura 39. Renombramiento campos dataset Descartadas 47
Figura 40. Cambio tipo de datos dataset Descartadas 48
Figura 41. Eliminar duplicados dataset Descartadas 48
Figura 42. Renombramiento campos dataset noDescartadas 48
Figura 43. Cambio de orden dataset noDescartadas 48
Figura 44. Eliminar duplicados dataset noDescartadas 49
Figura 45. Dataset Final 49
Figura 46. Identificación de valores nulos 50
Figura 47. Tratamiento de valores faltantes 50

Figura 48. Comprobación valores faltantes 51
Figura 49. Tratamiento de outliers 54
Figura 50. Tratamiento de variables categóricas 57
Figura 51. Comprobación desbalance de clases 59
Figura 52. Aplicación validación cruzada 60
Figura 53. Identificación de variables dependientes e independientes 60
Figura 54. Aplicación Random Forest 60
Figura 55. Aplicación SVM 61
Figura 56. Aplicación Regresión Logística 61
Figura 57. Entrenamiento modelo Random Forest 62
Figura 58. Entrenamiento modelo SVM 62
Figura 59. Entrenamiento modelo Regresión Logística 62
Figura 60. Métricas modelo Random Forest 62
Figura 61. Métricas modelo SVM 64
Figura 62. Métricas modelo Regresión Logística 65
Figura 63. Matriz de confusión Regresión Logística 65
Figura 64. Almacenamiento del modelo 67
Figura 65. Carga del nuevo dataset para evaluación del modelo 67
Figura 66. Transformación de variables categóricas nuevo dataset para evaluación 68
Figura 67. Carga del modelo 68
Figura 68. Aplicación función de predicción para descarte vacuno 68
Figura 69. Nuevas predicciones 69
Figura 70. Resultados Métricas Modelo Clasificador Vacuno 71
Figura 71. Curva ROC con Validación Cruzada 72

INDICE DE GRÁFICOS

Gráfico 1. Boxplot incidencia de enfermedades 51
Gráfico 2. Boxplot producción lechera 52
Gráfico 3. Boxplot edad ganado 52
Gráfico 4. Boxplot Días de lactancia 53
Gráfico 5. Boxplot número de partos 53
Gráfico 6. Boxplot número de abortos 54
Gráfico 7. Comprobación de outliers producción 55
Gráfico 8. Comprobación de outliers enfermedades 55
Gráfico 9. Comprobación de outliers días de lactancia 56
Gráfico 10. Comprobación outliers partos 56
Gráfico 11. Matriz de confusión Random Forest 63
Gráfico 12. Matriz de confusión modelo SVM 64

ÍNDICE DE TABLAS

Tabla 1. Plan de Proyecto de Titulación 29
Tabla 2. Comparación de modelos de clasificación 66

1. Introducción

A nivel mundial la industria ganadera se considera uno de los pilares más importantes ya que aporta significativamente a la producción de carne y productos lácteos, constituye una de las practicas más antiguas y es una fuente de alimentos e ingresos para muchas personas que pertenecen a poblaciones rurales alrededor del mundo, debido a esto es importante mejorar la eficiencia y productividad para asegurar su éxito económico y sostenibilidad.

La ganadería y en especial la producción lechera es considerado uno de los sectores económicos más relevantes en cuanto a la generación de empleo en el Ecuador, especialmente en la sierra, se estima que aproximadamente 600.000 personas dependen de la producción de leche. Los productores de leche se encargan de garantizar el autoabastecimiento al Ecuador y contribuyen con el crecimiento económico, la seguridad y soberanía alimentaria del país. La leche, un producto lácteo, ha sido durante mucho tiempo considerado una fuente de ingresos seguro, y se ha mantenido en constante crecimiento durante los últimos años (Hidalgo, 2007).

Las haciendas ganaderas buscan constantemente maximizar la eficiencia, eficacia y rentabilidad de la producción lechera, el descarte vacuno se considera uno de los desafíos más grandes que tienen los ganaderos como parte de la gestión adecuada de sus recursos, el descarte implica la identificación y eliminación de animales que no cumplen con ciertas características para mantener la calidad del rebaño y optimizar los recursos dentro de la hacienda.

Un bovino de descarte se define como aquel animal que debe ser removido del rebaño por distintas razones entre las cuales puede ser por regalo, venta, envió al matadero o muerte del animal. La práctica del descarte dentro de una hacienda lechera se lo hace basado frecuentemente en la experiencia y el juicio subjetivo de los ganaderos, lo que podría llevar muchas veces a tomar decisiones poco eficientes y limitadas de precisión. Bajo este contexto, la tecnología y el manejo avanzado de datos puede representar una oportunidad para optimizar este proceso.

La presente investigación busca desarrollar un modelo clasificador que permita identificar de manera objetiva y precisa los animales candidatos para el descarte vacuno en una hacienda ganadera considerada como caso de estudio, utilizando datos históricos y variables específicas del ganado. Este modelo no solo contribuiría a mejorar la toma de decisiones dentro de la hacienda, sino que también representaría una aplicación de las técnicas de machine learning en un sector de gran importancia dentro del país. A través de este trabajo, se espera ofrecer una

herramienta que se encuentre alineada con las tendencias de la Cuarta Revolución Industrial y apoye al desarrollo sostenible de la ganadería.

1.1 Antecedentes

La producción lechera es una de las actividades económicas más importantes dentro del país, actualmente debido a los avances tecnológicos se cuentan con sistemas de información que ayudan a automatizar ciertas tareas que se llevan a cabo dentro de una hacienda, estos sistemas son capaces de almacenar la información diaria de la hacienda generando de esta forma grandes volúmenes de datos que mediante las habilidades técnicas necesarias pueden convertirse en información valiosa que sirva como un instrumento para la toma de decisiones maximizando la producción lechera y minimizando los índices de pérdida económica.

El descarte vacuno es una práctica bastante común y es parte de la gestión de una hacienda lechera consiste en eliminar vacas del hato basándose en algunos aspectos como: problemas de salud, baja productividad, edad, peso, etc.

Durante los últimos años se ha visto un creciente uso de tecnologías digitales lo que ha hecho que el descarte vacuno no solo se base en la experiencia del ganadero, sino que mediante el procesamiento de la información y la aplicación de técnicas de machine learning se pueden obtener parámetros relevantes que ayuden a tomar mejores decisiones acerca del descarte de los animales.

Mediante el uso de la analítica de datos y otras tecnologías, los ganaderos pueden tomar decisiones más acertadas y debidamente fundamentadas. Actualmente la información representa poder y tener la capacidad de aprovechar al máximo los avances tecnológicos y la información es fundamental para garantizar el éxito en el sector pecuario (Tarifa,2024).

1.2 Planteamiento del Problema

La carencia de un enfoque organizado y basado en datos para el descarte de ganado vacuno puede acarrear una serie de inconvenientes dentro de hacienda como:

- Falta de precisión en la toma de decisiones: El ganadero al basarse únicamente en su experiencia esta propenso a cometer errores.
- Pérdida de productividad: El descarte equivocado de animales que son óptimos para el

hato puede representar perdidas en la producción lechera.

- Costos adicionales: Por el contrario, mantener animales que no son productivos puede incrementar significativamente los costos operativos.

En base a lo mencionado anteriormente, surge la necesidad de desarrollar un modelo clasificador que utilice técnicas de aprendizaje automático para mejorar la precisión y consistencia en el proceso de descarte de ganado vacuno. Este modelo debe ser capaz de analizar múltiples variables relevantes y proporcionar recomendaciones basadas en datos, lo que permitirá a los ganaderos mejorar su producción lechera y a obtener información útil para su manejo.

1.3 Justificación

En el país se estima una producción de aproximadamente 6,15 millones de litros diarios de leche cruda, según el Instituto Nacional de Estadísticas y Censos (INEC) (datos 2020). La producción de leche y la ganadería representan una fuente importante de ingresos para casi 1,2 millones de personas.

Se considera que existe una población de alrededor de 4,1 millones de bovinos, la producción lechera representa el 57% y se desarrolla más en la región interandina, mientras que la producción de carne representa el 43% y se realiza principalmente en las zonas templadas o tropicales de la Costa y Amazonía ecuatoriana. La producción lechera, en la región interandina se produce con una mayor concentración, teniendo un 73% de producción, mientras que en la Costa se tiene un 19% y en la Amazonía un 8%. En algunas zonas altas de la Sierra ecuatoriana, no existe mucha oportunidad de realizar actividades agrícolas, por lo que la leche se ha convertido en el único producto de venta estable y rentable para la población que se desarrolla en ese sector (Veterinaria Digital S.A., 2022).

La producción lechera es una actividad económica muy importante en el país por lo que se considera importante ayudar a su mejoramiento utilizando la ciencia de datos para generar conocimiento útil que proporcione indicadores que ayuden de forma positiva a la producción mediante el descarte oportuno de animales tomando en cuenta algunos factores como: animales enfermos, animales de edad avanzada o machos que no aportan en la producción.

En zonas dedicadas a la producción lechera el principal objetivo de la crianza de estos animales es sacar una cantidad óptima de leche y que sea de muy buena calidad, a un costo que sea

rentable. De igual forma se busca conseguir los animales necesarios para crianza, venta y reemplazos cuando se realiza un descarte.

El ganado vacuno es de mucha importancia en el sector ganadero básicamente por las siguientes razones:

- Son animales rumiantes, lo que significa que pueden consumir forrajes y subproductos agrícolas los cuales no son aptos para el consumo humano.
- Poseen una gran capacidad para producir leche en cantidades significativas (Agrobanco, 2012).

La producción lechera se centra en la crianza y manejo del ganado vacuno, este proceso muchas veces implica tener el conocimiento de algunos factores que pueden influir en la producción como: alimentación, salud del animal, ordeño, condiciones ambientales y prácticas.

Mediante la incorporación de modelos de clasificación, el manejo y procesamiento de datos se puede explicar de mejor forma estos factores permitiendo generar conocimiento útil para la toma de decisiones, la implementación de un modelo clasificador puede jugar un papel muy importante en la optimización de la producción lechera dentro de la hacienda volviendo su producción más eficiente y por consiguiente a la hacienda más competitiva dentro del mercado.

1.4 Objetivos

1.4.1 Objetivo General

Elegir el modelo clasificador que mejor se ajuste y ayude a identificar las principales causas de descarte, para ayudar a disminuir perdidas y optimizar la producción lechera en la hacienda del caso de estudio.

1.4.2 Objetivos Específicos

- Determinar el modelo clasificador que mejor se adapte a los datos, al modelo de negocio y que cumpla con el objetivo del proyecto.
- Realizar el preprocesamiento de la información obtenida de la hacienda con el fin de

encontrar información útil que pueda ser de ayuda para realizar los modelos predictivos y la toma de decisiones.

- Emplear la metodología CRISP-ML como parte del manejo y apoyo para cumplir con el objetivo del proyecto.

1.5 Alcance

El presente trabajo tiene como finalidad:

- Uso de los datos históricos de una hacienda lechera en donde se puede encontrar información acerca de la producción lechera y reproducción bovina correspondientes al periodo 1990-2019.
- Realizar la limpieza, verificación y transformación de los datos según la metodología CRISP-ML.
- Exploración y aplicación de diferentes algoritmos de clasificación (por ejemplo, árboles de decisión, SVM, redes neuronales).
- Identificar los principales indicadores o parámetros de producción y reproducción en la actividad pecuaria, basados en experiencia y estudios veterinarios.
- Aplicación de la metodología CRISP-ML para obtener resultados favorables en el desarrollo del proyecto.
- Se puede encontrar ausencia de datos, lo cual puede interferir con la exactitud de los valores generados mediante el estudio.

2. Marco Teórico

2.1. Producción Lechera

La producción lechera constituye un sistema comercial que implica la crianza, reproducción y gestión del ganado vacuno con el principal objetivo de producir leche. La leche considera a nivel mundial como un producto básico, se produce y consume en la mayoría de los países del mundo.

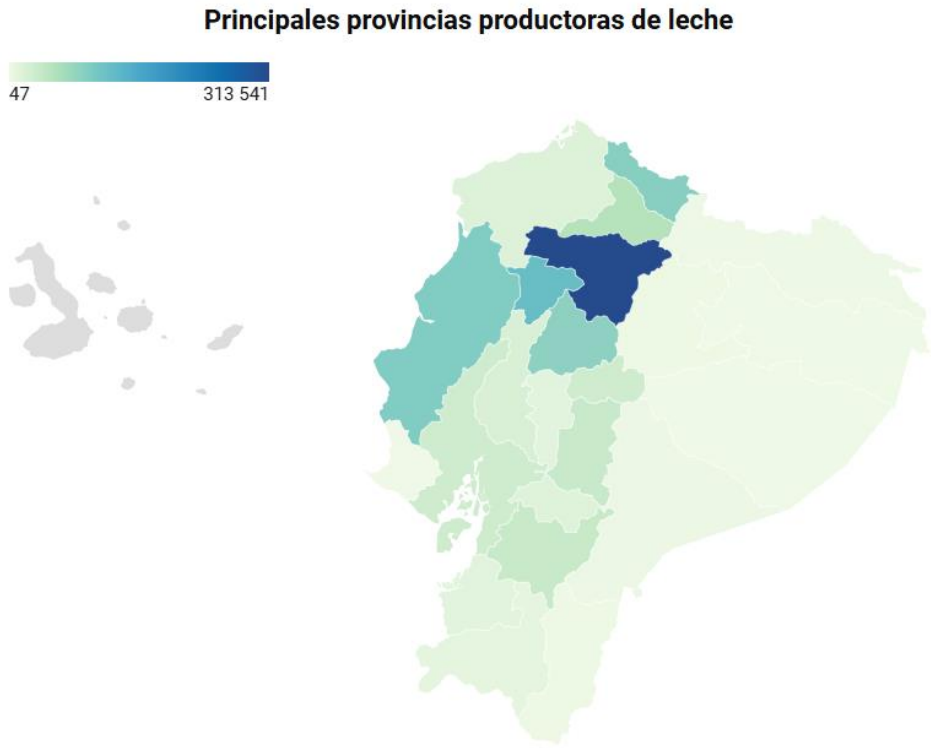
Aproximadamente 150 millones de familias alrededor del mundo se dedican a la producción lechera. En países en vías de desarrollo, la producción de la leche es generada por pequeños agricultores y contribuye a su sostenibilidad, la seguridad alimentaria y la nutrición de las familias. La leche es un producto que origina ganancias rápidas para los productores y es una fuente relevante de ingresos en efectivo (Producción Lechera, s. f.).

"En Ecuador la producción de leche es una de las actividades ganaderas más importantes, el país produce alrededor de 5,1 millones de litros/día de las tres regiones del país, Costa, Sierra y Amazonia" (INEC, 2018).

Tanto la ganadería como la producción lechera en Ecuador son significativas y constituyen una actividad económica relevante, principalmente para la región andina ya que las condiciones climáticas y geográficas favorecen a la cría de ganado lechero.

La región interandina y sus provincias son las que participan mayormente en la producción, comercio e industria de la leche en el país. Durante los últimos 50 años la provincia de Pichincha ha sido considerada como un sector del país que destaca por su tradición ganadera, que según el INEC (2016) posee la mayor producción de leche del Ecuador con un total del 15,9% a nivel nacional, aportando de esta forma a la economía del país (Cilecuador, 2019).

Figura 1. Principales provincias productoras de leche

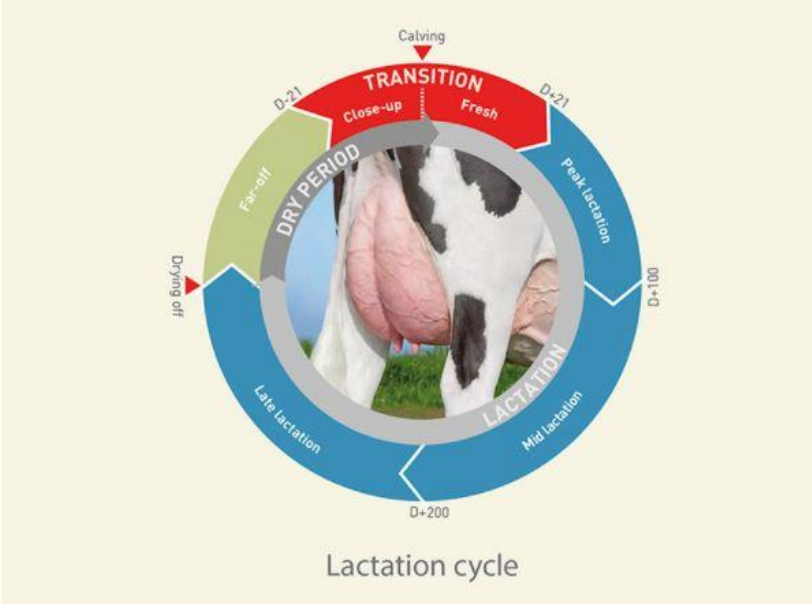


Fuente: (*Sector Lácteo En Ecuador: Una Mirada A los Desafíos y Oportunidades*, 2024)

2.1.1 Ciclo de Lactancia

El ciclo de lactancia de una vaca lechera es un proceso fisiológico que es clave en la producción lechera y que tiene algunas fases desde el parto hasta el secado. Abarca el periodo desde el parto hasta que la vaca entra nuevamente en un ciclo de inseminación para quedar nuevamente preñada, para los ganaderos es muy importante la optimización de este ciclo para mantener una producción lechera eficiente y saludable.

Figura 2. Ciclo de lactancia de una vaca lechera



Fuente: (Fussen, 2022)

Como se puede observar en la Figura 2, el ciclo de lactancia representa la producción de leche a lo largo del ciclo productivo de una vaca, el cual dura más o menos unos 305 días. El pico de lactancia es el nivel más alto de producción que una vaca puede alcanzar dentro de los primeros 90 días de lactación. Se evidencia una relación positiva entre el pico y la siguiente producción de leche a lo largo de la lactancia. En otras palabras, a medida que los litros de leche lleguen al pico, también se incrementan el total de litros producidos por lactancia. (Fussen, 2022)

2.1.2 Factores que influyen en la producción lechera

Existen algunos factores que pueden influir en la producción lechera:

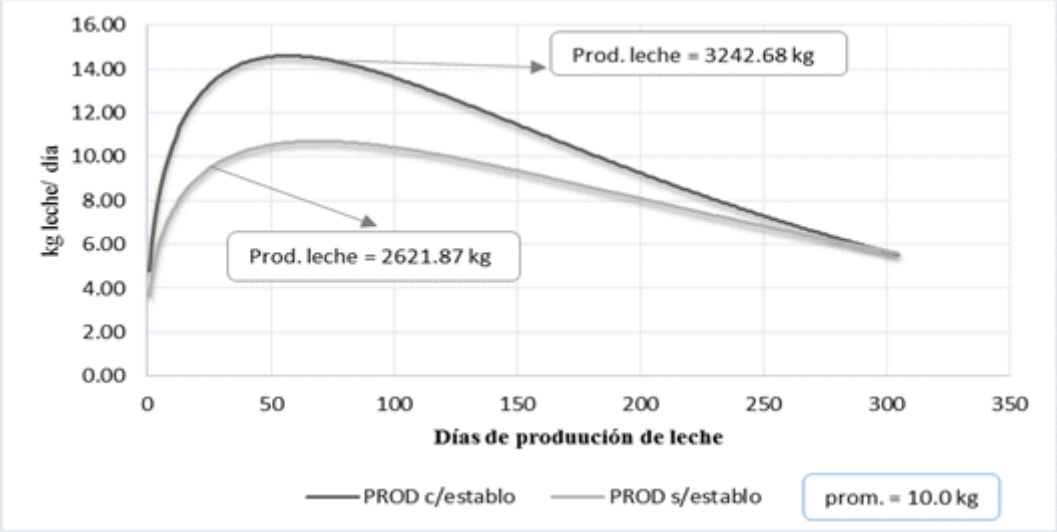
- **Raza**: Dependiendo de la raza se tienen distintos niveles de producción de leche y tiempos de lactancia. Algunas razas como la Holstein son especializadas en producción lechera y tienen un ciclo más corto entre períodos de lactancia.
- **Alimentación**: El ganado debe tener una dieta adecuada y equilibrada para que el rendimiento lechero sea optimo durante todo el ciclo de lactancia.
- **Manejo y salud**: Las vacas que cuentan con una atención sanitaria apropiada, tienen ciclos de lactancia más productivos. La presencia de enfermedades o infecciones puede afectar la producción de leche y la salud de la vaca.

- **Condiciones ambientales**: Existen algunos factores externos que pueden afectar el ciclo de lactancia como el clima, la temperatura y el estrés ambiental.

La cantidad de leche producida por una vaca durante su periodo de lactancia depende de algunos factores como: la raza, la edad, la alimentación, el clima y el bienestar animal.

- Una vaca Jersey produce entre 18 y 20 litros de leche al día, mientras que una vaca Girolando produce entre 9 y 12 litros.
- La edad constituye en un factor determinante en la producción de leche ya que la misma disminuye a medida que la vaca envejece.
- Una buena alimentación asegura un buen estado de salud y por consiguiente una maximización de la producción lechera.
- Las vacas pueden reducir su producción de leche en un 10-20% durante los días calurosos.

Figura 3. Días de producción vs kg/leche día



Fuente: (Apaza-Huallpa et al., s. f.-a)

2.2. Descarte Vacuno

"Vaca de descarte, es definida como aquel animal que es removido del rebaño por venta (individuo que se comercializa vivo y cuyo destino es otro predio o feria), envió a matadero o muerte del animal" (Fetrow y Col, 2006).

El periodo de vida productiva de una vaca lechera constituye la etapa comprendida entre el primer parto y la eliminación del animal del rebaño (Battistini, 1958), a esto se le define como longevidad y producción vitalicia (Maldonado,1991).

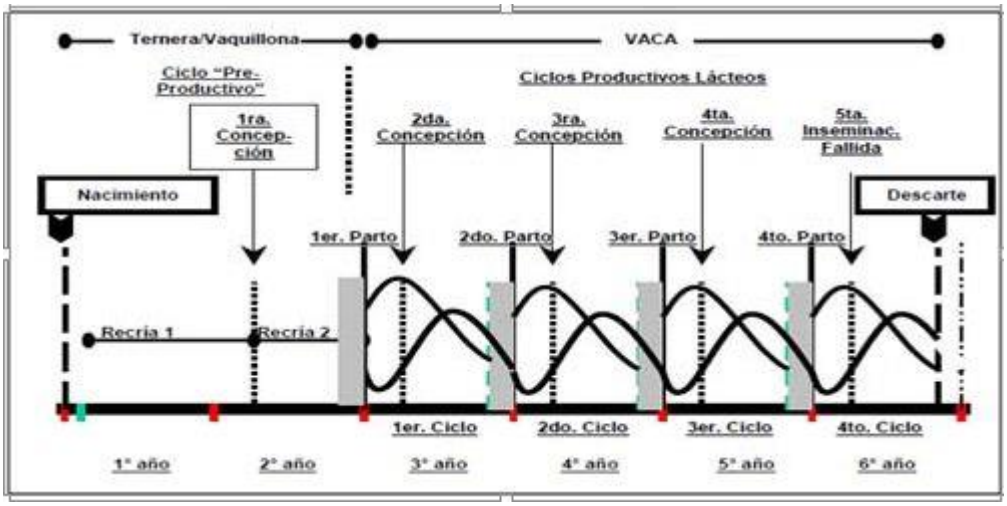
En haciendas lecheras, el acto de retirar o descartar una vaca es una práctica bastante normal, por lo que, es importante aclarar que en el descarte se incluye vacas de edad avanzada, con mala condición física y con problemas en las pezuñas, la ubre o reproductivos, lo que las convierte en un grupo de bajo rendimiento o vulnerables dentro del hato.

2.2.1 Vida productiva de una vaca

La vida productiva de un animal dentro de una hacienda lechera suele dividirse en dos fases: crianza y producción. La fase de crianza comienza con el nacimiento del animal hasta su primer parto, mientras que la fase productiva empieza desde el primer parto del animal hasta su descarte o muerte.

La vida útil de una vaca por lo general es medida en base a la cantidad de ciclos productivos como los que se muestran en la figura.

Figura 4. Ciclo productivo de una vaca lechera



Fuente: (Bovina, 2020)

Un animal que pertenece a un hato lechero por lo general tiene las siguientes características en cuanto a los tiempos en que cada fase de su vida productiva debe llevarse a cabo como: edad primera concepción (15 - 21 meses), edad primer parto (24 - 30 meses), duración de intervalo parto-concepción (3 - 5 meses), duración de CPL (10 - 12 meses); es bastante común que se considere que las vacas que han llegado a sus siete años hayan completado cuatro ciclos productivos. Sin embargo, pueden existir factores que ayudan a motivar que haya vacas que terminen su vida útil productiva con menos o con más de cuatro ciclos productivos (Bovina, 2020).

La incapacidad reproductiva es considera como uno de los principales factores para llevar a cabo el descarte de ganado vacuno, otro motivo importante es también la reducción del número promedio de lactancias para ordeño. No obstante, está comprobado que mejorar la fertilidad del ganado vacuno aumenta la producción lechera y se reduce las tasas de descarte (Phileo by Lesaffre, 2020).

Las vacas lecheras pueden ser descartadas en cualquier momento debido a múltiples factores involuntarios como: problemas de salud, muerte e infertilidad; o elección del ganadero, como un bajo rendimiento productivo o venta para reproducción. (Bovina, 2020).

2.2.1 Causas del Descarte

Según Orrego, Delgado y Echevarría (2003), las principales causas de descarte de ganado vacuno son:

1. Trastornos reproductivos: Entre los que se incluyen abortos, enfermedades en el aparato reproductivo, fetos momificados e incapacidad para concebir.
2. Problemas de ubre: Incluyen enfermedades relacionadas con las ubres y ubres con ligamentos vencidos.
3. Traumatismos: Incluyen cojeras, infecciones de la pezuña y problemas en la cadera.
4. Problemas peri pártales: Incluyen algunos trastornos de tipo metabólicos, partos distócicos y complicaciones secundarias.
5. Riesgo de muerte: Enfermedades que pueden causar el deceso del animal, leucosis enzoótica bovina, trastornos del corazón, pericarditis, neumonía, infecciones sistémicas con mastitis, toxemias y peritonitis aguda.
6. Otros: Enfermedades como tuberculosis, timpanismos, insuficiencia cardiaca congestiva, intoxicaciones, enfermedades hereditarias. Se incluye también, animales

que han sido vendidos para recría, intervenidos por cirugía, o que tienen una baja producción lechera.

2.2.2 Impacto del Descarte Vacuno

El descarte vacuno tiene múltiples implicaciones para el sector ganadero y su entorno. Algunos de los impactos son:

- **Impacto económico**: El costo de mantener animales no productivos o de bajo rendimiento puede ser elevado. Además, los costos de alimentación, cuidados veterinarios y otros gastos asociados no se recuperan si los animales no aportan en la productividad.
- **Impacto en la productividad**: El descarte oportuno y efectivo de animales con poca eficiencia productiva permite mejorar la gestión de la explotación ganadera. Al reducir el número de animales no rentables, los ganaderos pueden centrarse en aquellos que ofrecen un mayor rendimiento en términos de carne, leche o reproducción.
- **Impacto ambiental**: El descarte vacuno también puede tener efectos en el medio ambiente, ya que la sobrepoblación en ciertas explotaciones puede generar una mayor demanda de recursos, como el agua y los alimentos, y contribuir a la sobreexplotación de estos recursos.

2.3 Machine Learning

Machine learning o aprendizaje automático constituye un área de estudio que confiere a las maquinas o computadoras la capacidad de aprender sin requerir de un proceso de programación, permite aprender a partir de datos de entrenamiento para automatizar el proceso de construcción de modelos analíticos y resolver tareas asociadas, su aplicación está enfocada en enseñar a las máquinas como gestionar los datos de manera eficiente (Mahesh, 2018).

Es una subárea de la inteligencia artificial (IA) que se centra en desarrollar modelos y algoritmos que permiten a las computadoras aprender a partir de datos, para realizar una tarea específica, sin la necesidad de tener algún tipo de programación explícita. En el sector ganadero, el aprendizaje automático ofrece una herramienta poderosa para mejorar la toma de decisiones, optimizar los procesos productivos y garantizar el bienestar de los animales.

2.3.1 Aprendizaje Supervisado

Machine Learning Supervisado o Aprendizaje Supervisado, se destaca por el uso de conjuntos de datos etiquetados que se utilizan para entrenar a los algoritmos para que estos clasifiquen datos o realicen predicciones con precisión. A medida que el modelo recibe nuevos datos, ajusta sus ponderaciones hasta alcanzar un nivel óptimo de precisión. El aprendizaje supervisado sirve de mucha ayuda principalmente a las organizaciones ya que ayuda a resolver múltiples problemas presentes en la vida real, como clasificar correos no deseados en una carpeta separada de su bandeja de entrada (Janiesch et al., 2021).

El aprendizaje supervisado se emplea para construir modelos de clasificación aplicados a varios problemas, en este caso es de mucha utilidad para el descarte de vacunos, para esto se debe contar con un conjunto de datos en donde se tengan características como la edad, el peso, el rendimiento productivo, entre otras variables, el modelo aprende a identificar patrones y tendencias que puedan influir en la decisión de descarte.

El modelo evalúa la relación entre las características del animal y la clase de descarte durante el proceso de entrenamiento, a medida que el modelo procesa más información se van ajustando sus parámetros para mejorar su capacidad de predicción en nuevos casos no vistos. Los algoritmos comunes utilizados en aprendizaje supervisado incluyen Máquinas de Soporte Vectorial, Árboles de Decisión, Regresión Logística y Random Forest.

2.3.2 Modelos de Clasificación

Los algoritmos de clasificación en Machine Learning son técnicas de aprendizaje supervisado que se utilizan para asignar etiquetas o clases a nuevas instancias basándose en patrones aprendidos a partir de datos previamente etiquetados. Entre los algoritmos más comunes se encuentran los árboles de decisión, las máquinas de soporte vectorial, la regresión logística, el K-NN y las redes neuronales artificiales (Alpaydin, 2020).

Existen algunos tipos de algoritmos de clasificación, entre los más comunes se encuentran: Árboles de decisión, Random Forest, Máquinas de Soporte Vectorial (SVM), Regresión Logística, K-Vecinos Más Cercanos (K-NN), Redes Neuronales Artificiales

2.3.2.1 Random Forest

Es un algoritmo de aprendizaje automático supervisado el cual es ampliamente usado para resolver problemas tanto de clasificación y como de regresión, esta constituido por árboles de decisión que surgen a partir de distintas muestras y toman un voto predominante para decidir el promedio en caso de una regresión o una clasificación.

Una de las cualidades más relevantes de este algoritmo es que puede gestionar un conjunto de datos el cual puede contener variables continuas, como en el caso de la regresión, y variables categóricas, como en el caso de la clasificación, por lo que brinda mejor precisión para para problemas de relacionados con la clasificación (Inesdi, n.d.).

Es conocido por su robustez y versatilidad, ya que puede manejar tanto datos estructurados como no estructurados, y tiene la capacidad de evaluar la importancia de las características del modelo. Al ser un modelo basado en un ensemble de árboles, Random Forest combina los resultados de varios clasificadores débiles para formar un clasificador más fuerte, lo que mejora su capacidad para generalizar a nuevos datos.

2.3.2.2 Máquinas de Soporte Vectorial (SVM)

SVM o Máquinas de Soporte Vectorial buscan un hiperplano que maximice el margen entre las clases, y utilizan un truco de kernel para permitir la separación en espacios de características no lineales (Cortes & Vapnik, 1995).

Es muy efectivo en espacios de alta dimensión y es particularmente útil para problemas de clasificación binaria. Sin embargo, también se puede extender a problemas de clasificación multiclase mediante técnicas como la estrategia "uno contra todos" o "uno contra uno".

2.3.2.3 Regresión Logística

Es un algoritmo cuya técnica se basa en la analítica de datos ya que utiliza matemáticas para encontrar relaciones entre dos variables en los datos. Esta relación ayuda a estimar el valor de uno de los factores a partir del otro. Por lo general una predicción ofrece un conjunto finito de resultados, como por ejemplo una respuesta afirmativa o negativa. (Amazon Web Services, Inc., n.d.).

La fórmula básica de la regresión logística es una función sigmoide, también conocida como función logística, que transforma la salida de la combinación lineal de las características en un valor entre 0 y 1. Este valor se interpreta como la probabilidad de que una instancia pertenezca a la clase positiva. Si la probabilidad es mayor que un umbral predeterminado (generalmente 0.5), la instancia se clasifica como parte de la clase positiva; de lo contrario, se clasifica en la clase negativa.

2.3.3 Métricas de Clasificación

La evaluación de los modelos de clasificación es un paso importante para medir el rendimiento, la eficacia y la calidad de un modelo en predecir las clases correctas. Existen algunas métricas que permiten evaluar cómo un clasificador ha desempeñado su tarea de clasificación, cada una con su utilidad dependiendo del problema y los datos. Las principales métricas son:

- **Accurancy**: La precisión es la proporción de predicciones correctas, tanto para las instancias positivas como negativas, con respecto al número total de instancias (Zhang, 2004).
- **Recall**: Es una métrica fundamental en problemas donde la prioridad es identificar correctamente las instancias positivas, especialmente cuando los falsos negativos son costosos o peligrosos (Powers, 2011).
- **F1-Score**: es una medida combinada de precisión y recall, diseñada para proporcionar un balance entre ambas métricas, especialmente en situaciones donde el desbalance de clases puede afectar el rendimiento" (Sokolova & Lapalme, 2009).
- **ROC – AUC**: La curva ROC y el área bajo la curva (AUC) son métricas esenciales para evaluar modelos de clasificación, ya que permiten medir el rendimiento del modelo sin depender de un umbral específico de clasificación (Fawcett, 2006).

2.3.4 Validación Cruzada

“Es una técnica fundamental que ayuda a evaluar la capacidad de generalización de un modelo y se usa ampliamente para prevenir el sobreajuste y maximizar el uso de los datos disponibles" (Arlot & Celisse, 2010)

Es utilizada con el fin de evaluar el rendimiento de un modelo de machine learning y para asegurar que el modelo generalice bien a nuevos datos. Esta técnica se encarga de dividir el conjunto de datos en varias partes (o folds) y utiliza diferentes combinaciones de estos para entrenar y validar el modelo. El proceso más común es la validación cruzada k-fold, donde los datos se dividen en k subconjuntos.

2.3.4.1 La validación cruzada estratificada K-Fold

Es una técnica que garantiza que cada pliegue del conjunto de datos contenga aproximadamente el mismo porcentaje de muestras de cada clase que el conjunto completo, lo que hace que el proceso de entrenamiento y validación sea justo y más confiable (Olamendy, n.d.).

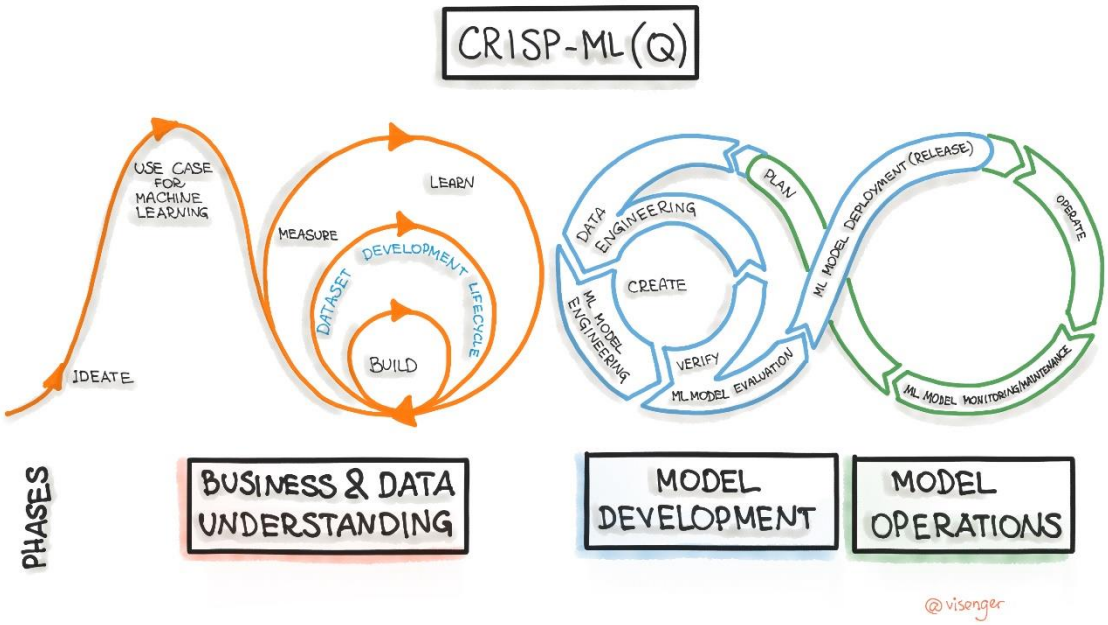
En la validación cruzada estratificada, se asegura que cada uno de los k folds tiene una distribución de clases similar a la del conjunto de datos original. Es decir, las proporciones de las clases en cada fold son aproximadamente iguales a las proporciones en el conjunto de datos completo. Esto mejora la evaluación del modelo y previene que las particiones aleatorias puedan generar resultados sesgados, especialmente en problemas donde una clase es mucho más prevalente que la otra.

2.4 Metodología CRISP- ML

Es un proceso que ha sido estandarizado para el desarrollo de aplicaciones de machine learning en donde se incluye una metodología de aseguramiento de calidad, está diseñado para guiar a los proyectos relacionados con machine learning a través de todas las fases del ciclo de vida que comprenden un proyecto de machine learning.

Los flujos de aprendizaje automático son iterativos y exploratorios, por lo que, dependiendo de los resultados de las fases posteriores, se puede volver a examinar los pasos anteriores.

Figura 5. Proceso de ciclo de vida de CRISP-ML



Fuente: (*Ml-ops.org*, 2024)

En general, el modelo de proceso CRISP-ML(Q) describe seis fases:

1. **Comprensión de datos y negocios**

   Definir e identificar claramente el problema que se resolverá con machine learning, alineado con los objetivos del negocio, el principal objetivo de esta fase es visualizar la viabilidad del proyecto, se recopila criterios de éxito junto con criterios relacionados con el negocio, estos criterios deben ser mensurables.

2. **Ingeniería de datos (preparación de datos)**

   En esta segunda fase se realiza la limpieza de los datos (tratamiento de valores faltantes, eliminación de duplicados), la transformación de los datos (como la normalización y *estandarización* de variables numéricas), codificación de variables categóricas y selección de características importantes necesarias para el entrenamiento futuro de los modelos. Además, se divide el conjunto de datos en conjuntos de entrenamiento y prueba.

3. **Ingeniería de modelos de aprendizaje automático**

Esta fase es primordial dentro del proceso de ML ya que se seleccionan y entrenan diversos algoritmos según el tipo de problema, por ejemplo: regresión lineal, árboles de decisión, redes neuronales, etc.

En general, la fase de modelado incluye *la selección del modelo*, el ajuste de los parámetros del modelo, el entrenamiento con los datos y la evaluación del rendimiento de cada modelo utilizando las métricas adecuadas (precisión, recall, F1-score, etc.).

4. Evaluación de modelos de aprendizaje automático

Luego de realizar la selección del modelo y su respectivo entrenamiento, el modelo debe ser evaluado con ayuda de un conjunto de pruebas, para brindar confianza y demostrar la solidez del mismo, se debe evaluar utilizando datos de entrada que sean ruidosos o erróneos, para determinar si el modelo es lo suficientemente robusto y generaliza bien a nuevos datos. Si el rendimiento no es adecuado, es posible que sea necesario volver a ajustar el modelo o los datos.

5. Despliegue

Luego de realizar la evaluación, el modelo debe ser implementado en un entorno de producción donde se pueda generar predicciones o insights útiles.

En esta fase se incluye la integración del modelo de ML en un sistema real, como una aplicación web, un proceso automatizado o un servicio que ofrezca predicciones en tiempo real.

6. Monitoreo y mantenimiento

Después de poner en producción un modelo de ML es fundamental monitorear el desempeño del modelo a lo largo del tiempo para asegurarse de que el modelo sigue siendo válido y efectivo, de ser necesario se debe realizar ajustes o actualizaciones para evitar la caída del desempeño del modelo.

3. Metodología

En el presente estudio se aplicará la metodología CRISP-ML para desarrollo y la implementación de un modelo clasificador para descarte vacuno. Esta metodología, básicamente posee un enfoque estructurado de CRISP-DM, lo que proporciona una guía bastante flexible y adaptable para la implementación de cualquier proyecto de aprendizaje automático, permitiendo una adecuada planificación desde la etapa de comprensión del problema hasta la evaluación del modelo. Al usar esta metodología se puede garantizar un proceso sistemático y replicable en la construcción del modelo clasificador.

3.1 Comprensión de Datos y Negocio

El descarte de ganado vacuno es un proceso en el que se identifican y separan animales que no cumplen con los estándares productivos o sanitarios, el principal objetivo es optimizar la producción y mejorar la rentabilidad del hato ganadero. Para llevar a cabo este proceso es importante considerar algunos aspectos productivos y reproductivos como: salud, edad, peso, producción lechera, partos, abortos, etc.

3.1.1 Comprensión de Negocio

Objetivos del negocio

- Maximizar la rentabilidad del hato lechero identificando los animales que son improductivos para asegurar que los costos de mantenimiento sean justificados por la producción.
- Optimizar la producción de leche dando de baja a vacas que tengan un bajo rendimiento para maximizar la producción promedio.
- Aumentar la eficiencia reproductiva descartando animales que presenten problemas de fertilidad.
- Mejorar la salud del hato en general descartando animales enfermos que puedan en un futuro contagiar al resto del hato.

Criterios de rendimiento

Obtener un modelo con precisión efectiva que permita realizar una adecuada predicción del ganado que debe ser descartado y facilitar la compresión de la relación entre variables.

Evaluación de la Situación

En el sector ganadero, el descarte vacuno es una práctica muy común y esencial que se enfoca en mantener un hato eficiente y rentable. Sin embargo, la decisión del descarte la mayoría de las veces suele basarse en la experiencia del ganadero, criterios subjetivos o recomendaciones de médicos veterinarios, lo que puede llevar a cometer errores como:

- No descartar animales improductivos, aumentando costos de alimentación y cuidado.
- Descartar animales aún rentables o útiles, perdiendo potencial productivo.

Entre los principales problemas que se puede evidenciar es que las decisiones de descarte que se toman de forma manual muchas veces pueden tomar bastante tiempo y ser propensas a errores, mantener vacas que son improductivas puede generar un impacto negativo en la rentabilidad.

El presente proyecto busca reducir los costos de mantenimiento al minimizar falsos negativos (vacas improductivas retenidas), en cuanto a los datos que se van a utilizar no tienen restricciones legales y serán anonimizados. Entre los riesgos que se pueden presentar durante la ejecución del proyecto se destaca la ausencia de datos en las variables de interés y un desbalance en las clases.

Traducir objetivos comerciales en objetivos de ML

- Desarrollar un modelo clasificador que permita identificar automáticamente las vacas que tienen que ser descartadas ya sea por su baja producción, problemas de salud, edad o problemas reproductivos.
- Optimizar las métricas de evaluación del modelo clasificador para minimizar los falsos positivos es decir descartar animales que sean útiles o los falsos negativos en donde se conservaría animales que sean poco productivos.
- Mostrar la importancia de cada característica en la decisión de descarte, lo que le permitirá al ganadero comprender las razones detrás de cada predicción.

Criterios de Rendimiento

En el presente proyecto las métricas de evaluación que se utilizarán son las siguientes:

- **Precisión (*Accuracy*):** Porcentaje de predicciones correctas.

- **Sensibilidad (*Recall*):** Minimiza los falsos negativos (*FN*), es decir, evita que animales improductivos sean retenidos.
- **Puntaje F1 (*F1-Score*):** Métrica adecuada cuando el conjunto de datos está desbalanceado.
- **Matriz de confusión:** Permite analizar en detalle los errores de clasificación (FP y FN).
- **ROC**: Mide la capacidad del modelo para distinguir entre los animales que necesitan ser descartados y los que no, en diferentes umbrales de decisión.

La selección del mejor modelo para el descarte vacuno se basará en la obtención de los valores más altos en dichas métricas.

Plan de proyecto

Tabla 1. Plan de Proyecto de Titulación

Fase	*Duración*	*Recursos*	*Riesgos*
Comprensión del negocio	1 semana	Responsable del proyecto	
Comprensión de los datos	2 semanas	Acceso a los datos históricos de los animales dados de baja.	
Preparación de los datos	4 semanas		Demasiados datos faltantes, desbalance de clases.
Modelado	4 semanas	Plataforma para el modelado (Jupyter Notebook)	
Evaluación	2 semanas		

3.1.2 Comprensión de Datos

Recopilación de los Datos

Registros correspondientes al periodo 1990-2019 almacenados en una base Visual FoxPro versión 9, el cual contiene los siguientes archivos en formato .csv.

- Cabezasexce
- HistoricoEXCE
- TipoBaja

- Reproducion
- Clinicoxml
- Producio
- Individual
- Lactancia

Los cuales fueron cargados en una Jupyter Notebook, para su revisión y procesamiento, previo al proceso de carga de los datos, se importan las librerías de pandas, numpy, seaborn y matplotlib.pyplot, para los siguientes procesos.

Descripción de los datos

La base de datos está compuesta por algunos archivos en formato .csv, los cuales aportan información acerca del ganado que ha sido dado de baja y de algunas características como los días de lactancia, enfermedad y producción lechera.

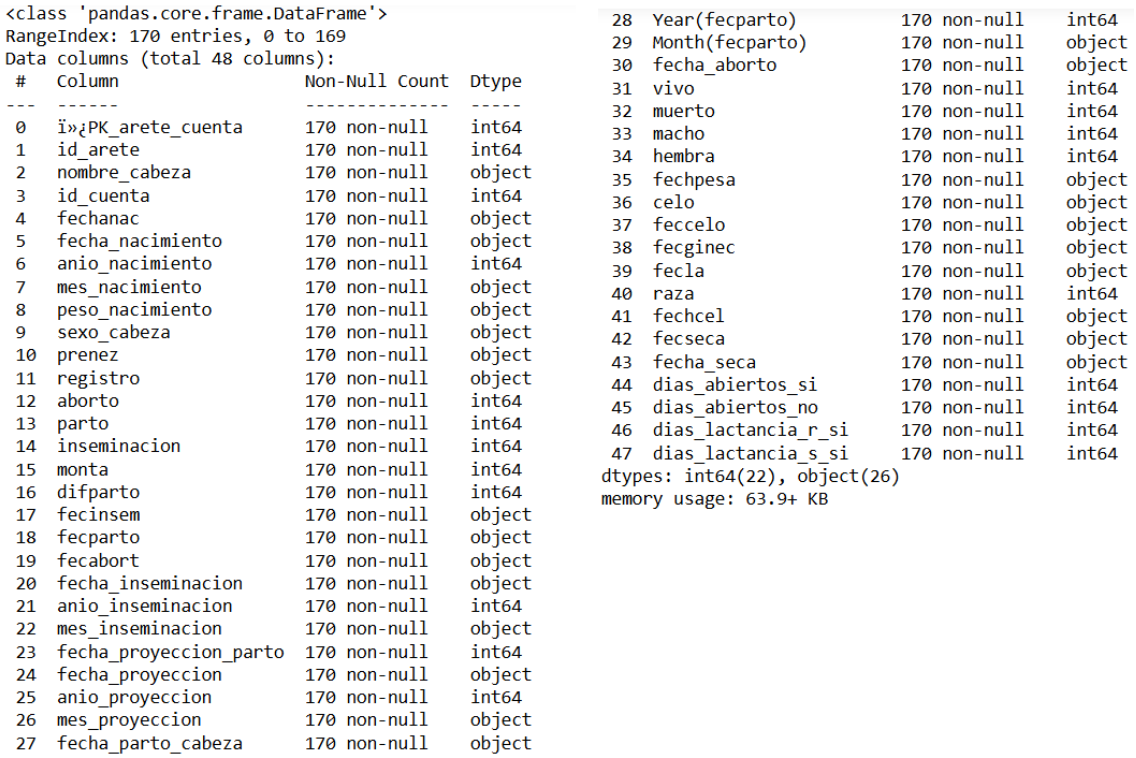
HistoricoEXCE.csv, el cual consta de 2010 registros y 59 variables: arete, nombre, cuenta, fechanac, pesonaci, sexo, preñez, aretepad, nombrpad, aretemad, nombrmad, registro, aborto, parto, insemina, monta, difparto, fecinsem, fecparto, fecabort, vivo, muerto, macho, hembra, observa, color, condicio, pesoulti, fechpesa, pajuela, celo, feccelo, fecginec, medicina, stvacia, cu, sm, ag, tl, fecla, total, numdipa,colarete, abuelop, abuelap, abuelom, abuelam, raza, foto, fechcel, pchequeo, tipbaja, valor, razón, fechbaja, edad, fingreso, costoc. En este data set se encuentra la información del ganado que ha sido dado de baja.

Figura 6. Tipo de datos HistoricoEXCE.csv

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2010 entries, 0 to 2009
Data columns (total 59 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   ï»¿codiemp  2010 non-null   int64
 1   arete       2010 non-null   object
 2   nombre      2010 non-null   object
 3   cuenta      2009 non-null   float64
 4   fechanac    2009 non-null   object
 5   pesonaci    2009 non-null   float64
 6   sexo        2009 non-null   object
 7   prenez      2009 non-null   object
 8   aretepad    2009 non-null   float64
 9   nombrpad    2009 non-null   object
 10  aretemad    2009 non-null   float64
 11  nombrmad    2009 non-null   object
 12  registro    2009 non-null   object
 13  aborto      2009 non-null   float64
 14  parto       2009 non-null   float64
 15  insemina    2009 non-null   float64
 16  monta       2009 non-null   float64
 17  difparto    2009 non-null   float64
 18  fecinsem    2009 non-null   object
 19  fecparto    2009 non-null   object
 20  fecabort    2009 non-null   object
 21  vivo        2009 non-null   float64
 22  muerto      2009 non-null   float64
 23  macho       2009 non-null   float64
 24  hembra      2009 non-null   float64
 25  observa     0 non-null      float64
 26  color       2009 non-null   object
 27  condicio    2009 non-null   object
 28  pesoulti    2009 non-null   float64
 29  fechpesa    2009 non-null   object
 30  pajuela     369 non-null    float64
 31  celo        2009 non-null   object
 32  feccelo     2009 non-null   object
 33  fecginec    2009 non-null   object
 34  medicina    2009 non-null   float64
 35  stvacia     2009 non-null   object
 36  cu          2009 non-null   object
 37  sm          2009 non-null   object
 38  ag          2009 non-null   object
 39  tl          2009 non-null   object
 40  fecla       2009 non-null   object
 41  total       2009 non-null   float64
 42  numdipa     2009 non-null   float64
 43  colarete    2009 non-null   object
 44  abuelop     2009 non-null   object
 45  abuelap     2009 non-null   object
 46  abuelom     2009 non-null   object
 47  abuelam     2009 non-null   object
 48  raza        2009 non-null   float64
 49  foto        2009 non-null   object
 50  fechcel     2009 non-null   object
 51  pchequeo    2009 non-null   object
 52  tipbaja     2009 non-null   float64
 53  valor       2009 non-null   float64
 54  razon       2002 non-null   object
 55  fechbaja    2000 non-null   object
 56  Edad        1 non-null      object
 57  fingreso    2000 non-null   object
 58  costoc      2000 non-null   float64
dtypes: float64(23), int64(1), object(35)
```

Cabezasexc.csv, el cual consta de 170 registros y 48 variables: id_arete, nombre_cabeza, id_cuenta, fechanac, fecha_nacimiento, anio_nacimiento, mes_nacimiento, peso_nacimiento, sexo_cabeza, preñez, registro, aborto, parto, inseminación, monta, difparto, fecinsem, fecparto, fecabort, fecha_inseminacion, año_inseminacion, mes_inseminacion, fecha_proyeccion_parto, fecha_proyeccion, año_proyeccion, mes_proyeccion, fecha_parto_cabeza, Year(fecparto), Month(fecparto), fecha_aborto, vivo, muerto, macho, hembra, fechpesa, celo, feccelo, fecginec, fecla, raza, fecseca, días_abiertos_si, días_abiertos_no, días_lactancia_r_si, días_lactancia_r_si. En este dataset se encuentra la información del ganado que aún no ha sido dado de baja.

Figura 7. Tipo de datos Cabezasexce.csv



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 170 entries, 0 to 169
Data columns (total 48 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   ï»¿PK_arete_cuenta      170 non-null    int64
 1   id_arete                170 non-null    int64
 2   nombre_cabeza           170 non-null    object
 3   id_cuenta               170 non-null    int64
 4   fechanac                170 non-null    object
 5   fecha_nacimiento        170 non-null    object
 6   anio_nacimiento         170 non-null    int64
 7   mes_nacimiento          170 non-null    object
 8   peso_nacimiento         170 non-null    object
 9   sexo_cabeza             170 non-null    object
 10  prenez                  170 non-null    object
 11  registro                170 non-null    object
 12  aborto                  170 non-null    int64
 13  parto                   170 non-null    int64
 14  inseminacion            170 non-null    int64
 15  monta                   170 non-null    int64
 16  difparto                170 non-null    int64
 17  fecinsem                170 non-null    object
 18  fecparto                170 non-null    object
 19  fecabort                170 non-null    object
 20  fecha_inseminacion      170 non-null    object
 21  anio_inseminacion       170 non-null    int64
 22  mes_inseminacion        170 non-null    object
 23  fecha_proyeccion_parto  170 non-null    int64
 24  fecha_proyeccion        170 non-null    object
 25  anio_proyeccion         170 non-null    int64
 26  mes_proyeccion          170 non-null    object
 27  fecha_parto_cabeza      170 non-null    object
 28  Year(fecparto)          170 non-null    int64
 29  Month(fecparto)         170 non-null    object
 30  fecha_aborto            170 non-null    object
 31  vivo                    170 non-null    int64
 32  muerto                  170 non-null    int64
 33  macho                   170 non-null    int64
 34  hembra                  170 non-null    int64
 35  fechpesa                170 non-null    object
 36  celo                    170 non-null    object
 37  feccelo                 170 non-null    object
 38  fecginec                170 non-null    object
 39  fecla                   170 non-null    object
 40  raza                    170 non-null    int64
 41  fechcel                 170 non-null    object
 42  fecseca                 170 non-null    object
 43  fecha_seca              170 non-null    object
 44  dias_abiertos_si        170 non-null    int64
 45  dias_abiertos_no        170 non-null    int64
 46  dias_lactancia_r_si     170 non-null    int64
 47  dias_lactancia_s_si     170 non-null    int64
dtypes: int64(22), object(26)
memory usage: 63.9+ KB
```

Clinicoxml.csv, el cual consta de 1951 registro y 10 variables: codiemp, arete, fecha, enfermed, medicina, cantidad, costo, días, columna1, unnamed. En este dataset se encuentra la información del número de veces que el animal se ha enfermado durante el último año, se identifica a cada animal por el número de arete.

Figura 8. Tipo de datos Clinicoxml.csv

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1951 entries, 0 to 1950
Data columns (total 10 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   codiemp     1951 non-null   int64
 1   arete       1951 non-null   object
 2   fecha       1951 non-null   object
 3   enfermed    1951 non-null   int64
 4   medicina    1942 non-null   float64
 5   cantidad    1933 non-null   float64
 6   costo       1919 non-null   float64
 7   dias        1306 non-null   float64
 8   Columna1    77 non-null     object
 9   Unnamed: 9  1 non-null      object
dtypes: float64(4), int64(2), object(4)
memory usage: 152.5+ KB
```

Individual.csv, el cual consta de 2313 registros y 5 variables: codiemp, arete, fecha, litros, precio. En este dataset se encuentra la cantidad de producción lechera de cada animal identificado por el número de arete.

Figura 9. Tipo de datos Individual.csv

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2313 entries, 0 to 2312
Data columns (total 5 columns):
 #   Column   Non-Null Count  Dtype
---  ------   --------------  -----
 0   codiemp  2313 non-null   int64
 1   arete    2313 non-null   object
 2   fecha    2313 non-null   int64
 3   litros   2313 non-null   int64
 4   precio   2313 non-null   float64
dtypes: float64(1), int64(3), object(1)
memory usage: 90.5+ KB
```

Reproducción.csv, el cual consta de 4044 registros y 9 variables: codiemp, arete, fecha, actual, próximo, días, cantidad, costo, pajuela. En este dataset se encuentra el número de días de lactancia de cada animal identificado por el número de arete.

Figura 10. Tipo de datos Reproducción.csv

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4044 entries, 0 to 4043
Data columns (total 9 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   codiemp   4044 non-null   int64
 1   arete     4044 non-null   object
 2   fecha     4044 non-null   int64
 3   actual    4044 non-null   int64
 4   proximo   4044 non-null   int64
 5   dias      4044 non-null   int64
 6   cantidad  4044 non-null   int64
 7   costo     4044 non-null   float64
 8   pajuela   4043 non-null   object
dtypes: float64(1), int64(6), object(2)
memory usage: 284.5+ KB
```

TipoBaja.csv, el cual consta de 6 registros y 3 variables: codiemp, tipbaja y descripcio. En este dataset se encuentra la información acerca del tipo de baja de ganado.

Figura 11. Tipo de datos TipoBaja.csv

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6 entries, 0 to 5
Data columns (total 3 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   codiemp     6 non-null      int64
 1   tipbaja     6 non-null      int64
 2   descripcio  6 non-null      object
dtypes: int64(2), object(1)
memory usage: 272.0+ bytes
```

3.2 Ingeniería de Datos

3.2.1 Selección de datos

La calidad y relevancia de los datos influyen directamente en el desempeño del modelo de clasificación. Para este estudio, la selección de datos se realizó con base a criterios específicos para garantizar que las variables utilizadas aporten información relevante en la identificación de los animales que serán descartados.

Carga de datos de Ganado dado de baja

Como parte del análisis de descarte vacuno es importante contar con información acerca de los animales descartados y las características que influyen en su separación del hato productivo para lo cual se carga el dataset historicoEXCE.csv que contiene la información de los animales que han sido dados de baja por distintos motivos entre los cuales se encuentra el descarte.

Figura 12. Carga de Datos historicoEXCE.csv

Carga historico

```
df = pd.read_csv('C:\GANADERIA\historicoEXCE.csv', header=0, sep=None, engine='python')
df.head()
```

	ï»¿codiemp	arete	nombre	cuenta	fechanac	pesonaci	sexo	prenez	aretepad	nombrpad	...	foto	fechcel	pchequeo	tipbaja	valor	
0	1	1	OJERAS	1.0	09/09/1996	50.0	MACHO		0.0		...	...	30/12/1899		3.0	0.0	
1	1	309	VANIDOSA	1.0	03/08/1995	0.0	HEMBRA		0.0	SHOREMAR	...	...	30/12/1899		2.0	104.0	SEPTI CA
2	1	9002	B	1.0	30/09/1999	32.0	MACHO		0.0	BUCKEYE	...	...	30/12/1899		6.0	7.0	
3	1	371	PITUFA	1.0	22/09/1996	32.0	HEMBRA		0.0	MASTHER	...	...	30/12/1899		1.0	80.0	PRODU
4	1	271	REINA	1.0	06/12/1994	37.0	HEMBRA		0.0	PULSER	...	...	30/12/1899		1.0	120.0	PRODU

5 rows × 59 columns

Una vez cargado el dataset se procede a escoger las variables relevantes para el estudio entre las cuales se tiene:

- **Arete (arete):** Corresponde al número de identificación del animal.
- **Fecha de nacimiento (fechanac):** Corresponde a la fecha de nacimiento del animal.
- **Peso (pesonaci):** Corresponde al peso de nacimiento del animal.
- **Sexo (sexo):** Corresponde al sexo del animal puede ser macho o hembra.
- **Registro (registro):** Corresponde a la raza del animal, entre las razas se tiene: Holstein Friesian (HF), Normanda, Jersey y Jersey F1 (cruce entre Holstein y Jersey).
- **Aborto (aborto):** Número de abortos registrados por cada animal.
- **Parto (parto):** Número de partos registrados por cada animal.
- **Tipo de Baja (tipbaja):** Tipo de baja del animal, entre las principales causas se tiene: descarte, enfermedad, muerte, obsequio, reproducción y venta.

Figura 13. Selección de características historicoEXCE.csv

```
historico_columnas = ['arete', 'fechanac','pesonaci','sexo','registro','aborto','parto',
                      'tipbaja','fechbaja']
historico_descarte = df[historico_columnas]
```

```
historico_descarte.head()
```

	arete	fechanac	pesonaci	sexo	registro	aborto	parto	tipbaja	fechbaja
0	1	09/09/1996	50.0	MACHO	HF	0.0	0.0	3.0	17/09/1999
1	309	03/08/1995	0.0	HEMBRA	HF	2.0	2.0	2.0	23/09/1999
2	9002	30/09/1999	32.0	MACHO	JERSEY F1	0.0	0.0	6.0	NaN
3	371	22/09/1996	32.0	HEMBRA	JERSEY F1	3.0	2.0	1.0	08/10/1999
4	271	06/12/1994	37.0	HEMBRA	HF	2.0	2.0	1.0	08/10/1999

Selección de hembras

Se observa que en el nuevo dataset generado llamado histórico_descarte, se tiene registros en donde existe información de ganado vacuno macho, los cuales no son considerados importantes para el estudio ya que el estudio se centra en el descarte de vacas lecheras, por lo que se procederá a seleccionar solo hembras.

Figura 14. Selección solo hembras histórico descarte

```
historico_hembras = historico_descarte.loc[historico_descarte['sexo'] == 'HEMBRA']
historico_hembras.head()
```

	arete	fechanac	pesonaci	sexo	registro	aborto	parto	tipbaja	fechbaja	edad
1	309	1995-03-08	0.0	HEMBRA	HF	2.0	2.0	2.0	1999-09-23	4.544832
3	371	1996-09-22	32.0	HEMBRA	JERSEY F1	3.0	2.0	1.0	1999-08-10	2.880219
4	271	1994-06-12	37.0	HEMBRA	HF	2.0	2.0	1.0	1999-08-10	5.160849
5	294	1995-08-05	36.0	HEMBRA	HF	0.0	2.0	1.0	1999-08-10	4.013689
6	323	1995-12-10	35.0	HEMBRA	NORMANDA	2.0	1.0	1.0	1999-08-10	3.665982

Cálculo de edad

Una vez filtrado el dataset con los registros de hembras, se procede a realizar el cálculo de la edad de cada vaca, para lo cual se utiliza las variables fechnac y fechbaja.

Figura 15. Cálculo edad histórico

```
from datetime import datetime
```

```
historico_descarte.loc[:, 'fechanac'] = pd.to_datetime(historico_descarte['fechanac'], errors='coerce')
historico_descarte.loc[:, 'fechbaja'] = pd.to_datetime(historico_descarte['fechbaja'], errors='coerce')
```

```
C:\Users\USUARIO\anaconda3\lib\site-packages\pandas\core\indexing.py:1745: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#
rsus-a-copy
  isetter(ilocs[0], value)
```

```
historico_descarte['edad'] = (historico_descarte['fechbaja'] - historico_descarte['fechanac']).dt.days / 365.25
```

```
<ipython-input-124-7b29babff55b>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-v
rsus-a-copy
  historico_descarte['edad'] = (historico_descarte['fechbaja'] - historico_descarte['fechanac']).dt.days / 365.25
```

Selección ganado mayor igual a dos años

Se aplica un filtro de edad para asegurar que los registros cuenten con un historial productivo y reproductivo relevante. Se excluyeron los registros de animales menores de dos años, ya que estos aún no han alcanzado su madurez productiva o reproductiva.

Con este procedimiento se evita sesgos derivados de animales que aún no han completado su desarrollo. En la Figura 16, se muestra cómo se realiza el filtrado mediante la siguiente instrucción:

Figura 16. Selección ganado histórico mayor a dos años

```
historico_hembras = historico_descarte.loc[historico_descarte['edad'] >= 2]
historico_hembras.head()
```

	arete	fechanac	pesonaci	sexo	registro	aborto	parto	tipbaja	fechbaja	edad
0	1	1996-09-09	50.0	MACHO	HF	0.0	0.0	3.0	1999-09-17	3.019849
1	309	1995-03-08	0.0	HEMBRA	HF	2.0	2.0	2.0	1999-09-23	4.544832
3	371	1996-09-22	32.0	HEMBRA	JERSEY F1	3.0	2.0	1.0	1999-08-10	2.880219
4	271	1994-06-12	37.0	HEMBRA	HF	2.0	2.0	1.0	1999-08-10	5.160849
5	294	1995-08-05	36.0	HEMBRA	HF	0.0	2.0	1.0	1999-08-10	4.013689

Selección por tipo de baja

Se observa que en el dataset se tiene la variable tipbaja el cual indica los motivos por el que el ganado ha sido dado de baja, por lo que es importante cargar el dataset TipoBaja para observar las razones por las cuales los animales han sido retirados del sistema productivo.

Figura 17. Carga dataset Tipo de Baja

	codiemp	tipbaja	descripcio
0	1	1	DESCARTE
1	1	2	ENFERMEDAD
2	1	3	MUERTE
3	1	4	OBSEQUIO
4	1	5	REPRODUCCION
5	1	6	VENTA

Una vez cargados los valores se observa que los descartes corresponden al tipbaja igual a 1, por lo que se procede a filtrar el dataset con los valores que tengan este valor para garantizar que el análisis se enfoque exclusivamente en los animales descartados por motivos productivos o sanitarios.

En la Figura 18, se procede a realizar la selección de ganado que ha sido dado de baja por descarte.

Figura 18. Filtro tipo de baja

```
historico_hembras = historico_hembras.loc[historico_descarte['tipbaja'] == 1]
historico_hembras.head()
```

	arete	fechanac	pesonaci	sexo	registro	aborto	parto	tipbaja	fechbaja	edad
3	371	1996-09-22	32.0	HEMBRA	JERSEY F1	3.0	2.0	1.0	1999-08-10	2.880219
4	271	1994-06-12	37.0	HEMBRA	HF	2.0	2.0	1.0	1999-08-10	5.160849
5	294	1995-08-05	36.0	HEMBRA	HF	0.0	2.0	1.0	1999-08-10	4.013689
6	323	1995-12-10	35.0	HEMBRA	NORMANDA	2.0	1.0	1.0	1999-08-10	3.665982
17	376	1996-05-10	34.0	HEMBRA	HF	1.0	1.0	1.0	1999-02-12	2.759754

```
historico_hembras.shape
```

```
(317, 10)
```

Luego de aplicar los filtros correspondientes se obtiene un dataset con 317 registros.

Estado de salud del ganado

Para complementar el dataset se debe incluir la información acerca del estado de salud del ganado vacuno, para lo cual se utiliza el dataset de enfermedades que contiene la información acerca de las patologías registradas en el ganado, lo que permite evaluar la relación entre la salud del ganado y su permanencia en el hato productivo.

En la figura 19, se procede a cargar el dataset clinicoxml.csv que contiene la información del estado de salud del hato.

Figura 19. Carga dataset Enfermedades

```
### Carga del datset que contiene la información del número de veces que una vaca se ha enfermado durante un año
clinico = pd.read_csv('C:\GANADERIA\clinicoxml.csv', header=0, sep=None, engine='python')
clinico.head(10)
```

	codiemp	arete	fecha	enfermed	medicina	cantidad	costo	dias	Columna1	Unnamed: 9
0	1	0	30/12/1899	0	0.0	0.0	0.0	0.0	NaN	NaN
1	1	412	30/09/1999	16	22.0	30.0	570.0	15.0	1	NaN
2	1	369	16/10/1999	4	25.0	5.0	1760.0	210.0	1	NaN
3	1	268	31/10/1999	23	34.0	1.0	20640.0	45.0	1	NaN
4	1	369	11/03/1999	14	24.0	10.0	0.0	2.0	2	NaN
5	1	369	11/03/1999	14	64.0	20.0	1800.0	2.0	3	NaN
6	1	360	11/06/1999	14	27.0	20.0	1180.0	1.0	1	NaN
7	1	3015	11/10/1999	7	23.0	200.0	200.0	45.0	NaN	NaN
8	1	3015	11/10/1999	7	47.0	10.0	2560.0	45.0	NaN	NaN
9	1	298	11/10/1999	8	40.0	10.0	616.0	45.0	1	NaN

Para evaluar la incidencia de enfermedades en el hato, se realiza un agrupamiento por número de arete, que funciona como identificador único de cada animal. Se procede a calcular la frecuencia con la que cada vaca ha sido diagnosticada con alguna enfermedad, sumando la cantidad de veces que ha presentado registros en el dataset de enfermedades. El procedimiento se implementa mediante la siguiente instrucción:

Figura 20. Frecuencia de enfermedades diagnosticadas por número de arete

```
### Agrupamiento por numero de arete y suma de la cantidad de veces que se ha enfermado una vaca
Enfermedades = pd.DataFrame(clinico)
enfermedadesConteo = Enfermedades['arete'].value_counts().reset_index()
enfermedadesConteo.columns = ['arete', 'enfermedades'] re
print(enfermedadesConteo)
```

```
     arete  enfermedades
0      489            18
1      123            15
2      417            14
3      360            13
4      109            12
..     ...           ...
393    909             1
394    923             1
395    128             1
396    563             1
397    947             1

[398 rows x 2 columns]
```

Se procede a realizar una integración de datos mediante una operación de left outer join entre el dataset principal y el dataset de enfermedades, como resultado se obtiene el siguiente dataset.

Figura 21. Left outer join enfermedades

```
# left outer join con las enfermedades
historico_hembras = pd.merge(historico_hembras, enfermedadesConteo, on='arete', how='left')
historico_hembras.head()
```

	arete	fechanac	pesonaci	sexo	registro	aborto	parto	tipbaja	fechbaja	edad	enfermedades
0	371	1996-09-22	32.0	HEMBRA	JERSEY F1	3.0	2.0	1.0	1999-08-10	2.880219	6.0
1	271	1994-06-12	37.0	HEMBRA	HF	2.0	2.0	1.0	1999-08-10	5.160849	9.0
2	294	1995-08-05	36.0	HEMBRA	HF	0.0	2.0	1.0	1999-08-10	4.013689	4.0
3	323	1995-12-10	35.0	HEMBRA	NORMANDA	2.0	1.0	1.0	1999-08-10	3.665982	5.0
4	376	1996-05-10	34.0	HEMBRA	HF	1.0	1.0	1.0	1999-02-12	2.759754	7.0

Días de lactancia

Para evaluar la duración de los días de lactancia, se incorpora el dataset Reproducción.csv que contiene información sobre los días de lactancia o la duración total del período de producción de leche en días de cada animal. Este dataset es muy relevante para analizar la relación entre la producción de leche y la probabilidad de descarte.

Figura 22. Carga dataset Días de lactancia

```
### Carga del dataset que contiene la información de los dias de lactancia de cada vaca durante un año
Reproduccion = pd.read_csv('C:\GANADERIA\Reproduccion.csv', header=0, sep=None, engine='python')
Reproduccion.head(10)
```

	codiemp	arete	fecha	actual	proximo	dias	cantidad	costo	pajuela
0	1	203	36419	1	7	60	0	1.8	3
1	1	272	42728	1	7	60	0	12.0	234
2	1	298	36424	4	8	281	0	0.0	0
3	1	395	36424	4	8	281	0	0.0	0
4	1	411	36424	4	8	281	0	0.0	0
5	1	417	36424	4	8	281	0	0.0	0
6	1	412	36388	1	7	60	0	1.1	7
7	1	448	36420	1	7	60	0	1.3	6
8	1	360	36426	6	7	7	0	0.0	0
9	1	412	36431	6	7	2	0	0.0	0

Una vez que se ha cargado el dataset se procede a calcular la duración total del período productivo de cada vaca o días de lactancia, para esto se realiza un agrupamiento por número de arete, sumando la cantidad total de días que ha estado en lactancia a lo largo de su último ciclo productivo.

Figura 23. Días de lactancia agrupados por número de arete

```
# Agrupamiento por número de arete y suma delos dias de lactancia de cada vaca
diasLactancia = pd.DataFrame(Reproduccion)
resultadoDiasLactancia = diasLactancia.groupby('arete')['dias'].sum().reset_index()
print(resultadoDiasLactancia)
```

```
          arete  dias
0                  21
1           100  2365
2          1000   338
3          1002   334
4           101   374
..          ...   ...
479       99532   257
480         996   436
481         997   452
482         998   567
483        A713   398

[484 rows x 2 columns]
```

Para evaluar la relación entre los días de lactancia y el descarte vacuno, se realiza una integración de datos mediante una operación de left outer join entre el dataset principal y el dataset de lactancia, mediante la siguiente instrucción.

Figura 24. Left outer join días de lactancia

```
# left outer join con los días de lactancia
historico_hembras = pd.merge(historico_hembras, resultadoDiasLactancia, on='arete', how='left')
historico_hembras.head()
```

	arete	fechanac	pesonaci	sexo	registro	aborto	parto	tipbaja	fechbaja	edad	enfermedades	dias
0	371	1996-09-22	32.0	HEMBRA	JERSEY F1	3.0	2.0	1.0	1999-08-10	2.880219	6.0	284.0
1	271	1994-06-12	37.0	HEMBRA	HF	2.0	2.0	1.0	1999-08-10	5.160849	9.0	1706.0
2	294	1995-08-05	36.0	HEMBRA	HF	0.0	2.0	1.0	1999-08-10	4.013689	4.0	862.0
3	323	1995-12-10	35.0	HEMBRA	NORMANDA	2.0	1.0	1.0	1999-08-10	3.665982	5.0	1059.0
4	376	1996-05-10	34.0	HEMBRA	HF	1.0	1.0	1.0	1999-02-12	2.759754	7.0	274.0

Producción lechera

Para evaluar la relación entre la producción lechera y el descarte vacuno, se incorpora el dataset Indivisual.csv que contiene la información de los litros de leche producidos durante un año por cada animal. Este dataset es importante para analizar el rendimiento productivo y su influencia en el descarte del ganado.

Figura 25. Carga dataset Producción lechera

```
### Carga del dataset que contiene la información de los litros de leche producidos por cada vaca durante un año
litrosProducidos = pd.read_csv('C:\GANADERIA\Individual.csv', header=0, sep=None, engine='python')
litrosProducidos.head(10)
```

	codiemp	arete	fecha	litros	precio
0	1	0	0	0	0.0
1	1	272	36419	20	2200.0
2	1	404	36434	108	2200.0
3	1	323	36434	124	2200.0
4	1	203	36434	240	2200.0
5	1	345	36434	119	2200.0
6	1	271	36434	144	2200.0
7	1	268	36434	191	2200.0
8	1	147	36434	14	2200.0
9	1	353	36434	186	2200.0

Para obtener el rendimiento productivo de cada vaca, se realiza un agrupamiento por número de arete, sumando la cantidad total de litros de leche producidos en los registros disponibles.

Figura 26. Producción lechera agrupada por número de arete

```
# Agrupamiento por número de arete y suma de los litros de leche producidos por cada vaca
litrosProducidos = pd.DataFrame(litrosProducidos)
resultadoLitros = litrosProducidos.groupby('arete')['litros'].sum().reset_index()
print(resultadoLitros)
```

```
     arete  litros
0        0       0
1      100    2965
2     1000    3400
3     1002    1883
4      101    1178
..     ...     ...
393  99532     753
394    996    1145
395    997     117
396    998    1454
397   A713     785

[398 rows x 2 columns]
```

Para incorporar la información de la producción lechera, se realiza una integración de datos mediante una operación de left outer join entre el dataset principal y el dataset de producción lechera, mediante la siguiente instrucción.

Figura 27. Left outer join producción lechera

```
# Left outer join con la producción lechera
historico_hembras = pd.merge(historico_hembras, resultadoLitros, on='arete', how='left')
historico_hembras.head()
```

	arete	fechanac	pesonaci	sexo	registro	aborto	parto	tipbaja	fechbaja	edad	enfermedades	dias	litros
0	371	1996-09-22	32.0	HEMBRA	JERSEY F1	3.0	2.0	1.0	1999-08-10	2.880219	6.0	284.0	948.0
1	271	1994-06-12	37.0	HEMBRA	HF	2.0	2.0	1.0	1999-08-10	5.160849	9.0	1706.0	3365.0
2	294	1995-08-05	36.0	HEMBRA	HF	0.0	2.0	1.0	1999-08-10	4.013689	4.0	862.0	1730.0
3	323	1995-12-10	35.0	HEMBRA	NORMANDA	2.0	1.0	1.0	1999-08-10	3.665982	5.0	1059.0	3382.0
4	376	1996-05-10	34.0	HEMBRA	HF	1.0	1.0	1.0	1999-02-12	2.759754	7.0	274.0	650.0

Inclusión del Campo "Descarte"

Se incorpora una nueva variable al dataset denominada descarte para identificar si un animal ha sido descartado, el cual toma el valor de "1" cuando una vaca ha sido retirada del hato productivo.

Figura 28. Inclusión campo descarte "SI"

```
### Inclusión campo descarte = SI
historico_hembras['descarte'] = 1
```

Nueva selección de características

Para preparar el dataset para los modelos predictivos, se realiza una nueva selección de características. Para lo cual se eliminan las variables que no aportan información relevante para el análisis del descarte vacuno.

Las variables que se mantienen son: arete, pesonaci, registro, aborto, parto, edad, enfermedades, días, litros y descarte.

Figura 29. Nueva selección de características dataset ganado descartado

```
### Nueva selección de características para el dataset
descartadas_columnas = ['arete','pesonaci','registro','aborto','parto','edad','enfermedades','dias','litros','descarte']
descartadas = historico_hembras[descartadas_columnas]
```

Carga del Dataset del Ganado No Descartado

Para completar el análisis es importante incluir información acerca del ganado que permanece en el hato productivo, para lo cual se carga el dataset Cabezasexc.csv que contiene únicamente la información de los animales que no han sido dados de baja. Este conjunto de datos es esencial para evaluar las características del ganado que continúa siendo productivo.

Figura 30. Carga dataset Ganado no descartado

```
df1 = pd.read_csv('C:\GANADERIA\Cabezasexc.csv', header=0, sep=None, engine='python')
df1.head()
```

eza	id_cuenta	fechanac	fecha_nacimiento	anio_nacimiento	mes_nacimiento	peso_nacimiento	sexo_cabeza	...	fecginec	fecla	raza	fechcel	fe
NIA	1	07/11/2008	07/11/2008	2008	nov.	34,0	HEMBRA	...	30/12/2018	30/12/1899	1	30/12/1899	30/12
NA	1	22/03/2009	22/03/2009	2009	mar.	35,0	HEMBRA	...	18/08/2019	30/12/1899	9	30/12/1899	30/12
SA	1	23/08/2009	23/08/2009	2009	ago.	35,0	HEMBRA	...	27/01/2019	30/12/1899	9	30/12/1899	30/12
NA	1	02/01/2010	02/01/2010	2010	ene.	35,0	HEMBRA	...	22/08/2018	30/12/1899	9	30/12/1899	30/12
NI	1	20/01/2010	20/01/2010	2010	ene.	35,0	HEMBRA	...	18/05/2019	30/12/1899	1	30/12/1899	30/12

Una vez cargado el dataset se procede a escoger las variables relevantes para el estudio entre las cuales son:

- **Arete (Id_arete):** Identificador único del animal.
- **Fecha de nacimiento (fechanac):** Fecha de nacimiento del animal.
- **Peso (peso_nacimiento**): Peso de nacimiento del animal al nacer.
- **Sexo(sexo_cabeza):** Sexo del animal.
- **Raza(registro):** Corresponde a la raza del animal, entre las razas se tiene: Holstein Friesian (HF), Normanda, Jersey y Jersey F1 (cruce entre Holstein y Jersey).

- **Abortos (aborto):** Registro del número de partos de cada animal.
- **Partos (parto):** Registro del número de abortos de cada animal.
- **Días de lactancia (dias_lactancia_r_si):** Días de lactancia de cada vaca.

Mediante la siguiente instrucción se realiza la selección de las variables de interés:

Figura 31. Selección de características dataset ganado no descartado

```
ganado_columnas = ['id_arete', 'fechanac','peso_nacimiento','sexo_cabeza','registro','aborto','parto',
                   'dias_lactancia_r_si']
ganado = df1[ganado_columnas]
```

```
ganado.head()
```

	id_arete	fechanac	peso_nacimiento	sexo_cabeza	registro	aborto	parto	dias_lactancia_r_si
0	31	07/11/2008	34,0	HEMBRA	HF	0	5	400
1	50	22/03/2009	35,0	HEMBRA	NORMANDA	1	6	450
2	74	23/08/2009	35,0	HEMBRA	NORMANDA	1	6	336
3	92	02/01/2010	35,0	HEMBRA	NORMANDA	1	6	387
4	96	20/01/2010	35,0	HEMBRA	HF	1	6	288

Cálculo de edad

Una vez filtrado el dataset con los registros de hembras, se procede a realizar el cálculo de la edad de cada vaca, para lo cual se utiliza las variables fechnac y Fecha_fija.

Figura 32. Cálculo edad dataset ganado no descartado

```
fecha_fija = datetime(2019, 2, 1)
ganado = ganado.assign(Fecha_Fija=fecha_fija)
```

```
ganado.loc[:, 'fechanac'] = pd.to_datetime(ganado['fechanac'], errors='coerce')
ganado['edad'] = (ganado['Fecha_Fija'] - ganado['fechanac']).dt.days / 365.25
ganado.head()
```

	id_arete	fechanac	peso_nacimiento	sexo_cabeza	registro	aborto	parto	dias_lactancia_r_si	Fecha_Fija	edad
0	31	2008-07-11	34,0	HEMBRA	HF	0	5	400	2019-02-01	10.559890
1	50	2009-03-22	35,0	HEMBRA	NORMANDA	1	6	450	2019-02-01	9.864476
2	74	2009-08-23	35,0	HEMBRA	NORMANDA	1	6	336	2019-02-01	9.442847
3	92	2010-02-01	35,0	HEMBRA	NORMANDA	1	6	387	2019-02-01	8.999316
4	96	2010-01-20	35,0	HEMBRA	HF	1	6	288	2019-02-01	9.032170

Selección hembras

El análisis debe enfocarse en las vacas hembras, por lo que se realiza una selección de registros correspondientes solo a hembras dentro del conjunto de datos.

Figura 33. Selección solo hembras dataset ganado no descartado

```
ganado = ganado.loc[ganado['sexo_cabeza'] == 'HEMBRA']
ganado.head()
```

	id_arete	fechanac	peso_nacimiento	sexo_cabeza	registro	aborto	parto	dias_lactancia_r_si	Fecha_Fija	edad
0	31	2008-07-11	34,0	HEMBRA	HF	0	5	400	2019-02-01	10.559890
1	50	2009-03-22	35,0	HEMBRA	NORMANDA	1	6	450	2019-02-01	9.864476
2	74	2009-08-23	35,0	HEMBRA	NORMANDA	1	6	336	2019-02-01	9.442847
3	92	2010-02-01	35,0	HEMBRA	NORMANDA	1	6	387	2019-02-01	8.999316
4	96	2010-01-20	35,0	HEMBRA	HF	1	6	288	2019-02-01	9.032170

Selección ganado mayor igual a dos años

Se aplica un filtro de edad para asegurar que los registros cuenten con un historial productivo y reproductivo relevante. Se excluyeron los registros de animales menores de dos años, ya que estos aún no han alcanzado su madurez productiva o reproductiva.

Con este procedimiento se evita sesgos derivados de animales que aún no han completado su desarrollo.

Figura 34. Selección de ganado mayor a dos años dataset ganado no descartado

```
ganado = ganado.loc[historico_descarte['edad'] >= 2]
ganado.head()
```

	id_arete	fechanac	peso_nacimiento	sexo_cabeza	registro	aborto	parto	dias_lactancia_r_si	Fecha_Fija	edad
0	31	2008-07-11	34,0	HEMBRA	HF	0	5	400	2019-02-01	10.559890
1	50	2009-03-22	35,0	HEMBRA	NORMANDA	1	6	450	2019-02-01	9.864476
3	92	2010-02-01	35,0	HEMBRA	NORMANDA	1	6	387	2019-02-01	8.999316
4	96	2010-01-20	35,0	HEMBRA	HF	1	6	288	2019-02-01	9.032170
5	102	2010-11-02	35,0	HEMBRA	HF	1	5	289	2019-02-01	8.249144

Estado de salud del ganado

Para evaluar la incidencia de enfermedades en el hato, se realiza un agrupamiento por número de arete, que funciona como identificador único de cada animal. Se procede a calcular la frecuencia con la que cada vaca ha sido diagnosticada con alguna enfermedad, sumando la cantidad de veces que ha presentado registros en el dataset de enfermedades.

Figura 35. Left outer join enfermedades dataset ganado no descartado

```
### left outer join con el número de veces que se ha enfermado una vaca durante un año
ganado = pd.merge(ganado, enfermedadesConteo, on='arete', how='left')
ganado.head()
```

	arete	fechanac	peso_nacimiento	sexo_cabeza	registro	aborto	parto	dias_lactancia	Fecha_Fija	edad	enfermedades
0	31	2008-07-11	34,0	HEMBRA	HF	0	5	400	2019-02-01	10.559890	1.0
1	50	2009-03-22	35,0	HEMBRA	NORMANDA	1	6	450	2019-02-01	9.864476	3.0
2	92	2010-02-01	35,0	HEMBRA	NORMANDA	1	6	387	2019-02-01	8.999316	4.0
3	96	2010-01-20	35,0	HEMBRA	HF	1	6	288	2019-02-01	9.032170	2.0
4	102	2010-11-02	35,0	HEMBRA	HF	1	5	289	2019-02-01	8.249144	3.0

Producción lechera

Para incorporar la información de la producción lechera, se realiza una integración de datos mediante una operación de left outer join entre el dataset principal y el dataset de producción lechera, mediante la siguiente instrucción.

Figura 36. Left outer join producción lechera dataset ganado no descartado

```
# left outer join con producción lechera por vaca
ganado = pd.merge(ganado, resultadoLitros, on='arete', how='left')
ganado.head()
```

	arete	fechanac	peso_nacimiento	sexo_cabeza	registro	aborto	parto	dias_lactancia	Fecha_Fija	edad	enfermedades	litros
0	31	2008-07-11	34,0	HEMBRA	HF	0	5	400	2019-02-01	10.559890	1.0	3322.0
1	50	2009-03-22	35,0	HEMBRA	NORMANDA	1	6	450	2019-02-01	9.864476	3.0	3060.0
2	92	2010-02-01	35,0	HEMBRA	NORMANDA	1	6	387	2019-02-01	8.999316	4.0	3815.0
3	96	2010-01-20	35,0	HEMBRA	HF	1	6	288	2019-02-01	9.032170	2.0	4752.0
4	102	2010-11-02	35,0	HEMBRA	HF	1	5	289	2019-02-01	8.249144	3.0	2868.0

Inclusión del Campo "Descarte"

Se incorpora una nueva variable al dataset denominada descarte para identificar si un animal ha sido descartado, el cual toma el valor de "0" que indica que una vaca no ha sido descartada es decir sigue siendo parte del hato productivo.

Figura 37. Inclusión campo descarte "NO"

```
##Creación campo descarte = NO
ganado['descarte'] = 0
ganado.head()
```

	arete	fechanac	peso_nacimiento	sexo_cabeza	registro	aborto	parto	dias_lactancia	Fecha_Fija	edad	enfermedades	litros	descarte
0	31	2008-07-11	34,0	HEMBRA	HF	0	5	400	2019-02-01	10.559890	1.0	3322.0	0
1	50	2009-03-22	35,0	HEMBRA	NORMANDA	1	6	450	2019-02-01	9.864476	3.0	3060.0	0
2	92	2010-02-01	35,0	HEMBRA	NORMANDA	1	6	387	2019-02-01	8.999316	4.0	3815.0	0
3	96	2010-01-20	35,0	HEMBRA	HF	1	6	288	2019-02-01	9.032170	2.0	4752.0	0
4	102	2010-11-02	35,0	HEMBRA	HF	1	5	289	2019-02-01	8.249144	3.0	2868.0	0

Nueva selección de características

Para preparar el dataset para los modelos predictivos, se realiza una nueva selección de características. Para lo cual se eliminan las variables que no aportan información significativa para el estudio del descarte vacuno.

Las variables que se mantienen son: arete, peso_nacimiento, registro, aborto, parto, edad, enfermedades, días_lactancia, litros y descarte.

Figura 38. Selección nuevas características ganado no descartado

```
### Selección de nuevas características
noDescartadas_columnas = ['arete','peso_nacimiento','registro','aborto','parto',
                          'dias_lactancia', 'enfermedades','litros','descarte','edad']
noDescartadas = ganado[noDescartadas_columnas]
```

Al finalizar con el proceso de selección de características se tienen dos dataset finales, Descartadas y noDescartadas.

3.2.2 Limpieza de datos

Dataset Descartadas

Como parte del proceso de limpieza, se realiza un renombramiento de campos para mejorar la claridad de los nombres de las columnas y facilitar su interpretación en los análisis y modelos predictivos.

Figura 39. Renombramiento campos dataset Descartadas

```
### Renombramiento de campos
descartadas = descartadas.rename(columns={'pesonaci': 'peso'})
descartadas = descartadas.rename(columns={'registro': 'raza'})
descartadas = descartadas.rename(columns={'aborto': 'abortos'})
descartadas = descartadas.rename(columns={'parto': 'partos'})
descartadas = descartadas.rename(columns={'dias': 'dias_lactancia'})
descartadas = descartadas.rename(columns={'litros': 'produccion'})
```

Para asegurar que los datos sean procesados correctamente por los modelos, se realiza un cambio de tipos de datos en las variables abortos y partos del dataset.

Figura 40. Cambio tipo de datos dataset Descartadas

```
###Cambio de tipos de datos
descartadas['abortos'] = descartadas['abortos'].astype(int)
descartadas['partos'] = descartadas['partos'].astype(int)
descartadas.head()
```

	arete	peso	raza	abortos	partos	edad	enfermedades	dias_lactancia	produccion	descarte
0	371	32.0	JERSEY F1	3	2	2.880219	6.0	284.0	948.0	1
1	271	37.0	HF	2	2	5.160849	9.0	1706.0	3365.0	1
2	294	36.0	HF	0	2	4.013689	4.0	862.0	1730.0	1
3	323	35.0	NORMANDA	2	1	3.665982	5.0	1059.0	3382.0	1
4	376	34.0	HF	1	1	2.759754	7.0	274.0	650.0	1

Se eliminan los registros duplicados en el dataset para asegurar que la información es única y precisa, para evitar que los duplicados afecten los resultados de los análisis y modelos predictivos.

Figura 41. Eliminar duplicados dataset Descartadas

```
### Quitar duplicados
descartadas = descartadas.drop_duplicates(subset='arete', keep='first')
```

Dataset noDescartadas

Como parte del proceso de limpieza, se realiza un renombramiento de campos para mejorar la claridad de los nombres de las columnas y facilitar su interpretación en los análisis y modelos predictivos.

Figura 42. Renombramiento campos dataset noDescartadas

```
### Renombramiento de campos
noDescartadas = noDescartadas.rename(columns={'peso_nacimiento': 'peso'})
noDescartadas = noDescartadas.rename(columns={'registro': 'raza'})
noDescartadas = noDescartadas.rename(columns={'aborto': 'abortos'})
noDescartadas = noDescartadas.rename(columns={'parto': 'partos'})
noDescartadas = noDescartadas.rename(columns={'dias': 'dias_lactancia'})
noDescartadas = noDescartadas.rename(columns={'litros': 'produccion'})
```

Para facilitar la interpretación de los datos, se realiza un cambio en el orden de los campos del dataset.

Figura 43. Cambio de orden dataset noDescartadas

```
### Cambio en el orden de los campos del dataset
nuevo_orden = ['arete', 'peso', 'raza','abortos','partos','edad','enfermedades','dias_lactancia','produccion','descarte']
noDescartadas = noDescartadas.reindex(nuevo_orden, axis=1)
```

Se eliminan los registros duplicados en el dataset para asegurar que la información es única y precisa, para evitar que los duplicados afecten los resultados de los análisis y modelos predictivos.

Figura 44. Eliminar duplicados dataset noDescartadas

```
### Quitar duplicados
noDescartadas = noDescartadas.drop_duplicates(subset='arete', keep='first')
```

Dataset Final

Durante el proceso de preparación de los datos, se trabajó con dos conjuntos de datos: uno que contiene la información de ganado descartado y otro que incluye a los animales no descartados.

El dataset final está constituido por ambos dataset los cuales contienen toda la información sobre el ganado tanto descartado como no descartado, lo que permite realizar un análisis integral y comparar las características de ambos grupos. La operación de unión se realizó utilizando la siguiente instrucción:

Figura 45. Dataset Final

```
### Union de la tabla de descartadas y no descartadas
final = pd.concat([descartadas, noDescartadas], axis=0)
final.head()
```

	arete	peso	raza	abortos	partos	edad	enfermedades	dias_lactancia	produccion	descarte
0	371	32.0	JERSEY F1	3	2	2.880219	6.0	284.0	948.0	1
1	271	37.0	HF	2	2	5.160849	9.0	1706.0	3365.0	1
2	294	36.0	HF	0	2	4.013689	4.0	862.0	1730.0	1
3	323	35.0	NORMANDA	2	1	3.665982	5.0	1059.0	3382.0	1
4	376	34.0	HF	1	1	2.759754	7.0	274.0	650.0	1

Al final del proceso se tiene un dataset con un total de 361 registros y 10 variables.

3.2.3 Identificación y tratamiento de valores nulos

Los valores nulos ayudan a identificar datos faltantes o erróneos, es crucial detectarlos para tomar decisiones adecuadas sobre cómo manejarlos, ya sea imputándolos, eliminándolos o analizando su impacto en los resultados.

Figura 46. Identificación de valores nulos

```
#### Identificación de Valores Nulos
print(final.isnull().sum())
```

```
arete              0
peso               0
raza               0
abortos            0
partos             0
edad               0
enfermedades      38
dias_lactancia    15
produccion        28
descarte           0
dtype: int64
```

Tratamiento de valores nulos

Se tienen valores nulos en las siguientes variables: enfermedades con 38 valores nulos, días_lactancia con 15 valores nulos y producción con 28 valores nulos.

Para enfermedades y días_lactancia, para evitar la pérdida de información y asegurar la coherencia en los datos, se optó por rellenar estos valores nulos utilizando la moda de la columna, es decir, el valor más frecuente, para asegurar una coherencia en la distribución de los datos, ya que número de registro de enfermedades y los días de lactancia son variables que pueden tomar los valores que más se repiten en el hato.

Se imputaron los valores faltantes en la variable produccion utilizando su mediana, dado que esta medida central es menos sensible a valores extremos en comparación con la media, evitando sesgos en la distribución de los datos.

Figura 47. Tratamiento de valores faltantes

```
# Rellenar valores nulos con la media de la columna enfermedades
final['enfermedades'] = final['enfermedades'].fillna(final['enfermedades'].mode()[0])

# Rellenar valores nulos con la mediana de la columna producción
final['produccion'] = final['produccion'].fillna(final['produccion'].median())

# Rellenar valores nulos con la moda de la columna
final['dias_lactancia'] = final['dias_lactancia'].fillna(final['dias_lactancia'].mode()[0])
```

Comprobación de valores nulos

Una vez aplicado los métodos para realizar la imputación de los valores faltantes se observa en la Figura 48, que ya no se presentan valores faltantes en el dataset.

Figura 48. Comprobación valores faltantes

```
###Comprobación Valores Nulos
print(final.isnull().sum())
```

```
arete            0
peso             0
raza             0
abortos          0
partos           0
edad             0
enfermedades     0
dias_lactancia   0
produccion       0
descarte         0
dtype: int64
```

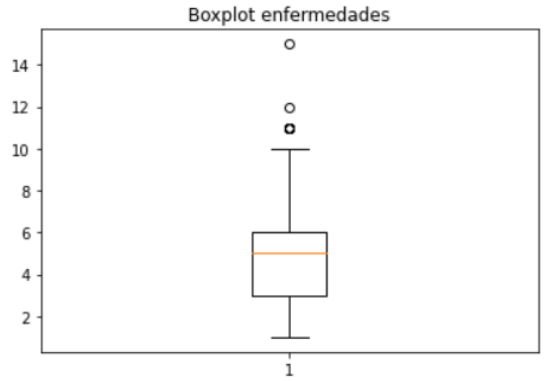
3.2.4 Identificación de Outliers

Los outliers o valores atípicos pueden afectar el rendimiento de los modelos de aprendizaje automático. Como método de identificación de outliers se utiliza Boxplot o diagrama de caja y bigotes, el cual es una técnica gráfica utilizada para detectar valores atípicos en los datos. Este método se basa en el rango intercuartílico (IQR), que se define como la diferencia entre el tercer cuartil (Q3) y el primer cuartil (Q1).

✓ **Enfermedades**

Gráfico 1. Boxplot incidencia de enfermedades

```
plt.boxplot(final['enfermedades'])
plt.title('Boxplot enfermedades')
plt.show()
```

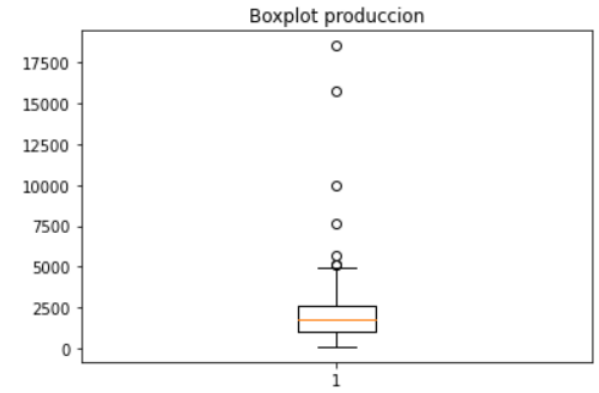


Se observan varios puntos por encima del valor 10, incluyendo valores extremos como 12 y 14, por lo que sugiere que existen algunos casos con un número inusualmente alto de enfermedades.

✓ ***Producción lechera***

Gráfico 2. Boxplot producción lechera

```
plt.boxplot(final['produccion'])
plt.title('Boxplot produccion')
plt.show()
```

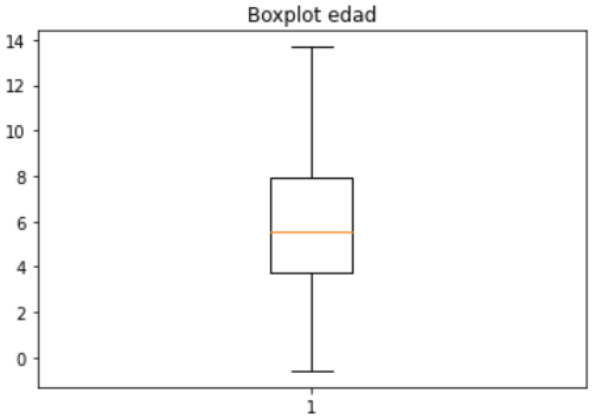


Se observa que existen varios valores atípicos que superan los 5000 litros de leche producidos por un animal, con algunos casos extremos que alcanzan los 17500 litros.

✓ ***Edad***

Gráfico 3. Boxplot edad ganado

```
plt.boxplot(final['edad'])
plt.title('Boxplot edad')
plt.show()
```

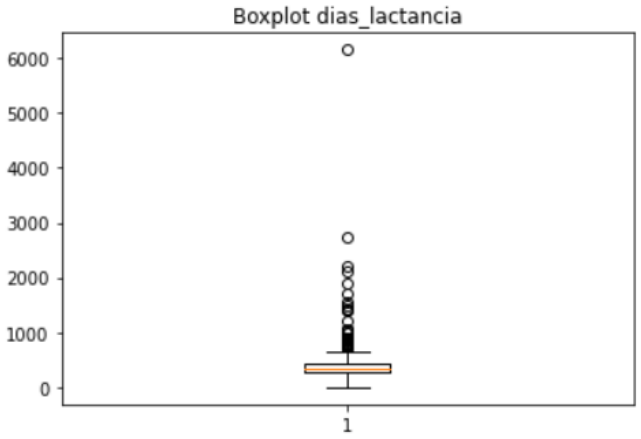


No se observan valores atípicos en la edad del ganado, pero se observa que la mediana esta entre los 4 y 6 años.

✓ ***Días lactancia***

Gráfico 4. Boxplot Días de lactancia

```
plt.boxplot(final['dias_lactancia'])
plt.title('Boxplot dias_lactancia')
plt.show()
```

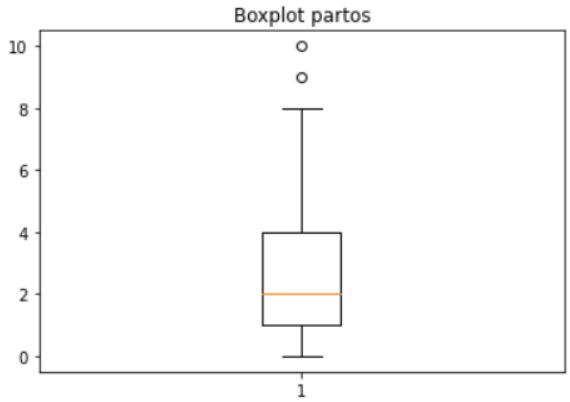


Se observa que existen varios valores atípicos, algunos de los cuales alcanzan hasta 6000 días de lactancia. Estos valores son significativamente diferentes del resto de los datos y pueden influir en el resultado del modelo.

✓ ***Partos***

Gráfico 5. Boxplot número de partos

```
plt.boxplot(final['partos'])
plt.title('Boxplot partos')
plt.show()
```

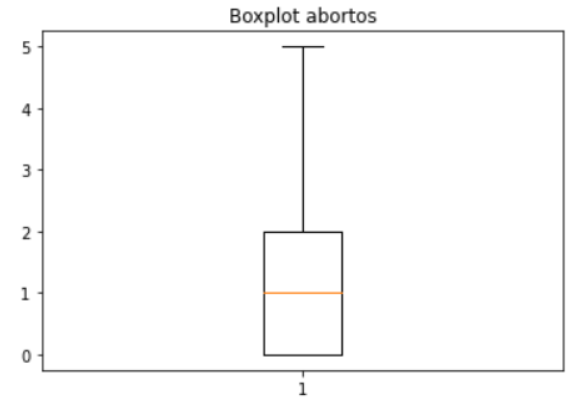


Se observan algunos valores atípicos, lo que sugiere que hay observaciones que son significativamente diferentes del resto de los datos.

✓ ***Abortos***

Gráfico 6. Boxplot número de abortos

```
plt.boxplot(final['abortos'])
plt.title('Boxplot abortos')
plt.show()
```



En el gráfico de abortos, hay varios puntos por encima del valor máximo del bigote superior, lo que sugiere que estos son valores atípicos.

Tratamiento de Outliers

Para mejorar la calidad de los datos se decide filtrar los valores atípicos en las variables enfermedades, días_lactancia, producción y partos, se utilizó el método del rango intercuartil (IQR). Este método establece límites inferior y superior basados en el primer cuartil (Q1) y el tercer cuartil (Q3), eliminando valores que se encuentran fuera del rango, se lo aplica mediante la siguiente instrucción:

Figura 49. Tratamiento de outliers

```
data_filt = final.copy()
#Filtro simple con 1.5^*rango_inter_cuantílico - Asumo distribución normal(comprobar) - eliminar directamente
for columna in ['enfermedades', 'dias_lactancia', 'produccion', 'partos']:
    Q1 = data_filt[columna].quantile(0.25)
    Q3 = data_filt[columna].quantile(0.75)
    rango_inter_quantil = Q3 - Q1
    lim_inf = Q1 - 1.5 * rango_inter_quantil
    lim_sup = Q3 + 1.5 * rango_inter_quantil
    data_filt = data_filt[(data_filt[columna] >= lim_inf) & (data_filt[columna] <= lim_sup)]

print(f"Número registros luego del filtro de valores atípicos: {data_filt.shape[0]}")
print(f"Registros eliminados: {final.shape[0] - data_filt.shape[0]}, es decir, el {(((final.shape[0] - data_filt.shape[0])/final.
```

```
Número registros luego del filtro de valores atípicos: 312
Registros eliminados: 49, es decir, el 13.57% de los datos
```

```
final = data_filt
```

Se eliminaron 312 registros, lo que representa el 13.57% del total de datos. La reducción de datos fue mínima, lo que indica que la presencia de valores atípicos no era significativa.

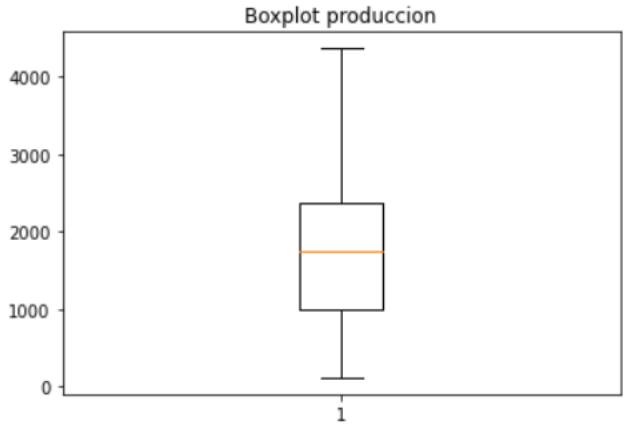
Comprobación de Outliers

Una vez aplicada la técnica de eliminación de outliers se procede a comprobar que no existan outliers en las variables de interés.

✓ ***Producción***

Gráfico 7. Comprobación de outliers producción

```
plt.boxplot(final['produccion'])
plt.title('Boxplot produccion')
plt.show()
```

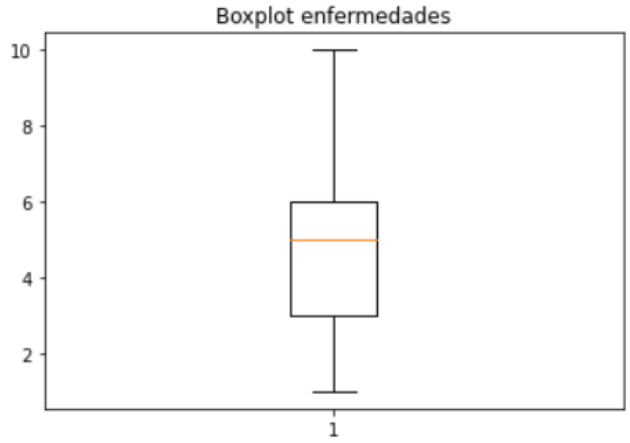


Se observan que los valores atípicos han sido eliminados exitosamente.

✓ ***Enfermedades***

Gráfico 8. Comprobación de outliers enfermedades

```
plt.boxplot(final['enfermedades'])
plt.title('Boxplot enfermedades')
plt.show()
```

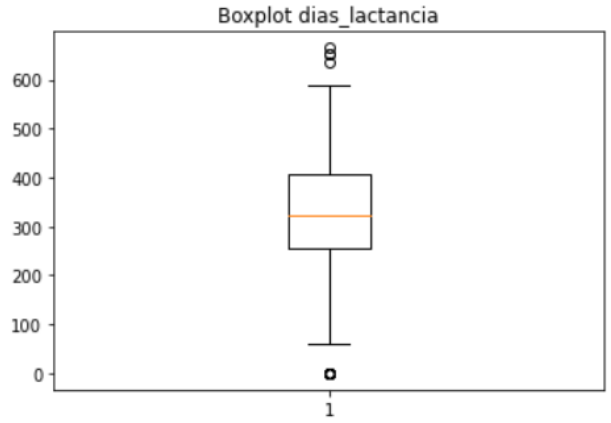


Se observan que los valores atípicos han sido eliminados exitosamente.

✓ ***Días de lactancia***

Gráfico 9. Comprobación de outliers días de lactancia

```
plt.boxplot(final['dias_lactancia'])
plt.title('Boxplot dias_lactancia')
plt.show()
```

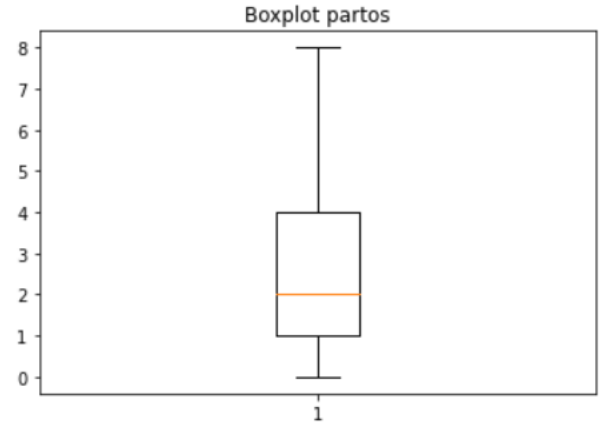


Se observa en el gráfico que se han eliminado una gran parte de valores atípicos de la variable días de lactancia, las variables que quedan no afectan en la calidad de datos ya que se aprecia valores de días de lactancia mayores a 600 lo que si se podría presentar en ganado que tenga dificultad para concebir.

✓ ***Partos***

Gráfico 10. Comprobación outliers partos

```
plt.boxplot(final['partos'])
plt.title('Boxplot partos')
plt.show()
```



Se observan que los valores atípicos han sido eliminados exitosamente.

3.2.5 Estandarización de datos

Tratamiento de variables categóricas

Las variables categóricas contienen valores no numéricos, por lo que se requiere un procesamiento adecuado antes de ser utilizadas en el modelo clasificador. La variable categórica raza se transforma mediante la técnica de codificación *one-hot* (variables *dummies*), con el fin de representar cada categoría como una nueva variable binaria.

Figura 50. Tratamiento de variables categóricas

```
# Crear variables dummies solo para la columna 'raza'
dummies = pd.get_dummies(final['raza'], prefix='raza')

# Unir las variables dummies al DataFrame original
final = pd.concat([final, dummies], axis=1)

# Opcional: eliminar la columna original 'raza'
final.drop('raza', axis=1, inplace=True)

final.head()
```

	peso	abortos	partos	edad	enfermedades	dias_lactancia	produccion	descarte	raza_HF	raza_JERSEY	raza_JERSEY F1	raza_NORMANDA
0	32.0	3	2	2.880219	6.0	284.0	948.0	1	0	0	1	0
4	34.0	1	1	2.759754	7.0	274.0	650.0	1	1	0	0	0
5	36.0	2	1	2.453114	6.0	435.0	706.0	1	1	0	0	0
6	35.0	3	1	4.035592	5.0	461.0	1048.0	1	1	0	0	0
7	36.0	2	1	2.652977	6.0	315.0	2466.0	1	1	0	0	0

3.3 Ingeniería de modelos de aprendizaje automático

Para el desarrollo del modelo clasificador para el descarte vacuno se debe asegurar adecuadas medidas de calidad para cada modelo, la selección del algoritmo de aprendizaje automático y la construcción de los modelos y su entrenamiento.

3.3.1 Definir la medida de calidad del modelo

La evaluación del desempeño del modelo de clasificación es un punto sumamente importante por lo cual para el presente proyecto se seleccionaron métricas en función de la naturaleza del problema y el balance de clases en el conjunto de datos.

Precisión (Accuracy): Métrica que ayuda a medir la proporción de predicciones correctas sobre el total de muestras, esta medida es bastante útil para el problema ya que se tienen clases desbalanceadas, además ayuda a evitar clasificar incorrectamente vacas sanas o productivas como candidatas a descarte.

Sensibilidad o Recall: Métrica que ayuda a Medir la capacidad que tiene el modelo para identificar correctamente los casos positivos. En problemas relacionados con el sector ganadero es importante identificar correctamente la vaca que necesita ser descartada, debido a que no descartar una vaca enferma o poco productiva puede llevar propagación de enfermedades, pérdida de eficiencia productiva y aumento de costos de producción.

F1 (F1-score): Es una medida que obtiene el promedio armónico entre precisión y recall, es útil para evaluar el rendimiento de un modelo de clasificación para el descarte vacuno, especialmente cuando se tienen clases desbalanceadas.

ROC-AUC: Es una medida que ayuda a determinar la capacidad del modelo para distinguir entre clases, considerando la relación entre la tasa de los verdaderos positivos y la tasa de los falsos positivos, es bastante útil también en problemas en donde se tenga clases desbalanceadas.

3.3.2 Selección del algoritmo de ML

Para abordar el problema de clasificación de descarte vacuno, se seleccionaron y evaluaron tres algoritmos de aprendizaje automático los cuales son ampliamente utilizados en problemas de clasificación: **Random Forest, Support Vector Machine (SVM) y Regresión Logística.** La selección de estos algoritmos se basó básicamente en la capacidad de cada uno para manejar problemas de clasificación binaria, por su rendimiento y su aplicabilidad en el contexto de la ganadería.

Random Forest

Es un algoritmo de ensamble que contruye múltiples árboles de decisión y combina sus predicciones para mejorar la precisión y reducir el sobreajuste, cada árbol se entrena con un subconjunto aleatorio de los datos. Se seleccionó este algoritmo debido a su capacidad para manejar datos con alta dimensionalidad, su habilidad para proporcionar información sobre la importancia de las variables en la clasificación y la robustez que tiene ante valores atípicos que a menudo se presentan en análisis de datos ganaderos.

Máquinas de Vectores de Soporte (SVM)

Es un algoritmo que busca encontrar el hiperplano óptimo para separar las clases en base a sus características. Su ventaja radica en su eficiencia clasificando problemas binarios y para manejar datos no lineales mediante el uso de kernels lo que lo convierten en una opción para ser aplicado en problemas de descarte vacuno, en donde las relaciones entre las variables pueden resultar ser complejas.

Regresión Logística

Es un algoritmo de clasificación binaria que estima la probabilidad de pertenencia a una clase mediante una función logística. Se selecciono este algoritmo por su facilidad de implementación, efectividad y eficiencia para comprender la influencia de las variables que afectan la decisión de descarte vacuno.

3.3.3 Construcción y Evaluación de los modelos

Para la construcción de los modelos se tomó en cuenta la cantidad de registros del dataset y como no existieron suficientes datos para dividir el dataset en: datos de entrenamiento, datos de pruebas, y datos de evaluación. Los modelos fueron construidos y evaluados con el 100% de los datos con validación cruzada.

Verificación de balanceo de clases

Antes de construir un modelo clasificador se realiza la verificación del balanceo de clases, esto es importante para saber si los datos del dataset tienen una distribución igual entre las dos clases, se lo hace mediante la siguiente instrucción:

Figura 51. Comprobación desbalance de clases

```
final["descarte"].value_counts()
```

```
1    261
0     51
Name: descarte, dtype: int64
```

Como se puede observar en la Figura 51, el dataset tiene una distribución desigual de las clases dentro de la variable objetivo. En este caso, la clase correspondiente a las vacas que deben ser descartadas es significativamente mayor en cantidad en comparación con la clase de las vacas que no deben ser descartadas. Este fenómeno es conocido como desbalanceo de clases y es común en problemas como el descarte vacuno.

El desbalanceo de clases puede generar problemas en el proceso de entrenamiento de los modelos, ya que los algoritmos de clasificación podrían estar sesgados hacia la clase mayoritaria, lo que puede generar predicciones incorrectas de la clase minoritaria (en este caso, las vacas no destinadas a descarte).

Para solventar este problema, se emplea la técnica de la validación cruzada estratificada, que asegura que cada partición de los datos mantiene una distribución similar de las clases, lo que es particularmente útil cuando hay un desbalanceo de clases.

Mediante esta técnica se consigue que el modelo se entrene y evalúe en condiciones más equilibradas y que la evaluación del rendimiento no se vea sesgada por la clase mayoritaria.

Se aplica la validación cruzada estratificada a 10 folds mediante la siguiente instrucción:

Figura 52. Aplicación validación cruzada

```
# Validación cruzada con 10 folds (todos los datos se usan en entrenamiento y validación)
cv = StratifiedKFold(n_splits=10, shuffle=True, random_state=42)
```

Para la construcción de los modelos de clasificación, se debe separar las variables dependientes de las independientes. En este caso la variable dependiente es **descarte**, que indica si un animal debe ser descartado o no, mientras que las variables independientes son todas las características que pueden influir en esta decisión.

Figura 53. Identificación de variables dependientes e independientes

```
#Identificacion de variable dependientes e independientes
X = final.drop('descarte', axis=1) #Matriz X de entrada -> Variables independientes
y = final['descarte'] #Vector de salida Y -> Variable dependiente
```

Modelo 1 – Random Forest

El modelo fue configurado con una semilla aleatoria para garantizar la reproducibilidad de los resultados, se aplicó validación cruzada con 10 pliegues, lo cual permite evaluar el rendimiento del modelo de manera más robusta y mitigar los efectos del desbalanceo de clases y se definen las métricas para medir el desempeño del modelo.

Figura 54. Aplicación Random Forest

```
# RANDOM FOREST
model1 = RandomForestClassifier(random_state=42)
```

```
# Aplicamos validación cruzada
scores = cross_val_score(model1, X, y, cv=cv, scoring='accuracy')
```

```
# Definir métricas personalizadas
scoring = {
    'accuracy': make_scorer(accuracy_score),
    'f1': make_scorer(f1_score, average='weighted'),
    'recall': make_scorer(recall_score, average='weighted'),
    'roc_auc': make_scorer(roc_auc_score, needs_proba=True)
}
```

Modelo 2 – SVM

Se aplicó validación cruzada con 10 pliegues utilizando la técnica de Stratified K-Fold, para asegurar que cada pliegue mantenga la proporción de las clases original del conjunto de datos y se definen las métricas personalizadas que permiten evaluar el rendimiento del modelo.

Figura 55. Aplicación SVM

```
# MAQUINAS DE SOPORTE VECTORIAL
model2 = SVC(probability=True, kernel='rbf', random_state=42)
```

```
# Definir métricas personalizadas
scoring2 = {
    'accuracy': make_scorer(accuracy_score),
    'f1': make_scorer(f1_score, average='weighted'),
    'recall': make_scorer(recall_score, average='weighted'),
    'roc_auc': make_scorer(roc_auc_score, needs_proba=True)
}
```

Modelo 3 – Regresión Logística

Se aplica validación cruzada con 10 pliegues utilizando la técnica de Stratified K-Fold y se definen las métricas para evaluar el rendimiento del modelo en aspectos clave, considerando especialmente el desbalanceo de clases.

Figura 56. Aplicación Regresión Logística

```
# REGRESION LOGISTICA
model3 = LogisticRegression(max_iter=1000, random_state=42)
```

```
# Definir métricas personalizadas
scoring3 = {
    'accuracy': make_scorer(accuracy_score),
    'f1': make_scorer(f1_score, average='weighted'),
    'recall': make_scorer(recall_score, average='weighted'),
    'roc_auc': make_scorer(roc_auc_score, needs_proba=True)
}
```

3.3.4 Entrenamiento del modelo

Este proceso tiene como objetivo ajustar los parámetros de los algoritmos de clasificación para que pueda predecir la variable dependiente, 'descarte', con la mayor precisión posible.

Modelo 1 – Random Forest

Se aplica la función cross_validate para realizar la validación cruzada, donde además de calcular las métricas definidas, se devolvieron los estimadores entrenados de cada pliegue.

Figura 57. Entrenamiento modelo Random Forest

```
# Aplicamos validación cruzada con múltiples métricas
scores = cross_validate(model1, X, y, cv=cv, scoring=scoring, return_estimator=True)
```

Modelo 2 – SVM

Figura 58. Entrenamiento modelo SVM

```
# Aplicamos validación cruzada
scores2 = cross_validate(model2, X, y, cv=cv, scoring=scoring, return_estimator=True)
```

Modelo 3 – Regresión Logística

Figura 59. Entrenamiento modelo Regresión Logística

```
# Aplicamos validación cruzada
scores3 = cross_validate(model3, X, y, cv=cv, scoring=scoring, return_estimator=True)
```

3.3.5 Selección del modelo

Una vez entrenados los modelos de Random Forest, Máquinas de Soporte Vectorial (SVM) y Regresión Logística, se realiza la selección del mejor modelo basado en un análisis detallado de los resultados obtenidos mediante validación cruzada y las métricas de desempeño definidas previamente, con un enfoque particular en las métricas que son más relevantes debido al desbalanceo de clases presente en el conjunto de datos.

Métricas Random Forest

Figura 60. Métricas modelo Random Forest

```
# Imprimir resultados promedio
print(f"Accuracy promedio: {scores['test_accuracy'].mean():.4f}")
print(f"F1-Score promedio: {scores['test_f1'].mean():.4f}")
print(f"Recall promedio: {scores['test_recall'].mean():.4f}")
print(f"ROC-AUC promedio: {scores['test_roc_auc'].mean():.4f}")
```

```
Accuracy promedio: 0.9647
F1-Score promedio: 0.9641
Recall promedio: 0.9647
ROC-AUC promedio: 0.9831
```

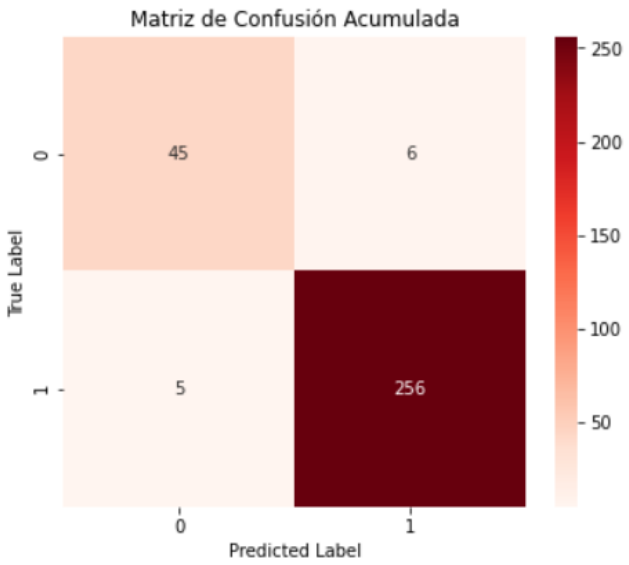
Los resultados obtenidos indican un buen desempeño del modelo. Accuracy y F1-score son notablemente altos, lo que indica una excelente capacidad para clasificar correctamente tanto las clases mayoritarias como las minoritarias, a pesar del desbalance de clases. Recall tiene un promedio de 0.9647 lo que sugiere que el modelo es muy eficaz para identificar correctamente las vacas destinados al descarte, minimizando los falsos negativos y el promedio de ROC-AUC es 0.9831 lo que indica una buena capacidad discriminativa del modelo.

Matriz de Confusión

Se obtiene de la matriz de confusión los siguientes valores:

- ✓ Precisión: El 97.7% de las vacas que el modelo predijo como "Descarte" realmente lo son.
- ✓ Recall: El modelo identificó correctamente el 98.1% de las vacas que debían ser descartadas.
- ✓ Exactitud: El modelo es 96.4% preciso en sus predicciones totales.

Gráfico 11. Matriz de confusión Random Forest



Métricas SVM

Figura 61. Métricas modelo SVM

```
# Imprimir resultados promedio
print(f"Accuracy promedio: {scores2['test_accuracy'].mean():.4f}")
print(f"F1-Score promedio: {scores2['test_f1'].mean():.4f}")
print(f"Recall promedio: {scores2['test_recall'].mean():.4f}")
print(f"ROC-AUC promedio: {scores2['test_roc_auc'].mean():.4f}")
```

```
Accuracy promedio: 0.8785
F1-Score promedio: 0.8705
Recall promedio: 0.8785
ROC-AUC promedio: 0.8845
```

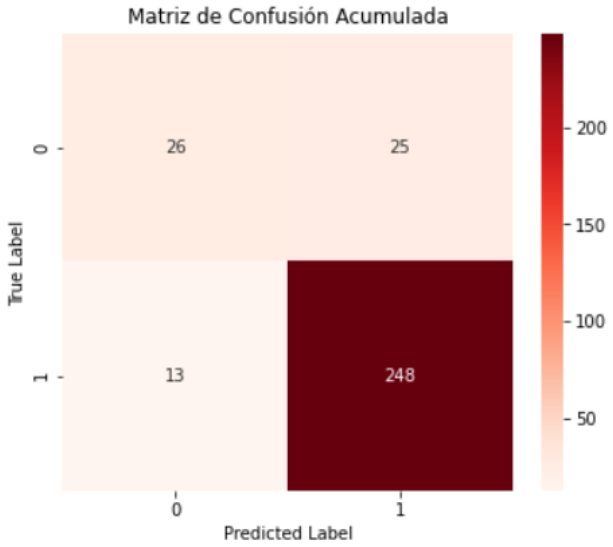
Los resultados indican que el modelo SVM tiene un buen desempeño, sin embargo, es ligeramente inferior al modelo Random Forest en términos de accuracy, F1-score y ROC AUC. Aunque el modelo tiene un buen recall y accuracy, no se desempeña con la misma eficacia en la clasificación de las clases desbalanceadas como el modelo de Random Forest.

Matriz de confusión

Se obtiene de la matriz de confusión los siguientes valores:

- ✓ Precisión: El 90.8% de las vacas que el modelo marcó como "Descarte" realmente lo son.
- ✓ Recall: El 95% de las vacas que debían ser descartadas fueron correctamente identificadas.
- ✓ Exactitud: El modelo es 87.8% preciso en sus predicciones totales.

Gráfico 12. Matriz de confusión modelo SVM



Métricas Regresión Logística

Figura 62. Métricas modelo Regresión Logística

```
# Imprimir resultados promedio
print(f"Accuracy promedio: {scores3['test_accuracy'].mean():.4f}")
print(f"F1-Score promedio: {scores3['test_f1'].mean():.4f}")
print(f"Recall promedio: {scores3['test_recall'].mean():.4f}")
print(f"ROC-AUC promedio: {scores3['test_roc_auc'].mean():.4f}")
```

```
Accuracy promedio: 0.9232
F1-Score promedio: 0.9194
Recall promedio: 0.9232
ROC-AUC promedio: 0.9526
```

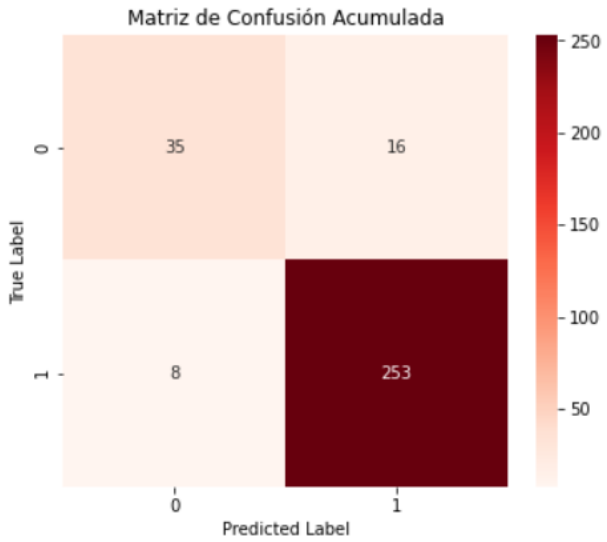
Los resultados obtenidos del modelo de Regresión Logística muestran un desempeño sólido, con un buen accuracy, F1-score, recall y ROC AUC, sin embargo, no alcanza los niveles del modelo Random Forest.

Matriz de confusión

Se obtiene de la matriz de confusión los siguientes valores:

- ✓ Precisión: El 94.0% de las vacas que el modelo marcó como "Descarte" realmente lo son.
- ✓ Recall: El 96.9% de las vacas que debían ser descartadas fueron correctamente identificadas.
- ✓ Exactitud: El modelo es 92.3% preciso en sus predicciones totales.

Figura 63. Matriz de confusión Regresión Logística



Comparación de modelos

Tabla 2. Comparación de modelos de clasificación

Modelo	***Accurancy***	***F1-score***	***Recall***	***ROC AUC***
Random Forest	0.9647	*0.9641*	*0.9647*	*0.9831*
SVM	*0.8785*	*0.8705*	*0.8785*	*0.8845*
Regresión Logística	*0.9232*	*0.9194*	*0.9232*	*0.9526*

Como se observa, en la Tabla 2, el modelo Random Forest tiene mejores resultados en cuanto a accuracy, F1-score, recall y ROC AUC. Sin embargo, el modelo 3 de Regresión Logística ofrece un desempeño muy cercano, con un accuracy de 0.9232 y un recall de 0.9232, lo que lo hace una opción sólida. El modelo SVM, por su parte, muestra un rendimiento inferior en todas las métricas.

Por lo tanto, se selecciona el modelo Random Forest como el modelo final para la predicción del descarte vacuno y se justifica por los siguientes motivos:

- Alto rendimiento en todas las métricas c (accuracy, F1-score, recall, y ROC AUC).
- capacidad para manejar el desbalanceo de clases, asegurando una clasificación precisa de las clases minoritarias.
- Robustez y capacidad para capturar relaciones complejas y no lineales entre las variables.

3.4 Despliegue

Una vez que el modelo ha sido evaluado utilizando técnicas de validación cruzada y evaluado mediante métricas claves como accuracy, F1-score, recall y ROC-AUC, se selecciona Random Forest como el modelo de clasificación para la predicción del descarte vacuno ya que ha demostrado tener el mejor rendimiento y una alta capacidad para manejar el desbalanceo de clases.

Para asegurar la disponibilidad del modelo para su uso futuro, se procede a guardar el modelo entrenado utilizando la librería joblib de Python.

El modelo se guardó en un archivo con la extensión. pkl para su persistencia. La instrucción utilizada para guardar el modelo es la siguiente:

Figura 64. Almacenamiento del modelo

```
# Guardar el modelo
from sklearn.ensemble import RandomForestClassifier
import joblib

# Entrenamos el modelo Random Forest
model1 = RandomForestClassifier(n_estimators=100, random_state=42)
model1.fit(X, y)  # Entrenar el modelo

# Guardamos el modelo en un archivo
joblib.dump(model1, "modelo_random_forest.pkl")
print("Modelo Random Forest guardado con éxito.")
```

```
Modelo Random Forest guardado con éxito.
```

Para evaluar el modelo en un conjunto de datos nuevo:

- Se creó un nuevo **Notebook** en Jupyter para realizar la evaluación.
- Se importó un **nuevo conjunto de datos** con información realista para comprobar el rendimiento del modelo en condiciones no vistas durante el entrenamiento.

Figura 65. Carga del nuevo dataset para evaluación del modelo

```
import pandas as pd

# Especifica la ruta del archivo Excel
ruta_archivo = 'C:\GANADERIA\Datos de Prueba.xlsx'

# Carga el archivo Excel en un DataFrame
df = pd.read_excel(ruta_archivo)

# Imprime las primeras filas del DataFrame para verificar
print(df.head())
```

```
   peso  abortos  partos       edad  enfermedades  dias_lactancia  produccion  \
0    34        0       5  10.257534             5             669        3507
1    35        0       6   9.890411             8             669        2894
2    35        1       6   9.471233             9             343        1985
3    35        0       6   9.112329             4             394        6587
4    35        0       6   9.065753            12             125        2524

  raza
0   HF
1   HF
2   HF
3   HF
4   HF
```

Una vez cargado el nuevo dataset se procede a transformar la variable categórica raza, de modo que el nuevo dataset tenga los mismos tipos de datos y variables con que fue entrenado el modelo.

Figura 66. Transformación de variables categóricas nuevo dataset para evaluación

In [2]:
```
# Crear variables dummies solo para la columna 'Genero'
dummies = pd.get_dummies(df['raza'], prefix='raza')

# Unir las variables dummies al DataFrame original
nuevosDatos = pd.concat([df, dummies], axis=1)

# Opcional: eliminar la columna original 'Genero'
nuevosDatos.drop('raza', axis=1, inplace=True)

nuevosDatos.head()
```

Out[2]:

	peso	abortos	partos	edad	enfermedades	dias_lactancia	produccion	raza_HF	raza_JERSEY	raza_JERSEY F1	raza_NORMANDA
0	34	0	5	10.257534	5	669	3507	1	0	0	0
1	35	0	6	9.890411	8	669	2894	1	0	0	0
2	35	1	6	9.471233	9	343	1985	1	0	0	0
3	35	0	6	9.112329	4	394	6587	1	0	0	0
4	35	0	6	9.065753	12	125	2524	1	0	0	0

Como siguiente paso se realiza la carga del modelo previamente guardado utilizando la biblioteca joblib.

Figura 67. Carga del modelo

```
# Cargar modelo
import pandas as pd
import joblib
modelo_rf_cargado = joblib.load("modelo_random_forest.pkl")
```

Para evaluar el rendimiento del modelo en datos nuevos, se aplica la función de predicción sobre el nuevo conjunto de datos de evaluación.

Figura 68. Aplicación función de predicción para descarte vacuno

```
# Hacer predicción
prediccion = modelo_rf_cargado.predict(nuevosDatos)
print("Predicción:", prediccion)
```

```
Predicción: [0 1 1 0 1 1 1 0 1 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 1 1 1
 1 1 0 0 0 0 0 0 0 1 1 1 1 0]
```

Para una mejor visualización de datos se procede a transformar la variable descarte la cual toma valores de 0 y 1 a SI y NO para indicar si el animal debe ser descartado o no.

Figura 69. Nuevas predicciones

```
nuevosDatos['descarte']= prediccion
```

```
nuevosDatos = nuevosDatos.replace({0: 'NO', 1: 'SI'})
```

```
nuevosDatos.head(10)
```

	peso	abortos	partos	edad	enfermedades	dias_lactancia	produccion	raza_HF	raza_JERSEY	raza_JERSEY F1	raza_NORMANDA	descarte
0	34	NO	5	10.257534	5	669	3507	SI	NO	NO	NO	NO
1	35	NO	6	9.890411	8	669	2894	SI	NO	NO	NO	SI
2	35	SI	6	9.471233	9	343	1985	SI	NO	NO	NO	SI
3	35	NO	6	9.112329	4	394	6587	SI	NO	NO	NO	NO
4	35	NO	6	9.065753	12	125	2524	SI	NO	NO	NO	SI
5	35	NO	5	9.008219	6	194	6321	SI	NO	NO	NO	SI
6	35	NO	5	9.038356	7	195	4125	SI	NO	NO	NO	SI
7	34	NO	4	8.698630	5	734	4789	SI	NO	NO	NO	NO
8	35	NO	4	8.687671	8	411	1295	SI	NO	NO	NO	SI
9	34	NO	5	8.432877	5	468	3225	SI	NO	NO	NO	NO

3.5 Monitoreo y mantenimiento

Una vez desplegado el modelo clasificador de descarte vacuno, es fundamental implementar estrategias para monitorear y brindar mantenimiento con el fin de garantizar su óptimo rendimiento con datos nuevos.

3.5.1 Monitoreo del Desempeño del Modelo

Realizar el monitoreo continuo del modelo permite asegurar su correcto funcionamiento detectando posibles degradaciones en su funcionamiento debido a cambios que podrían presentarse en los datos o en las condiciones del entorno, por lo que se propone:

- Métricas de rendimiento: Se deberá calcular las siguientes métricas precisión, recall, F1-score, ROC-AUC y matriz de confusión de forma periódica (mensual o semanalmente) para verificar que el modelo no ha perdido su precisión, estas métricas deberán ser comparadas con los valores obtenidos en la fase de evaluación.
- Frecuencia de reentrenamiento: Es recomendable reentrenar el modelo trimestralmente, sin embargo, se debe tomar en cuenta el desempeño y si se tiene nuevos datos.

  En el caso que las métricas de evaluación presenten una degradación de mas de 5% se recomienda realizar un reentrenamiento de forma inmediata.

  Si se tiene una cantidad significativa de nuevos datos, se recomienda también realizar un reentrenamiento de forma inmediata.

3.5.2 Mantenimiento del Modelo

Darle al mantenimiento al modelo es de suma importancia ya que permite asegurar que el modelo esté funcionando de manera adecuada y para llevar a cabo este proceso de deben de tomar en cuenta las siguientes consideraciones:

- Se deberá incorporar nuevos datos al conjunto de datos de entrenamiento de forma continua con el objetivo de realizar la respectiva actualización de registros existentes y corregir posibles errores.
- Como parte del mantenimiento del modelo se debe evaluar el rendimiento, si se observa que las métricas disminuyen significativamente, se debería explorar la posibilidad de ajustar los hiperparámetros del algoritmo.
- Se debe mantener actualizado el software y las bibliotecas que fueron utilizadas para la implementación del modelo con el fin de garantizar su buen funcionamiento y compatibilidad.

4. Resultados

Desempeño del modelo seleccionado

El modelo Ramdon Forest fue seleccionado como el modelo clasificador de descarte vacuno, fue evaluado con un conjunto de datos independiente para medir su capacidad de generalización. Se calcularon métricas claves para medir su calidad y desempeño, obteniendo los siguientes resultados:

Figura 70. Resultados Métricas Modelo Clasificador Vacuno

Métrica	Valor
Exactitud (Accuracy)	0.9647
F1-Score	0.9641
Recall (Sensibilidad)	0.9647
Área Bajo la Curva ROC (AUC-ROC)	0.9831

- **Exactitud (Accuracy):** Según los resultados obtenido se observa que el modelo clasificó correctamente el 96.47% de los casos totales.
- **F1-Score:** Esta métrica indica que el equilibrio entre precisión y recall es alto (0.9641), lo que indica que el modelo tiene un buen balance entre evitar falsos positivos y falsos negativos.
- **Recall (Sensibilidad):** El modelo tiene un recall de 0.9647, lo que significa que el modelo identifica correctamente el 96.47% de los animales que deben ser descartados, minimizando falsos negativos.
- **AUC-ROC:** El modelo obtiene un valor de 0.9831, lo que muestra una excelente capacidad para distinguir entre animales de descarte y no descarte en distintos umbrales de decisión.

Matriz de Confusión

En la matriz de confusión se obtuvo los siguientes resultados:

- Verdaderos Negativos (TN = 45): Casos correctamente clasificados como No Descarte.

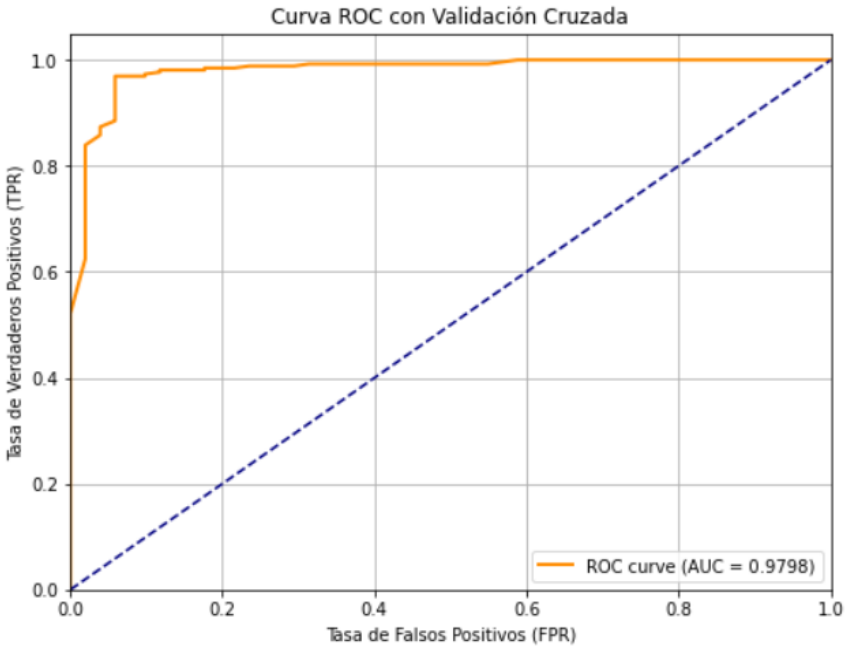
- Falsos Positivos (FP = 6): Casos que el modelo predijo como Descarte, pero en realidad no lo eran.
- Falsos Negativos (FN = 5): Casos que el modelo predijo como No Descarte, pero sí eran descarte.
- Verdaderos Positivos (TP = 256): Casos correctamente clasificados como Descarte.

Lo que indica que el modelo tiene un bajo número de errores (FN y FP), una alta calidad de TP y TN lo que indica que el modelo tiene un buen desempeño.

Curva de ROC

La Curva ROC ayuda a evaluar el desempeño del modelo en términos de la tasa de verdaderos positivos (TPR o sensibilidad) frente a la tasa de falsos positivos (FPR).

Figura 71. Curva ROC con Validación Cruzada



Como se puede observar en el gráfico, La curva está muy cerca del vértice superior izquierdo (0,1), lo que indica que el modelo logra detectar la mayoría de los casos positivos con una baja tasa de falsos positivos, lo que sugiere un buen desempeño del modelo.

El área bajo la curva AUC es cerca de 1, lo que significa que el modelo tiene una excelente capacidad de diferenciar entre las clases.

La línea diagonal azul representa un clasificador aleatorio es decir sin información, como la curva naranja que representa al modelo está muy por encima de esta línea se confirma que el modelo es mucho mejor que un clasificador aleatorio.

Consideraciones sobre el Balanceo del Dataset

El dataset utilizado presentaba un desbalanceo entre las clases, con 261 registros correspondientes a la clase “1” (Descarte Vacuno) y 51 registros de la clase "0" (No Descarte Vacuno). Cuando se presenta este tipo de distribución se podría afectar el desempeño del modelo, ya que los algoritmos de clasificación por lo general suelen inclinarse hacia la clase mayoritaria, generando una menor capacidad de detección en la clase minoritaria, sin embargo, se decide aplicar la técnica de validación cruzada Stratified K-Fold Cross-Validation, dando como resultado en la evaluación unas métricas con valores bastante satisfactorios, con un accuracy de 96.47%, un F1-score de 96.41%, , un AUC-ROC de 98.31% y una matriz de confusión con un bajo número de errores, lo que indica que el modelo es capaz de distinguir con una alta precisión entre ambas clases a pesar de tener una clase minoritaria.

Dado este alto rendimiento, no fue necesario aplicar técnicas de balanceo del dataset como oversampling (ejemplo: SMOTE - Synthetic Minority Over-sampling Technique) o undersampling, ya que el modelo logró una clasificación efectiva sin necesidad de modificar la distribución de las clases.

Los resultados obtenidos sugieren que el modelo posee una capacidad robusta de generalización, logrando un equilibrio entre precisión y sensibilidad sin sesgarse excesivamente hacia la clase mayoritaria.

5. Conclusiones y Recomendaciones

5.1 Conclusiones

- Mediante el desarrollo del modelo clasificador de descarte vacuno, se puede generar una ventaja competitiva dentro de cualquier hacienda ya que el modelo representa un aporte significativo para el sector ganadero, particularmente en la toma de decisiones relacionadas con el descarte de vacas lecheras.
- A través de un enfoque basado en aprendizaje automático, se puede optimizar la eficiencia productiva dentro de una hacienda ganadera ya que se logra evitar la permanencia de animales con baja rendimiento en el hato lechero.
- El modelo permite indicar con precisión aquellas vacas no contribuyen de forma óptima a la producción, reduciendo de esta forma costos operativos ya que se evita el mantenimiento de animales improductivos y mejorando la rentabilidad del sistema ganadero.
- La inclusión de un modelo de clasificación en la ganadería lechera asegura mejorar la gestión productiva facilitando la toma de decisiones, las cuales están basadas en criterios previamente estudiados y probados por el modelo desarrollado, haciendo que la gestión de la hacienda sea más precisa y sustentable del hato lechero.
- El modelo de clasificación les permite a los ganaderos tomar decisiones basadas en datos objetivos sobre el estado de salud y la productividad de los animales, en lugar de depender únicamente de criterios tradicionales basados en la experiencia del ganadero que podrían estar influenciados por factores como la intuición o el sesgo humano. Con el modelo se busca asegurar que los animales que necesitan ser descartados por razones de salud o ineficiencia productiva sean identificados con mayor precisión.
- La selección de variables que se utilizarón en el modelo clasificador fue un factor determinante para el desempeño y precisión. Entre los hiperparámetros considerados se incluyeron características claves para evaluar el desempeño del ganado lechero, como: edad, número de partos, número de abortos, días de lactancia, producción lechera, presencia de enfermedades y raza. La inclusión de estas variables permitió construir un modelo que refleja con mayor precisión la realidad del sector ganadero, facilitando decisiones más objetivas y fundamentadas en la gestión del descarte vacuno.

- Mediante el modelo se busca también prevenir el sufrimiento innecesario de los animales ya que se puede evitar la fatiga, la desnutrición y otras condiciones que pueden ocurrir si los animales continúan en el sistema productivo cuando ya no son aptos para ello, mejorando así su calidad de vida.
- El dataset con el que se trabajó el desarrollo del modelo presentaba un problema en la distribución de clases ya que se tenían más registros en una clase que en la otra por lo que se decidió aplicar la técnica de la validación cruzada para mejorar la evaluación y el rendimiento del modelo clasificador. Al dividir el dataset en múltiples pliegos y realizar la evaluación sobre distintos subconjuntos, se ayudó a mitigar el sesgo hacia la clase mayoritaria, por lo que el resultado fue un mejor ajuste del modelo a las características de ambas clases, mejorando la capacidad predictiva del modelo clasificador.
- Para el desarrollo del modelo clasificador de descarte vacuno, se probaron tres enfoques diferentes: Random Forest, Support Vector Machine (SVM) y Regresión Logística, en base a las métricas de calidad aplicadas a cada uno de estos modelos, se optó por seleccionar Random Forest como el modelo final para la predicción del descarte vacuno, ya que presentó un accuracy promedio de 96.47%, un F1-score de 96.41%, y un ROC-AUC de 98.31%, con lo que demostró ser un modelo capaz de realizar una clasificación precisa, equilibrada y robusta, incluso en presencia de clases desbalanceadas.

5.2 Recomendaciones

- Se debería recolectar más datos en aquellos casos donde las clases están desbalanceadas. Si bien el modelo ha mostrado buenos resultados al abordar el desbalanceo, contar con un conjunto de datos más equilibrado ayudaría a mejorar la capacidad predictiva, especialmente para la clasificación de las clases menos representadas.
- Realizar una validación regular del modelo de clasificación utilizando datos actualizados, para seguir garantizar la precisión y calidad de las predicciones a medida que cambian las condiciones y los factores que influyen en el descarte vacuno.
- El estudio se podría extender a más enfoques en el área ganadera como por ejemplo: incluir la predicción de enfermedades, la optimización de la dieta o la predicción de la productividad de los animales. Machine Learning puede tener un papel bastante relevante en la industria ganadera.

- El modelo se podría aplicar a otras especies ganaderas, como ovinos o porcinos, donde el proceso de descarte también es crucial para mejorar la eficiencia productiva. Las metodologías utilizadas en este trabajo pueden ser extendidas a otras especies para mejorar la gestión y el bienestar animal en la ganadería.
- Una vez que el modelo esté implementado, sería importante establecer un sistema de monitoreo continuo para evaluar su desempeño y realizar ajustes periódicos. Es posible que los patrones de descarte cambien con el tiempo debido a cambios en las prácticas ganaderas.

6. Bibliografía.

Agrobanco. (2012.). Ganado bovino: Manejo y producción. Agrobanco. Obtenido de https://www.agrobanco.com.pe/data/uploads/ctecnica/018-e-ganado.pdf

Alpaydin, E. (2020). Introduction to Machine Learning (4th ed.). MIT Press.

Amazon Web Services, Inc. (n.d.). ¿Qué es la regresión logística?. Recuperado de https://aws.amazon.com/es/what-is/logistic-regression/

Battistini, V.E. 1958. Estudio sobre algunos aspectos relativos a la reproducción del ganado Holstein en el establo de Maranga. Tesis Bachillerato. Facultad de Zootecnia, Universidad Nacional Agraria La Molina. Lima. 96 p

Bovina, G. (2020, 6 julio). Cuál es la vida productiva de una vaca lechera - Revista Genética Bovina Colombiana. Revista Genética Bovina Colombiana. https://revistageneticabovina.com/reproduccion/vaca-lechera/

Cilecuador. (2019, 13 agosto). La Sierra ecuatoriana: cuna de la leche. Más Leche Ecuador. https://www.masleche.ec/post/la-sierra-ecuatoriana-cuna-de-la-leche

Communications. (2024, 16 julio). «Machine learning»: ¿qué es y cómo funciona el maestro en reconocer patrones? BBVA NOTICIAS. Obtenido de https://www.bbva.com/es/innovacion/machine-learning-que-es-y-como-funciona/

Cortes, C., & Vapnik, V. (1995). Support-vector networks. Machine Learning, 20(3), 273-297. https://doi.org/10.1007/BF00994018

DeVries T.J, M.A.G. von Keyserlingk, D.M. Weary. 2004. Effect of feeding space on the inter–cow distance, aggression, and feeding behavior of free-stall housed Holstein dairy cows. J. Dairy Sci., 87 (2004), pp. 1432–1438. Fetrow, J; Nordlund, K.V; Norman, H.D. Invited Review: Culling: Nomenclature, Definitions, and Recommendations, Journal of Dairy Science, Volume 89, Issue 6, 2006.

DeVries T.J, M.A.G. von Keyserlingk, D.M. Weary. 2004. Effect of feeding space on the inter–cow distance, aggression, and feeding behavior of free-stall housed Holstein dairy cows. J. Dairy Sci., 87 (2004), pp. 1432–1438. Fetrow, J; Nordlund, K.V;

Norman, H.D. Invited Review: Culling: Nomenclature, Definitions, and Recommendations, Journal of Dairy Science, Volume 89, Issue 6, 2006.

Ecuador, A. H. (2018). Asociación Holstein Friesian del Ecuador. [Holstein Friesian Association of Ecuador]. Obtenido de http://www.holsteinecuador.com/

Fawcett, T. (2006). An introduction to ROC analysis. Pattern Recognition Letters, 27(8), 861-874. https://doi.org/10.1016/j.patrec.2005.10.010

Fussen. (2022, 25 marzo). Ciclo de lactancia. Fussen. https://fussen.us/ciclo-de-la-lactancia-de-la-vaca/

Hidalgo, E. P. F. B. y. F. (2007). Libre comercio y lácteos: la producción de leche en el Ecuador entre el mercado nacional y la globalización. FLACSO Andes. https://biblio.flacsoandes.edu.ec/libros/110959-opac

Inesdi. (n.d.). Random Forest: ¿Qué es?. Recuperado de https://www.inesdi.com/blog/random-forest-que-es/

Infocarne, (2010). [Infocarne, (2010)]. Obtenido de http://www.infocarne.com

Mahesh, B. (2018). Machine Learning Algorithms-A Review. International Journal of Science and Research. https://doi.org/10.21275/ART20203995

Martinez, K. G. (2018). Descarte de vacas. Zootecnia y Veterinaria Es Mi Pasión. Obtenido de https://zoovetesmipasion.com/ganaderia/descarte-de-vacas#Descarte_de_vacas

Ml-ops.org. (2024, 5 julio). Obtenido de https://ml-ops.org/content/crisp-ml#machine-learning-model- engineering

Olamendy, J. C. (n.d.). A Comprehensive Guide to Stratified K-Fold Cross-Validation for Unbalanced Data. Recuperado de https://medium.com/@juanc.olamendy/a-comprehensive-guide-to-stratified-k-fold-cross-validation-for-unbalanced-data-014691060f17

Orrego, J., Delgado, A., & Echevarría, L. (2003). Vida productiva y principales causas de desecho de vacas Holstein en la cuenca de Lima. Revista de Investigaciones Veterinarias del Perú, 14(1), 68-73.

Palacios-Espinosa, Alejandro, Domínguez-Viveros, Joel, Padrón-Quintero, Yamariz, Rodríguez Castro, Manuel, Rodríguez-Almeida, Felipe Alonso, Espinoza-Villavicencio, José Luis, & Ávila-Serrano, Narciso Ysac. (2016). Caracterización de la curva de lactancia de bovinos Siboney con modelos no lineales mixtos. Revista mexicana de ciencias pecuarias, 7(2), 233-242. Recuperado en 24 de diciembre de 2024, de http://www.scielo.org.mx/scielo.php?script=sci_arttext&pid=S2007-11242016000200233&lng=es&tlng=es.

Phileo by Lesaffre. (2020, 26 noviembre). Fertilidad ganado lechero • Phileo by Lesaffre. Phileo By Lesaffre. https://phileo-lesaffre.com/es/ganado-lechero/fertilidad-ganado-lechero/

Powers, D. M. (2011). Evaluation: From precision, recall and F-measure to ROC, informedness, markedness & correlation. Journal of Machine Learning Technologies, 2(1), 37-63.

Producción lechera. (s. f.). DairyProductionProducts. https://www.fao.org/dairy-production-products/production/es/

Sector lácteo en Ecuador: una mirada a los desafíos y oportunidades. (2024, 11 junio). https://revistagestion.ec/analisis-economia-y-finanzas/sector-lacteo-en-ecuador-una-mirada-los-desafios-y-oportunidades/

Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. Information Processing & Management, 45(4), 427-437. https://doi.org/10.1016/j.ipm.2009.03.002

Tarifa, L. (2024, 3 mayo). 4 Razones para aplicar la Inteligencia artificial en la Ganadería. ISAM Education. Obtenido de https://isam.education/4-razones-para-aplicar-la-inteligencia-artificial-en-la-ganaderia/

Veterinaria Digital S.A. (2022, 13 junio). La producción de leche en Ecuador. Veterinaria Digital - Avicultura, Porcicultura, Rumiantes y Acuicultura. Obtenido de https://www.veterinariadigital.com/articulos/la-produccion-de-leche-en-ecuador/#:~:text=mantequilla%20etc.).-,En%20Ecuador%20se%20producen%20aproximadamente%206%2C15%20millones%20de%20litros,1%2C2%20millones%20de%20personas.

Zhang, H. (2004). The optimality of naive Bayes. AAAI 2004, Proceedings of the 16th Conference on Artificial Intelligence, 2, 122-127.