Detección de fraude alimentario en leche

Análisis de especiación de leche y detección de leche de cabra adulterada con leches de menor calidad, empleando aprendizaje automático e implementación en aplicación web.

Universitat Oberta de Catalunya

UNIVERSITAT DE BARCELONA

Miguel Ángel López González

Máster universitario en bioinformática y bioestadística.

Bioinformática estadística y aprendizaje automático

Nombre Tutor/a de TF

Romina Astrid Rebrij

Profesor/a responsable de la asignatura

Carles Ventura Royo

20 de junio de 2023

FICHA DEL TRABAJO FINAL

Título del trabajo:	Detección de fraude alimentario en leche**:** Análisis de especiación de leche y detección de leche de cabra adulterada con leches de menor calidad, empleando aprendizaje automático e implementación en aplicación web.
Nombre del autor:	Miguel Ángel López González
Nombre del consultor/a:	Romina Astrid Rebrij
Nombre del PRA:	Carles Ventura Royo
Fecha de entrega (mm/aaaa):	06/2023
Titulación o programa:	Máster universitario en bioinformática y bioestadística
Área del Trabajo Final:	Bioinformática estadística y aprendizaje automático
Idioma del trabajo:	Castellano
Palabras clave	*Machine learning*, fraude alimentario, adulteración

Resumen del trabajo

El fraude alimentario es un riesgo que compromete la calidad y seguridad alimentaria e implica un agravio económico. El presente trabajo realiza un estudio de especiación y de adulteración de leches a partir de datos de espectrometría de masas. Tiene la finalidad de encontrar los modelos más eficientes e implementarlos en una herramienta que haga accesible la detección de fraude en leche. Se han empleado los algoritmos: support vector

machines (SVM), artificial neural networks (NN) y random forests (RF), diseñando varios modelos con diferentes parámetros. La alta dimensionalidad de los datos y escasez de muestras ha hecho necesario un aumento de muestras mediante la técnica SMOTE y reducir dimensiones mediante principal component analysis (PCA). La aplicación de estas técnicas juntas y por separado ha generado cuatro escenarios de trabajo. La determinación del mejor modelo se basó en las métricas de exactitud, Kappa, sensibilidad y especificidad, además de priorizar el modelo más simple. Se escogieron como óptimos los modelos SVM con kernel lineal, consiguiendo un 100% de exactitud en especiación y un 96.3% en adulteración. Se ha demostrado la capacidad de los modelos seleccionados para detectar de fraude en leche con una exactitud mínima de un 90%. También se demuestra que reducir dimensiones y aumentar datos es el mejor escenario, mejorando la eficiencia. Por ende, se ha podido apreciar que PCA y SMOTE son buenas técnicas para dichas tareas. Finalmente se implementaron los modelos seleccionados en una herramienta web, facilitando la comprobación de fraude de una forma sencilla y rápida.

Abstract

Food fraud is a risk that compromises the quality and safety of food and involves an economic disadvantage. This work carries out a study of speciation and adulteration of milk from mass spectrometry data. It aims to find the most efficient models and implement them into a tool that facilitate the detection of fraud in milk. The following algorithms have been used: support vector machines (SVM), artificial neural networks (NN) and random forests (RF), from which several models with different parameters where designed. High dimensionality of data and scarcity of samples involved sample increasing using the SMOTE technique and also involved dimension reduction by principal component analysis (PCA). The application of these techniques together and separately has generated four working scenarios. The determination of the best model was based on accuracy, Kappa, sensitivity, and specificity metrics, as well as the prioritization of simplest model. SVM models with linear kernel were chosen as optimal, achieving 100% accuracy in speciation and 96.3% in adulteration. The ability of selected models to detect fraud in milk with a minimum accuracy of 90% has been demonstrated. It is also demonstrated that reducing dimensions and increasing data is the best scenario, improving efficiency. Therefore, it was found that PCA and SMOTE are good techniques for this aim. Finally, the selected models were implemented in a web-based tool, making accessible milk fraud checking in a simple and fast way.

Índice

Lista de figuras

Lista de tablas

1 Introducción

1.1 Contexto y justificación del trabajo

El fraude alimentario es un tipo de fraude que tiene lugar cuando se engaña al consumidor en relación con la calidad o contenido del alimento en cuestión [1]. Habitualmente, el fraude suele tener lugar cuando se quiere sustituir el alimento entero o algún elemento por algún otro de calidad inferior o bien añadiendo algún ingrediente para otorgar una apariencia de mayor volumen o peso [2].

La causa principal del fraude alimentario recae en la obtención de un mayor beneficio o ventaja, habitualmente económica para el operador alimentario que las ofrece [1]. Aunque, al fin y al cabo, el aspecto económico sea el principal motor del fraude alimentario, hay diversos motivos que estimulan la comisión de este fraude tal como:

- Oferta menor a la demanda.
- Reducción de coste para mejorar la competitividad.
- Aumentar los márgenes de beneficio.
- La incapacidad del consumidor para permitirse un producto con sus ingredientes originales.
- Falta de formación de la mano de obra ya sea en la relación a las técnicas productivas como del conocimiento en seguridad alimentaria. Para estos casos particulares la adulteración puede ser inintencionada.

[2]

Aunque se ha mencionado la adición y sustitución de componentes como ejemplo de adulteración alimentaria, este campo es mucho más amplio. La figura 1 extraída de Momtaz et al. [3], muestra un esquema de los diferentes tipos de adulteración de los alimentos.

El presente trabajo se centrará en la substitución alimentaria de especies y la dilución de alimentos.

La adulteración de alimentos es un enemigo silencioso, al ser una práctica relativamente extendida y difícil de detectar. Aun así, se detectan mensualmente una gran cantidad de casos [4], especialmente derivados de países con una legislación laxa en materia de seguridad alimentaria. A continuación, se citan algunos ejemplos:

- Pakistán. Diciembre de 2022: Las autoridades decomisan un total de 9.800 litros de leche adulterada con ghee, polvo, detergente, agua contaminada y otros químicos peligrosos.

- Estado Unidos. Diciembre de 2022: La FDA toma muestra y testea 144 unidades de miel por edulcorantes no declarados añadidos al producto, evidenciando una tasa de adulteración del 10%.

- Unión europea. Noviembre de 2022: Las autoridades en el marco de la operación OPSON XI implicado un total de 26 países, decomisan un total de 27.000 toneladas de comida fraudulenta y un total de 15 millones de litros de bebidas alcohólicas entre diciembre y de 2021 y mayo de 2022.

- Turquía. Noviembre de 2022: Cuatro personas fallecen al consumir alcohol adulterado con metanol.

- España. Septiembre de 2022: AESAN notifica a los consumidores sobre marcas de aceite de oliva adulteradas con otros aceites vegetales con falta de documentación de trazabilidad fiable.

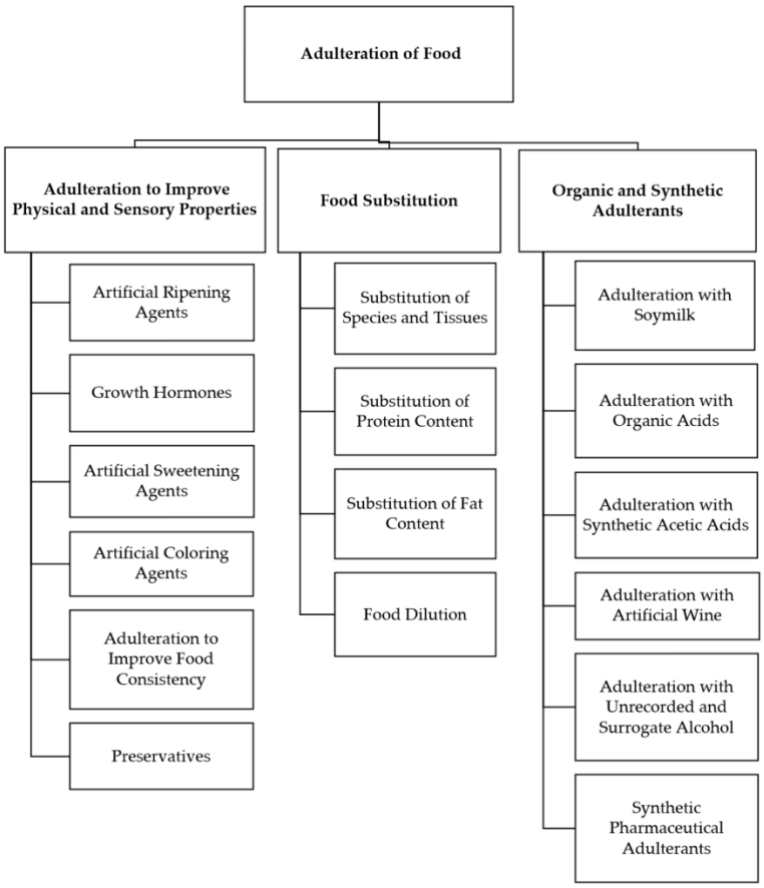

Figura 1. Tipos más comunes de adulteración alimentaria [3].

Si bien, en ocasiones el fraude y las adulteraciones pueden derivar en modificaciones de los parámetros nutricionales del alimento original sin ir más allá, en muchas otras ocasiones pueden mermar la salud de los consumidores. Por ejemplo, la adición de ingredientes no declarados en los alimentos puede derivar en alergias e intolerancias que pueden llegar a causar anafilaxis, mareos, náuseas, pérdida de visión e incluso la muerte. En otras ocasiones, los resultados pueden verse más a largo plazo, causando insuficiencia renal, cáncer, daños neurológicos, daños renales entre muchas otras [3].

De aquí nace la necesidad de desarrollar nuevos métodos de detección para prevenir la salida al mercado de estos productos o poder detectar antes de que causen algún daño, aquellos que sean sospechosos de ser fraudulentos. Si bien, muchos de los operadores alimentarios son los principales ejecutores de este tipo de fraudes, también suelen ser víctimas de sus proveedores de materias primas.

En los entornos industriales tan importante es la integridad de las materias primas como la premura a la hora de fabricar. En determinados casos, las industrias reciben la materia prima, la cual es procesada prácticamente al momento, ya sea por demandas de la producción o por que el material es perecedero y no es conveniente almacenarlo largos tiempos. Esto provoca que se dispongan tiempos mínimos para realizar un control de calidad, que en la mayoría de los casos trata de sencillos análisis que aseguran en cierta medida que cumplen con la adecuación al uso y las medidas sanitarias para procesar el producto, pero que en raras ocasiones ponen foco en la adulteración. Por lo general, los análisis de adulteración requieren de equipo especializado y un personal formado capaz de interpretar los datos o de accesibilidad a un laboratorio externo que los pueda analizar, pero esta última opción repercute muy negativamente en la agilidad de obtener un veredicto rápido para la aceptación de una materia prima antes de ser procesada, además de las implicaciones económicas que tiene. Es por este motivo que la industria requiere de nuevas técnicas que permitan una rápida identificación de adulterantes y que cualquier operario de forma sencilla pueda ejecutar.

1.2 Objetivos del trabajo

a. Objetivo general

1. Desarrollo de un modelo de aprendizaje automático que introduciéndose como entrada datos de espectroscopía de masas liquida, determine la especie a la cual pertenece la leche y de un segundo modelo de aprendizaje automático que sea capaz de discriminar leche de cabra adulterada en un grado del 0% al 10% de la que no e implementarlo de forma práctica en una herramienta.

b. Objetivos específicos

1. Reducir la dimensión de los conjuntos de datos.
2. Incrementar el número de muestras artificialmente.
3. Construir modelos de machine learning con el requisito que superen una exactitud (*accuracy*) del 90% y evaluar su rendimiento.
4. Comparar el rendimiento de los algoritmos de machine learning antes y después de realizar las tareas de aumento de datos y reducción de dimensionalidad y escoger el mejor escenario de trabajo.
5. Determinar cuál de los modelos sometidos a estudio se adapta mejor a los datos y al problema a resolver.
6. Crear una aplicación web destinada al uso por usuarios externos y que implemente de forma rápida y sencilla, para cada clase de problema a resolver, los algoritmos y escenarios de trabajo seleccionados.

1.3 Impacto ético-social, sostenibilidad y diversidad

El presente trabajo tiene como objetivo encontrar un método que permita destapar fraude alimentario y adulteración de leches de forma rápida y accesible a diferentes tipos de usuarios. Si bien, las leches de vaca, cabra, camello y oveja son el objetivo, el presente trabajo puede servir como ejemplo o como base para aplicar la misma técnica a otras matrices alimentarias o para la prevención de otros fraudes.

Tratando el compromiso ético y global del trabajo, este aborda únicamente la dimensión de comportamiento ético y responsabilidad social ya que no tiene impactos, ni positivos ni negativos, en las dimensiones de sostenibilidad ni en la de diversidad, género y derechos humanos y, además, el objetivo principal del trabajo está diseñado ad hoc para poner solución a los problemas que se enmarcan en la primera dimensión.

El alcance de este trabajo impacta directamente y de forma positiva a los objetivos de desarrollo sostenible (en adelante ODS) 2 y 3, correspondientes a hambre y seguridad alimentaria, y salud respectivamente. El concepto de seguridad alimentaria es el nexo entre este trabajo y los dos ODS mencionados anteriormente. La FAO define literalmente la seguridad alimentaria como “el acceso físico, social y económico en todo momento, de las personas a alimentos suficientes, inocuos y nutritivos que satisfacen las necesidades energéticas

diarias y preferencias alimentarias para llevar una vida activa y sana" [5]. Como bien predica la definición anterior, es necesario que estos alimentos sean nutritivos e inocuos. La inocuidad de los alimentos es también conocida como seguridad alimentaria, la cual, según la OMS, se refiere a "la producción, manejo, almacenamiento y preparación de alimentos de manera que se eviten infecciones y contaminación en la cadena productiva y que contribuyan a garantizar la calidad y salubridad de estos mismos para que no repercuta negativamente en la salud de los consumidores" [6]. Como se ha dicho anteriormente en el apartado introductorio, el fraude alimentario, si bien en ocasiones tiene únicamente repercusiones económicas, muchas otras pueden afectar directamente en la salud de los consumidores, introduciendo ingredientes no declarados que pueden malmeter la integridad de los consumidores por diferentes vías, o bien mermar las capacidades nutritivas originales del alimento.

Por lo tanto, no se plantea que este trabajo repercuta positivamente en alguno de estos ODS por separado, dado que el concepto de seguridad alimentaria abarca ambos objetivos con un significado distinto en cada uno, pero que al fin y al cabo no se entiende el uno sin el otro. La consecución del hambre cero y de la salud y el bienestar de las personas pasa necesariamente, en parte, por generar una capacidad productiva y puesta en el mercado de alimentos seguros y nutritivos. Aquí es donde se enmarca la tarea de prevención de fraude alimentario, la cual afecta a todos los consumidores de alimentos independientemente del país, aunque con especial vulnerabilidad en aquellas regiones menos desarrolladas donde escasean las técnicas, infraestructura y la formación para prevenir y detectar este tipo de prácticas.

1.4 Enfoque y método seguido

El presente trabajo se basa en los datos analizados en el estudio "Speciation and milk adulteration analysis by rapid ambient liquid MALDI mass spectrometry profiling using machine learning" [7] y obtenidos del repositorio público de la Universidad de Reading [8]. El propio estudio se divide en dos partes:

- Un estudio de especiación que trata de discriminar muestras de cuatro tipos de leche (vaca, cabra, oveja y camello). Este genera el conjunto de datos llamado "Lipids different species.data.csv" con entre 21 y 23 replicados por cada clase y con 600 variables correspondientes a las relaciones de masa/carga (m/z) en las que se ha obtenido señal.
- Un estudio de adulteración de leche de cabra adulterada con un 5% y 10% de leche de vaca. Este estudio genera 2 conjuntos de datos, uno para cada grado de adulteración. En total se tienen entre 48 y 54 muestras por clase a base de 2 replicados biológicos, 2 replicados de mezcla y 2 repeticiones de mezcla, con un total de 1600 variables correspondientes a las relaciones de masa/carga en las que se ha obtenido señal. Cada

uno se clasifica en dos clases en función si la muestra está adulterada (al 5% o al 10% según el conjunto de datos) o se trata de 100% leche de cabra.

En ambos casos se trata de datos de espectrometría de masas AP-MALDI sobre líquidos. En todos los casos, se dispone de una etiqueta que indica a qué especie o clase de muestra (adulterada o no) corresponde cada muestra.

En el presente trabajo, emplea una parte de los datos como herramienta para entrenar el modelo y otra parte del conjunto de datos para validar la capacidad de clasificación de los diferentes modelos con nuevas instancias de datos.

Se quiere reproducir los mismos estudios de especiación y adulteración que propone Piras et al. [7], diferenciándose en:

- La reducción de la dimensionalidad el conjunto de datos: Como se explicará en este mismo apartado posteriormente, los conjuntos de datos disponen de una gran cantidad de variables que superan con creces el número de muestras. Eso supone la introducción de ruido en el conjunto de datos que puede enmascarar el efecto de las variables más discriminantes.
- El aumento del número de muestras: Se dispone de un número limitado de muestras, inferior a las 50 muestras por cada clase. Puede provocar sesgo y sobreajuste de los modelos, por lo que se quiere realizar un aumento de muestras artificial.
- El estudio de adulteración se realiza con un único conjunto de datos conjuntamente para las leches del 5% y 10% de adulteración, a diferencia de Piras et al. [7] que lo lleva a cabo por separado. Además, a diferencia de Piras et al [7], el estudio de adulteración propuesto en el presente trabajo, se enfoca en la detección de la leche adulterada de la que no lo está, independientemente del grado de adulteración de esta. Por ello se trabaja con dos clases, leche no adulterada correspondiente a leche 100% de cabra y leche adulterada, que engloba las clases originales de leche de cabra al 10% y 5% de adulteración.
- Los algoritmos de machine learning a emplear: Se emplean algoritmos de machine learning alternativos al análisis discriminante lineal empleado por Piras et al. [7], como son random forests (RF), support vector machines (SVM) y artificial neural networks (NN).
- Establecimiento de un umbral mínimo de eficiencia: Se fija un 90% mínimo de eficiencia, en cuanto a exactitud se refiere, como objetivo a perseguir en la construcción de los modelos, pero también como un requisito para seleccionar el mejor modelo. La finalidad es construir modelos con buena capacidad de predicción e igualar los resultados de estos al de los

modelos obtenidos en estudios similares que también emplean algoritmos de machine learning para la detección del fraude alimentario empleando conjuntos de datos analíticos. Este valor mínimo de exactitud se fija en base a los resultados que han sido obtenidos en dichos estudios (véase el apartado *2. Estado del arte*), los cuales rondan en gran frecuencia el 90%.

- La implementación en una aplicación web: La finalidad del estudio reside en la creación de una herramienta que sea capaz de predecir el fraude en nuevas muestras, para la aplicación práctica de los modelos seleccionados.

Si bien, en Piras et al. [7] se obtiene una precisión del 100% en el estudio de especiación y una sensibilidad y especificidad del 92.5% y 94.5% para la leche adulterada al 5% y del 99.2% y 99.1% para la leche adulterada al 10%, el presente trabajo trata de encontrar un algoritmo, que a pesar de una precisión menor tenga una buena capacidad de generalización, con la motivación de ser empleado como ejemplo de posible herramienta de aplicación práctica en industria para la detección preventiva de los fraudes mencionados.

Para la construcción de los modelos de machine learning se siguen los pasos propuestos en Lantz 2015 [9] y que se muestran a continuación:

- Recopilación de los datos.
- Exploración y preparación de los datos.
- Construcción de los modelos y entrenamiento: Hay que comentar que en este aparatado se han planteado cuatro escenarios de trabajo en los cuales se implementarán los mismos algoritmos con las mismas mejoras para tratar de averiguar qué escenario se ajusta mejor a las necesidades de cada problema:
    - Escenario con el conjunto de datos sin transformar.
    - Escenario con datos aumentados.
    - Escenario con dimensiones reducidas.
    - Escenario combinado con datos aumentados y dimensiones reducidas.
- Validación del modelo.
- Mejora del modelo.
- Validación del modelo mejorado.

Sobre este esquema se ha elaborado la planificación del trabajo que se puede ver detallada en el siguiente apartado.

1.5 Planificación del Trabajo

1.5.1 Tareas

Como se ha comentado con anterioridad, el trabajo consta de la reproducción de dos estudios, uno de especiación y otro de adulteración y por último el desarrollo de una planificación web. Estos dos estudios se han repartido entre las entregas de las PEC2 y PEC3 respectivamente, contando como hitos. La tabla 1 detalla todas las tareas y sus plazos de ejecución:

Tabla 1. Tareas y plazos de ejecución.

Tarea	Fecha de inicio	Fecha de finalización
PEC1 - Definición del TFM y plan de trabajo		
Elección de tema y búsqueda de datos	01/03/2023	03/03/2023
Definición de objetivos	04/03/2023	05/03/2023
Justificar y contextualizar el trabajo	06/03/2023	09/03/2023
Definición de enfoque y método a seguir	10/03/2023	18/03/2023
Definición del plan de trabajo	13/03/2023	14/03/2023
Entrega del plan de trabajo (PEC1)	19/03/2023	20/03/2023
Correcciones y mejoras PEC1	27/03/2023	02/04/2023
PEC2 - Desarrollo del trabajo. Fase 1		
Exploración de librerías y métodos a usar	21/03/2023	24/03/2023
Exploración de datos de especiación	21/03/2023	22/03/2023
Preprocesado de datos de especiación	23/03/2023	24/03/2023
Construcción de modelos pre-transformación	27/03/2023	29/03/2023
Validación de modelos pre-transformación	30/03/2023	30/03/2023
Mejorar modelos modificando parámetros	31/03/2023	05/04/2023
Validación de modelos mejorados	06/04/2023	06/04/2023
Aumento de muestras	25/03/2023	28/03/2023
Reducción de dimensionalidad	29/03/2023	01/04/2023
Construcción de modelos para especiación	02/04/2023	07/04/2023
Validación de modelos pre-transformación	08/04/2023	08/04/2023
Mejorar modelos modificando parámetros	09/04/2023	18/04/2023
Validación de modelos mejorados	18/04/2023	18/04/2023
Selección del mejor modelo	20/04/2023	22/04/2023
Selección del mejor escenario de trabajo	20/04/2023	22/04/2023
Entrega de resultados (PEC2)	23/04/2023	24/04/2023
Correcciones y mejoras PEC2	08/05/2023	14/05/2023
PEC3 - Desarrollo del trabajo. Fase 2		
Exploración de los datos de adulteración	25/04/2023	25/04/2023
Preprocesado de los datos	26/04/2023	26/04/2023

Tarea	Fecha de inicio	Fecha de finalización
Construcción de modelos pre-transformación	27/04/2023	28/04/2023
Validación de modelos pre-transformación	29/04/2023	29/04/2023
Mejorar modelos modificando parámetros	30/04/2023	05/05/2023
Validación de modelos mejorados	06/05/2023	06/05/2023
Aumento de muestras	28/04/2023	30/04/2023
Reducción de la dimensionalidad	01/05/2023	02/05/2023
Construcción de modelos para adulteración	03/05/2023	07/05/2023
Validación de modelos	08/05/2023	08/05/2023
Mejorar modelos modificando parámetros	09/05/2023	13/05/2023
Validación de modelos mejorados	14/05/2023	14/05/2023
Selección del mejor modelo	15/05/2023	17/05/2023
Selección del mejor escenario de trabajo	15/05/2023	17/05/2023
Implementación de modelos en una app web	17/05/2023	27/05/2023
Entrega de resultados (PEC3)	28/05/2023	29/05/2023
Correcciones y mejoras PEC3	05/06/2023	11/06/2023
PEC4 - Cierre de la memoria y de la presentación		
Escritura de la memoria	30/05/2023	10/06/2023
Diseño de la presentación en PPT	11/06/2023	14/06/2023
Grabación y edición de la presentación	15/06/2023	18/06/2023
Revisión y entrega de memoria y presentación (PEC4)	19/06/2023	20/06/2023
PEC5 - Defensa pública		
Defensa pública del trabajo		06/07/2023

1.5.2 Calendario

Se puede apreciar en la figura 2 el calendario de ejecución de las tareas. Este mismo calendario se ha elaborado con la herramienta online *Tom's Planner* y puede verse también de forma más interactiva en el siguiente enlace: https://plan.tomsplanner.com/public/malopezgonzalez

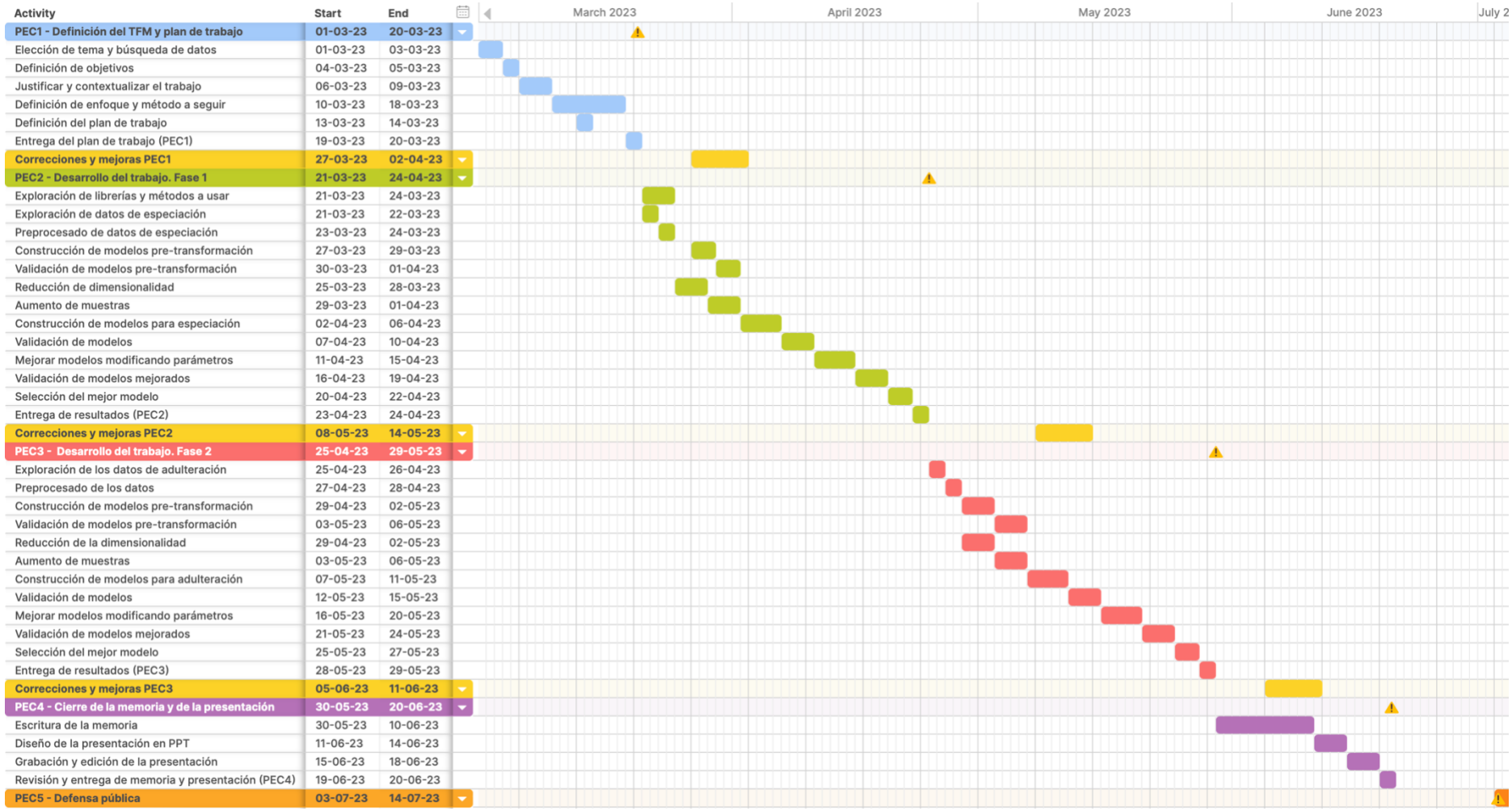

Figura 2. Calendario de ejecución de las distintas tareas. El icono ⚠ marca los hitos del trabajo.

1.5.3 Hitos

La tabla 2 muestra los hitos del trabajo, correspondientes a las diferentes entregas de PECs, que también se muestran en el calendario (figura 2).

Tabla 2. Hitos y términos de ejecución de estos.

Descripción	Fecha
Definición del TFM y plan de trabajo	20/03/2023
Desarrollo del trabajo. Fase 1	24/04/2023
Desarrollo del trabajo. Fase 2	29/05/2023
Cierre de la memoria y de la presentación	20/06/2023
Defensa pública	06/07/2023

1.6 Sumario de productos obtenidos

Se obtiene como producto final, la presente memoria del trabajo, el material audiovisual para su presentación y la aplicación web desarrollada, a la cual se puede acceder mediante el siguiente enlace:

https://miguelangellg.shinyapps.io/detectorFraudeLeche/

Puede consultarse todo el código empleado para consecución de los resultados, así como los conjuntos de datos originales con los cuales se han entrenado y validado los modelos en el repositorio de acceso público siguiente:

https://github.com/malopezgonzalez/Deteccion-de-fraude-en-leche

En este repositorio se pueden encontrar varios elementos:

- **Resultados parciales:** Son archivos en formato Rmarkdown (*.rmd*) donde consta el código comentando todos los pasos seguidos en el planteamiento: carga de los datos, implantación de los algoritmos, mejora y elección de los mejores modelos y escenarios de trabajo. Consta de un archivo por estudio, los cuales se encuentran en las carpetas de cada estudio, *1.Estudio_especiacion* y *2.Estudio_adulteracion.*

- **Conjuntos de datos:** Se adjuntan los conjuntos de datos en formato *.csv* con los cuales se ha llevado a cabo el estudio y se ha entrenado y validado el modelo. Se encuentran en las carpetas de cada estudio, *1.Estudio_especiacion* y *2.Estudio_adulteracion.*

- **Aplicación web:** Se adjunta el código de la aplicación web en formato *.R* y los archivos dependientes. Se encuentran en la carpeta *3.WebApp.*

- **Espacios de trabajo:** Archivos en formato *.Rdata.* que contienen los modelos y funciones para ser usados en la aplicación web. Se encuentran en la carpeta *3.WebApp.* Además, se adjunta el código empleado para su obtención, que es una selección del código esencial de los resultados parciales en la carpeta *4.codigo_espacios_trabajo*.

1.7 Breve descripción de los otros capítulos de la memoria

La memoria se ha planteado en los apartados que se detallan a continuación y que suceden al presente apartado *1. Introducción*:

2. **Estado del arte:** Se realiza una búsqueda y revisión de bibliografía relevante en el que se emplean algoritmos de aprendizaje automático para resolver para detección de fraude y adulteraciones alimentarias.

3. **Metodología:** Se detalla los métodos (algoritmos de machine learning y técnica de reducción de dimensiones y aumento de datos) y materiales (software) se han empleado para la consecución de los objetivos planificados.

4. **Resultados:** Tal y como indica su nombre, se realiza una presentación de los resultados obtenidos, mostrando la precisión obtenida en cada algoritmo y variación aplicada en los escenarios de trabajo propuestos.

5. **Discusión:** Se interpretan los resultados obtenidos comparándolos entre ellos.

6. **Conclusiones:** Justificación de la elección de los modelos y los escenarios de trabajo idóneos. Se realiza una revisión de los objetivos planteados y los impactos ético-sociales, de sostenibilidad y de diversidad planteados al principio de esta memoria; en relación con los resultados finalmente obtenidos. También se proponen líneas de futuro del presente trabajo.

7. **Glosario:** Se definen los términos y acrónimos más relevantes empleados en el presente trabajo y presentado en orden alfabético.

8. **Bibliografía:** Se numeran por orden de aparición en el texto las fuentes de información empleadas para la realización del trabajo.

2 Estado del arte

La premura para procesar ciertos alimentos dentro de la industria alimentaria unida a la necesidad de garantizar la calidad y, sobre todo, la seguridad alimentaria de los productos obtenidos ha creado la necesidad de métodos de análisis que provean una rápida respuesta a la par de una ejecución de bajo coste y sencilla. El machine learning es una técnica que cumple con estos requisitos. Varios autores han aplicado este tipo de técnicas dentro del ámbito de la detección del fraude y, siendo más generales, en el ámbito de análisis de alimentos.

En 1996, Holland et al. [10], emplearon Partial Least Squares Regression para detector adulteración en purés de fresa. Las muestras fueron analizadas mediante Fourier transform infrared spectroscopy, dando lugar a un conjunto de datos de alta dimensionalidad con 235 variables. La técnica logró detectar adulterantes con una tasa de acierto del 94.3% y una tasa del 96.6% cuando analizaron fruta con dos años más de antigüedad con la usada para la construcción del modelo. Además, en una prueba a ciegas consiguió clasificar correctamente 22 de 23 muestras analizadas.

Cui et al. [11] analizaron muestras de té negro de diferentes regiones geográficas con el objetivo de detectar fraude en el origen geográfico del té. Emplearon un conjunto de datos obtenido del análisis de hojas de té mediante resonancia magnética nuclear. Emplearon random forests, support vector machines, análisis discriminante lineal y k nearest neighbors para la identificación del origen, resultando en una exactitud del 92.7%, 91.8%, 86.3% y 86.3% respectivamente.

Momeny et al. [12] emplea una red neuronal convolucional, además de realizar combinaciones de redes neuronales convolucionales y algoritmos como árboles de decisión, k nearest neighbors, random undersampling boosted trees y support vector machines, para detectar adulteración y fraude en muestras de azafrán. Para ello emplea imágenes tomadas con un smartphone, consiguiendo un 99.5% de exactitud en sus predicciones.

Li et al. [13], analizaron 210 muestras de Fourier transformed near-infrared repartidas entre carne de vacuno, gel a base de sangre de cerdo y gel a base de sangre de cerdo pura, con la intención de encontrar el modelo que mejor detectara la adulteración de la carne de vacuno. Se emplearon los algoritmos de partial least squares, support vector machines y extreme learning machines, consiguiendo un 100% de exactitud en el modelo creado con este último algoritmo.

Yakar et al. [14] probaron diferentes algoritmos de machine learning en datos generados por cromatografía de gases para crear un modelo que

permitiera detectar muestras de aceite de oliva adulteradas con otros aceites vegetales. Emplearon support vector machines, k nearest neighbors y árboles de decisión, de entre los cuales support vector machines proporcionó unas exactitudes del 94.6%, 96.4% y 98.2% para los 3 tipos de muestras adulterades.

Lima et al. [15] emplearon redes neuronales perceptrón multicapa y CART (classification and regression trees) para determinar la adulteración de la leche con suero de queso en concentraciones de 1, 2, 5, 10, 15, 20, 25 y 30%. Para ello se emplearon 512 muestras obtenidas mediante Fourier transformed infrared spectroscopy obteniendo unas exactitudes del 96.2% para CART y 97.8% para las redes neuronales, demostrando con ellos que se trata de un potencial método de cribado para detectar leche adulterada en laboratorios de calidad.

Tian et al. [16] realizaron un estudio sobre adulteración de leche cruda con aceites vegetales. Los datos fueron obtenidos mediante una nariz electrónica basada en la técnica analítica flash gas chromatography. Emplearon los algoritmos de random forests, multilayer precreptron (redes neuronales), support vector machines y XGBoost, siendo random forests y support vector machines los que mejores exactitudes obtuvieron, con un valor de 100% y 95.65%.

Nataranjan & Ponnusamy [17] realizaron un estudio para la clasificación de verduras orgánicas y ecológicas (concretamente tomate, chile verde y berenjena), enfocado a ser un método de bajo coste y portable. Para ello emplearon un sensor multiespectral trabajando en el espectro visible, ultravioleta e infrarrojo. Para ello emplearon los algoritmos de random forests y neural networks los cuales consiguieron un 92% y un 89% de precisión respectivamente. Además, integraron la difusión de los resultados empleando una página web emparejada con un módulo de “internet de las cosas” para hacer accesibles los resultados.

Calle et al. [18] se centraron en la adulteración de zumos de frutas. Se trabajó con zumos de frutas 100% exprimidos (piña, naranja y manzana) y con sus muestras adulteradas con zumo de uva a concentraciones de 5%, 10%, 15%, 20%, 30%, 40% y 50%. Obtuvieron los datos mediante Fourier transformed infrared spectroscopy, sobre los cuales aplicaron support vector machines, random forests y análisis discriminante lineal, obteniendo una exactitud en los tres algoritmos por encima del 97% cuando se quería detectar la adulteración.

Huang et al. [19] llevaron a cabo un estudio de detección de leche en polvo adulterada con datos de laser-induced breakdown spectroscopy. Se analizaron 25 muestras de leche en polvo adulteradas con 4 proteínas exógenas distintas. Se emplearon algoritmos de machine learning clásicos como support vector machines, k nearest neighbors, random forests y linear discriminant analisys. De entre estos support vector machines logró una exactitud del 93.9%. También se

empleó una red neuronal convolucional para la detección de la adulteración, resultando en una mejor exactitud que alcanzó el 97.8%.

El presente trabajo tiene como objetivo conseguir resultados similares o mejores a los obtenidos por los recientes estudios. Si bien varios autores emplean diferentes algoritmos de machine learning con buenos resultados y se trabajan conjuntos de datos con características similares a los que se han tratado en el presente estudio (espectros de alta dimensionalidad), no se han encontrado estudios en los que se realicen técnicas de aumentos de datos en conjuntos de datos tabulares y son pocos los que aplican técnicas de reducción de la dimensionalidad. Este hecho diferencia el presente trabajo de los recientes estudios, mostrando la efectividad de las técnicas de aumento de datos y reducción de dimensiones para determinar el mejor espacio de trabajo.

Otro rasgo diferencial del presente trabajo frente a los mencionados es la aplicación práctica de los modelos seleccionados como óptimos al uso que se les espera dar. Si bien, son pocos los autores mencionados los que han integrado alguna en alguna herramienta los modelos creados. Es por eso por lo que se ha realizado el desarrollo de la aplicación web como herramienta de rápida respuesta, bajo coste y fácil compresión destinada a los operadores alimentarios o laboratorios que deseen realizar un control de fraude en los lotes de leche adquiridos.

3 Materiales y métodos

3.1 Datos de trabajo

Como se ha mencionado en el apartado introductorio, el presente trabajo se ha llevado a cabo a partir de los datos de generados por Piras et al. [7] en su estudio y que se encuentran alojados en el repositorio de la Universidad de Reading [8]. Se trata de un total de 3 conjuntos de datos para los dos tipos de estudios. Las variables de interés y con las que se trabajará son las señales obtenidas por cada fracción masa/carga (m/z) del espectro de masas analizado. Los datos se describen a continuación:

3.1.1 Datos para el estudio de especiación

Los datos se alojan en el archivo “Lipids different species.data.csv” Una vez eliminadas las columnas de información irrelevante (veáse el apartado 3.2.1 *Limpieza de información irrelevante*) tiene unas dimensiones de 87 muestras y 601 variables, de las cuales una variable corresponde a la indicación de clase de cada muestra. Las variables de interés se extienden desde la fracción masa/carga 400.5 hasta la 999.5

Se comprueba que el conjunto de datos no tiene datos perdidos que puedan dificultar el estudio.

Se cuantifican los valores atípicos (veáse el apartado *3.2.2 Valores atípicos (outliers)* para más información sobre el método de detección y las medidas adoptadas) por cada una de las clases y variables obteniendo los siguientes resultados: Vaca (cow) 341 valores, cabra (goat) 197 valores, camello (camel) 236 valores y oveja (sheep) 340 valores. Se desestima su eliminación ya que para ello habría eliminar una gran cantidad de muestras o bien imputar una gran cantidad de valores, con lo que se modificaría en gran medida los datos originales y se sesgaría el modelo.

En cuanto la cantidad de muestras de cada clase, la figura 3 muestra que existe un ligero desbalanceo entre clases, aunque por lo general la cantidad de muestras de cada una es bastante igualada teniendo 21 muestras para la clase camello, 23 muestras para la clase vaca, 21 muestras para la clase cabra y 22 muestras para la clase oveja. Se considera que el conjunto de datos está balanceado por lo que no se aplica ninguna modificación a este conjunto de datos.

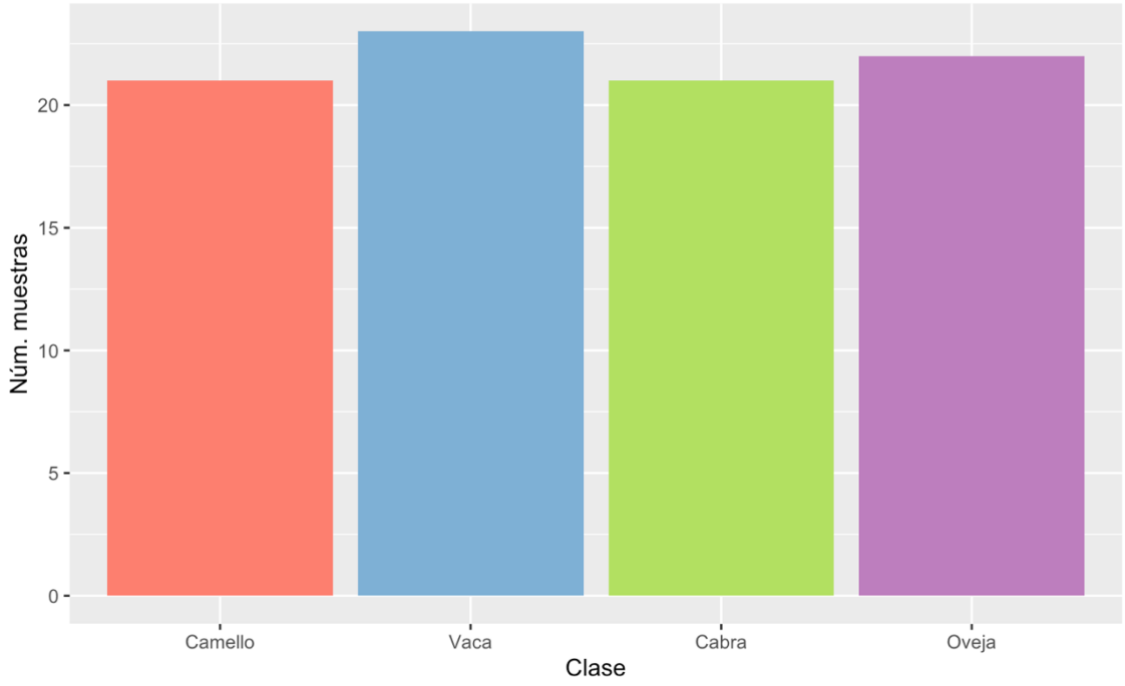

Figura 3. Cantidad de muestras por clase en el conjunto de datos de especiación.

En la tabla 3 se puede apreciar un resumen de las 10 primeras variables. La naturaleza de estas variables es la misma, por lo que ante la dificultad de revisar cada una de las 600 variables numéricas se hará una extrapolación de las suposiciones halladas en la exploración.

Tabla 3. Exploración de datos para el conjunto de especiación.

	Media	Desv. est.	Mín.	Máx.	Rango	IQR	Dist. normal?
X400.5	4,4E-04	2,0E-04	1,8E-04	1,5E-03	1,4E-03	2,0E-04	NO
X401.5	5,0E-04	1,9E-04	2,0E-04	1,2E-03	9,8E-04	2,5E-04	NO
X402.5	4,9E-04	2,2E-04	1,7E-04	1,3E-03	1,1E-03	2,7E-04	NO
X403.5	3,8E-04	1,4E-04	1,5E-04	7,8E-04	6,3E-04	1,8E-04	NO
X404.5	2,8E-04	1,3E-04	9,6E-05	6,9E-04	5,9E-04	1,6E-04	NO
X405.5	3,2E-04	1,4E-04	1,4E-04	9,9E-04	8,5E-04	1,6E-04	NO
X406.5	3,7E-04	2,1E-04	1,3E-04	1,1E-03	9,3E-04	2,7E-04	NO
X407.5	4,6E-04	3,1E-04	1,6E-04	2,1E-03	2,0E-03	3,1E-04	NO
X408.5	2,7E-04	1,3E-04	1,0E-04	7,4E-04	6,4E-04	1,5E-04	NO
X409.5	1,8E-03	1,9E-03	4,1E-04	1,5E-02	1,5E-02	1,1E-03	NO

Se aplica la prueba de Shapiro-Wilk por cada variable para determinar su normalidad, obteniéndose 538 variables no normales y 62 que siguen una distribución normal. Aunque la falta de esta no es un problema para los métodos que se quieren aplicar y que se verán en próximos apartados. Por otra parte, se puede ver que el rango de cada una de las variables no es el mismo, por lo que será necesario realizar un escalado de los datos para que las variables con

valores más extremos no condicionen los resultados del estudio. En el apartado *3.2.3 Escalado de datos* se muestra más información sobre el tipo de escalado llevado a término.

3.1.2 Datos para el estudio de adulteración

Los datos se alojan en dos archivos separados “Adulteration 30 seconds 0-5%.data.csv” y “Adulteration 30 seconds 0-10%.data.csv”. Para trabajar este estudio se han juntado los dos conjuntos de datos para trabajar con un único conjunto de datos. Ambos conjuntos de datos contienen muestras de leche 100% de cabra como referencia de leche sin adulterar. En el conjunto de datos de leche adulterada al 0-5% se dispone un total de 50 muestras de leche no adulterada, mientras que en el conjunto de datos de leche adulterada al 0-10% la cantidad de muestras de leche no adulterada es de 48. Estas muestras no adulteradas de ambos conjuntos son coincidentes, por lo que a la hora de juntar ambos conjuntos de datos se borran aquellas que estén duplicadas, resultando en un total de 50 muestras 100% leche de cabra no adulteradas. Por otra parte, el resto de las muestras se encuentran etiquetadas como leche de cabra al 95% (Goat 95) y leche de cabra al 90% (Goat 90). Estas clases son renombradas como “Adulterada”, ya que como se ha comentado con anterioridad, se pretende detectar la adulteración de la leche independientemente del grado en que esta se da. Tras la eliminación de variables irrelevantes, el conjunto final de datos con el que se trabaja cuenta con unas dimensiones de 152 muestras y 1601 variables, de las cuales una variable corresponde a la indicación de clase de cada muestra. Las variables de interés se extienden desde la fracción masa/carga 400 hasta la 1999.

Se comprueba que el conjunto de datos no tiene datos perdidos que puedan dificultar el estudio.

Se cuantifican los valores atípicos por cada una de las clases y variables obteniendo los siguientes resultados: No adulterada (leche 100% cabra) 1676 valores y adulterada (leches diluidas al 5 y 10%) 4860 valores. Se desestima la eliminación de los valores atípicos por los mismos motivos que se ha desestimado para los datos de especiación. Cabe comentar que para la clase de leche adulterada se encuentra una cantidad de valores atípicos muy superior, debido a que, cómo se verá a continuación, la cantidad de muestras de esta clase es superior y a que esta clase contiene muestras de dos clases distintas, con lo que es probable que si la búsqueda de valores atípicos se hubiera hecho sobre las clases originales la cantidad hubiera sido menor.

En cuanto a la cantidad de muestras de cada clase, la figura 4 muestra que existe un notable desbalanceo entre las dos clases, teniendo 102 muestras para la clase de leche adulterada y 50 muestras para la clase de leche no adulterada. Este hecho es importante, ya que será necesario introducir el estadístico Kappa

y valorarlo junto el resto de las métricas (exactitud, sensibilidad y especificidad, tal y como se comenta en el apartado *3.7 Métricas para a la evaluación del rendimiento del modelo*), para obtener información sobre el rendimiento de los modelos, ya que, al estar las clases desbalanceadas, un modelo puede obtener aparentemente buenos resultados con tan solo predecir la clase más frecuente.

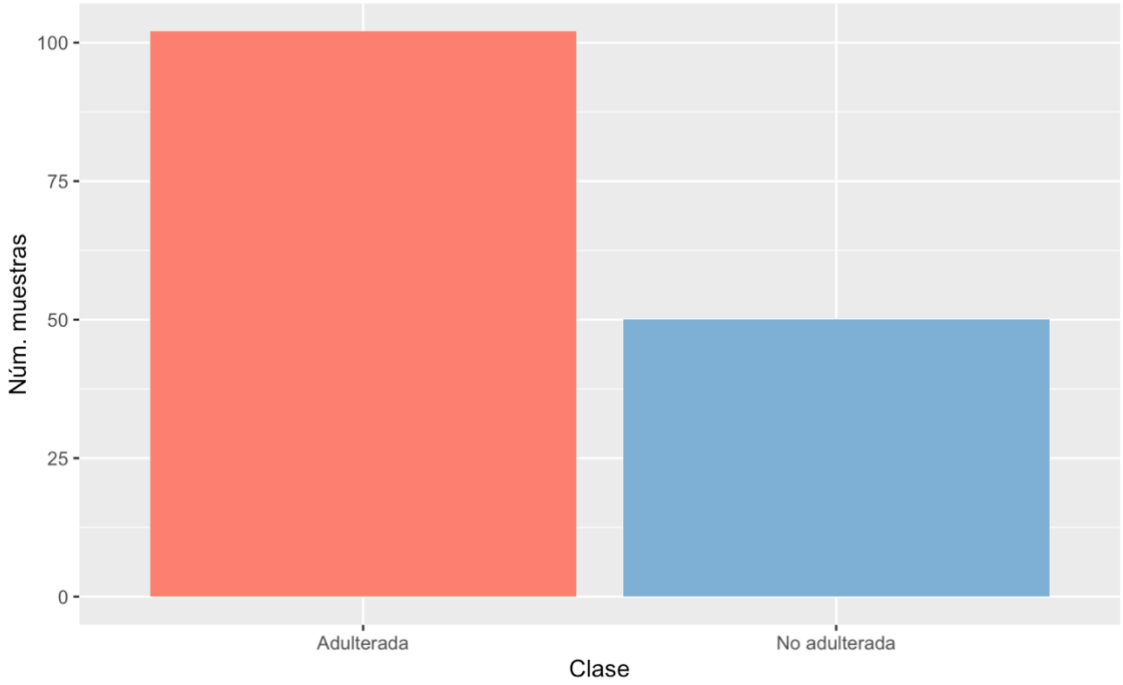

Figura 4. Cantidad de muestras por clase en el conjunto de datos de adulteración.

En la tabla 4 se puede apreciar un resumen de las 10 primeras variables. La naturaleza de estas variables es la misma, por lo que ante la dificultad de revisar cada una de las 1600 variables numéricas se hará una extrapolación de las suposiciones halladas en la exploración, de la misma manera que se ha hecho para los datos de especiación.

Tabla 4. Exploración de datos para el conjunto de adulteración.

	Media	Desv. Est.	Mín.	Máx.	Rango	IQR	Dist. Normal?
X400	3,3E-05	9,7E-06	1,2E-05	6,1E-05	4,9E-05	1,4E-05	NO
X401	4,4E-05	1,2E-05	2,4E-05	8,6E-05	6,2E-05	1,8E-05	NO
X402	3,5E-05	9,4E-06	1,6E-05	6,4E-05	4,8E-05	1,3E-05	NO
X403	3,4E-05	1,0E-05	1,5E-05	7,6E-05	6,0E-05	1,5E-05	NO
X404	3,7E-05	9,2E-06	1,7E-05	6,0E-05	4,3E-05	1,1E-05	NO
X405	5,4E-05	1,2E-05	3,2E-05	1,1E-04	7,5E-05	1,6E-05	NO
X406	4,8E-05	1,2E-05	3,0E-05	8,8E-05	5,8E-05	1,4E-05	NO
X407	6,1E-05	1,7E-05	3,6E-05	1,2E-04	7,9E-05	2,5E-05	NO
X408	4,0E-05	1,1E-05	2,2E-05	7,2E-05	5,0E-05	1,5E-05	NO
X409	1,7E-04	6,4E-05	7,9E-05	3,6E-04	2,9E-04	8,4E-05	NO

El resultado es similar al anterior. Se aplica la prueba de Shapiro-Wilk para determinar la normalidad de las variables, obteniendo 645 variables no normales frente a 955 variables normales. Como se ha dicho en la exploración del conjunto de especiación, la no normalidad no supone un inconveniente para los métodos aplicados. Se puede ver que el rango de cada una de las variables, aunque muy similar, no es el mismo, por lo que será también necesario realizar un escalado de los datos para que las variables con valores más extremos no condicionen los resultados del estudio. De esta forma se asegura trabajar en las mismas condiciones que en el estudio anterior.

3.2 Preprocesado de datos

3.2.1 Limpieza de información irrelevante

Se ha eliminado aquella información contenida en los conjuntos de datos que no tiene relevancia en el resultado, tales como etiquetas, nombres de archivo, etc. Para ambos conjuntos de datos trabajado se ha procedido de la misma forma y se han eliminado las siguientes columnas: La columna 2 se corresponde al nombre del archivo RAW que ha generado el espectro, no es de utilidad para el estudio. Las columnas 3 y 4 de los conjuntos de datos se corresponden con las variables *Start.scan* y *End.scan* al tiempo en el que se inyectó cada muestra en el cromatograma. La columna 5 se corresponde a la variable *Sum.* que indica la intensidad total medida en el rango específico de masa/carga (m/z). Son variables empleadas para el mapeo de cada archivo específico en el cromatograma, por lo que se excluirán del estudio. Además, se ha confirmado que estas no son empleadas en Piras et al. [7].

3.2.2 Valores atípicos (outliers)

Se realiza la búsqueda de valores atípicos en los conjuntos de datos de partida para el entrenamiento y testeo de los modelos con tal de conocer su presencia y magnitud. Para ello se emplea el mismo criterio utilizado en Lantz 2015 y Renesh Bedre 2022 [9,20], en el que se consideran outliers todos aquellos valores que se encuentren 1.5 veces el valor del rango intercuartílico (IQR) por encima del tercer cuartil y 1.5 veces el valor del IQR por debajo del primer cuartil. La detección de outliers se aplica dentro de cada una de las clases por cada variable por separado. Finalmente se desestima su eliminación o su imputación, ya que al ser los conjuntos de datos pequeños pueden modificar las muestras hasta el punto a que se parezcan mucho entre sí y, en consecuencia, sesgar el modelo.

3.2.3 Escalado de datos

Se ha optado por realizar un escalado de datos mediante estandarización por z-score, con lo que se modifica el rango de los datos, se centra la media a 0 y la desviación estándar a 1 [9]:

$$X_{nueva} = \frac{X - Media(Variable)}{DesvEstd(Variable)}$$

Donde *X* es el valor que se quiere transformar perteneciente a una determinada variable.

Para no introducir sesgos en el conjunto de datos, se decidió escalar conjunto de datos de entrenamiento (train) en base a la media y la desviación estándar de cada una de sus variables. Posteriormente, el conjunto de validación (test) se escala en el mismo rango que el subconjunto de entrenamiento usando la media y la desviación estándar empleadas para escalar el mismo subconjunto de entrenamiento. En caso de que se quisiera trabajar con una nueva muestra o conjunto de datos externo al que se está trabajando, se aplicaría el mismo método de escalado seguido con el subconjunto de validación.

3.3 Tipo de aprendizaje automático y algoritmos empleados

Existen diversos tipos de aprendizaje automático, que según el autor pueden variar entre 2 y 5, aunque los 2 fundamentales son los siguientes:

- **Aprendizaje supervisado:** Se entrena al modelo en cuestión proporcionándole unos datos conocidos y etiquetados con los nombres de las clases que posteriormente se quieren predecir y se valida analizando la tasa de aciertos y errores de nuevas instancias de datos con etiquetas conocidas que permitan realizar la comprobación [9].
- **Aprendizaje no supervisado:** En este caso no hay ningún objetivo a aprender exactamente, dicho en otras palabras y comparando con el aprendizaje supervisado, en este caso los datos no están etiquetados. Trata de descubrir las características subyacentes del conjunto de datos que permita establecer afinidades entre muestras, conocer la propia estructura de los datos o como estos se distribuyen. Esto da lugar a las técnicas de *clustering* y *association*, que respectivamente son el agrupamiento de datos y las reglas de asociación que describen los datos [21,22].

Dada la estructura de los datos y disponiendo de la clase a la cual corresponde cada muestra, se ha decidido abordar el presente trabajo mediante aprendizaje supervisado. Si se indaga en este campo es posible ver que existen numerosos algoritmos, por lo que es importante estudiar sus características para hacer una pequeña selección de aquellos más adecuados para aplicar en la resolución del problema, tal y como se muestra en la tabla 5 la cual es una recopilación de las fortalezas y debilidades de cada algoritmo que se muestran en Lantz 2015 [9].

Se dispone de tres conjuntos de datos. Un conjunto de datos de especiación que contiene 87 muestras, mientras que los otros dos conjuntos de datos corresponden al estudio de adulteración y contienen 104 muestras para el de leche al 5% de adulteración y 96 para el de 10%. Las primeras 48-50 muestras en ambos conjuntos de datos de adulteración de leche son compartidas en ambos conjuntos de datos, por lo que el número de muestras independientes se verá reducido para estos dos conjuntos de datos.

Los conjuntos de datos presentan un par de inconvenientes:

- Se dispone de escasas muestras para aplicar algoritmos de machine learning, por debajo de las 200 muestras en ambos conjuntos de datos. También hay que tener en cuenta que el entrenamiento y validación de los modelos requiere de dividir el conjunto de datos original en dos subconjuntos más pequeños. Este hecho puede condicionar la eficiencia del modelo, derivando en sobreajuste, es decir en poca capacidad de generalización. Para obtener una buena capacidad de generalización, se ha demostrado que con conjuntos de entrenamiento con abundantes datos se obtiene mayor efectividad al usar el modelo con datos no observados previamente, aunque no es siempre posible obtener tal cantidad de datos [23].
- Por otra parte, se dispone de gran cantidad de variables que superan con creces el número de muestras. La llamada maldición de la dimensionalidad que se agrava con la escasez de muestras. Este hecho podría suponer problemas teniendo en cuenta que las variables que no intervienen directamente en la discriminación de clases pueden introducir ruido en el conjunto de datos que afecte a la eficiencia del modelo. Un ejemplo claro es la afectación que tiene las dimensiones sobre las distancias, aumentando de manera exponencial con estas, lo que implica riesgo de alta dispersión que a su vez implica una posible caída en la en la fiabilidad de las predicciones y tendencia al sobreajuste. [24,25] Además, puede ser computacionalmente costoso entrenar un modelo de tales características.

Tabla 5. Listado de algoritmos de aprendizaje supervisado aplicables a problemas de clasificación, la tarea ejecutada, sus fortalezas y debilidades [9].

Modelo	Fortalezas	Debilidades
Nearest Neighbor (kNN)	Simple y efectivo. No hace suposiciones sobre la distribución de los datos. Rápido de entrenar.	Limita el entendimiento de cómo se relacionan clase y variables. Requiere de escoger el valor apropiado de vecinos próximos. La clasificación es lenta. Requiere de procesado adicional cuando se trata de datos nominales o datos faltantes.
Naive Bayes (NB)	Simple, rápido y efectivo. Trabaja bien con datos con ruidos y datos faltantes. Requiere pocos ejemplos para ser entrenado. Puede proporcionar el dato de probabilidad estimada de una predicción.	Se basa en que las variables son de igual importancia e independientes. No es adecuado para conjuntos de datos con muchas variables numéricas. Las probabilidades estimadas son menos fiables que las previstas.
Decision Trees (DT)	Se adapta a la resolución de varios tipos de problema. Proceso de aprendizaje muy automatizado que no presenta problemas con los tipos de variables ni los datos faltantes. Excluye variables no importantes. Adaptable a varios tipos de tamaño de los conjuntos de datos. Fácil interpretación del modelo. Gran eficiencia.	Variables con un gran número de niveles provocan sesgo en el modelo. Tendencias sobreajuste y subajuste. Algunas relaciones pueden presentar problemas al tener tendencia a realizar las separaciones paralelas a los ejes. Pequeños cambios en el conjunto de entrenamiento pueden suponer grandes cambios de la lógica de decisión. Árboles grandes pueden ser difíciles de interpretar y parecer contraintuitivos

Modelo	Fortalezas	Debilidades
Random Forests (RF)	Funciona bien en la mayoría de los problemas. Versátil al poder usarse con datos ruidosos, datos faltantes, con variables continuas o categóricas. Selecciona solo las variables importantes. Puede ser usado con datos con extremado nombre de variables o muestras.	El modelo resultante no es fácilmente interpretable a diferencia de los *decision trees* de los cuales deriva. Resulta algo laborioso ajustar el modelo a los datos.
Classification rule learners (CL)	Genera una regla empírica sencilla de interpretar. Buen rendimiento. Puede servir como referencia para algoritmos más complejos.	Utiliza una sola variable. Demasiado simplista.
Neural Networks (NN)	Versátil: para clasificación y predicción numérica. Puede modelizar patrones mucho más complejos que otros algoritmos. Realiza algunas suposiciones sobre las relaciones subyacentes de los datos.	Computacionalmente de demasiado compleja lo que supone un entrenamiento lento que se acentúa con mayor complejidad de la red. El entrenamiento es muy dado al sobreajuste. Genera modelos difíciles de interpretar.
Support Vector Machines (SVM)	Versátil: para clasificación y predicción numérica. No se ve afectado por datos ruidosos ni tiene tendencia al sobreajuste. De uso más sencillo que las redes neuronales.	La búsqueda del mejor modelo supone probar diversas combinaciones de los hiperparámetros. Lento en entrenamiento cuando se usa conjuntos de datos grandes. Modelos difíciles de interpretar.

Con esta información se decide emplear los algoritmos: neural networks, support vector machines y random forests para llevar a cabo los estudios comentados anteriormente. Se dispone de unos datos con alto número de variables y escaso número de muestras, lo que ha hecho descartar el uso de naive bayes. Esta misma complejidad hace que algoritmos como neural networks y support vector machines sean adecuados para los conjuntos de datos de alta dimensionalidad que se disponen, aunque ello supondrá un entrenamiento más lento debido a su complejidad. Por otra parte, se descarta classification learners y nearest neighbors. Todo y ser sencillos de comprender e implantar es su sencillez la que no los adecua a un conjunto de datos tan complejo. Tanto decision trees como random forests se adaptan bien a conjuntos de datos como los que se van a trabajar (pocas muestras y gran dimensionalidad), sin embargo, todo y su complejidad de interpretación se escoge el uso de random forests al ser una solución que no tiende al sobreajuste ni ofrece tan invariables resultados a los cambios en los conjuntos de entrenamiento.

3.4 Parametrización de los modelos

Se han construido los modelos iniciales en la configuración más básica posible y dejando los parámetros posibles con su valor por defecto. En la tabla 6 puede verse la parametrización realizada para cada algoritmo escogido y para cada estudio. Para la mejora de los modelos se han modificado parámetros y añadido funciones de control de entrenamiento. Las tablas 7, 8 y 9 detallan las mejoras aplicadas en los algoritmos RF, SVM y NN respectivamente y para cada tipo de estudio.

Para el caso particular del algoritmo NN y para el estudio de especiación, como capa final se usa una capa densa con 4 salidas y la función softmax, debido a que esta devuelve valores entre 0 y 1 relacionados con la probabilidad de la entrada a pertenecer a una determinada clase. Es por este motivo que al tratarse de una clasificación multiclase se escoge esta función [26]. Se escoge el optimizador ADAM por tener una buena velocidad de convergencia a la par que una calidad de convergencia correcta y ser adecuado para conjuntos de datos con gran cantidad de parámetros [24]. Por otra parte, para la función de coste se escoge categorical crossentropy, ya que se trabaja con un clasificador multiclase [27]. Para el caso del estudio de adulteración, como capa final se usa una capa densa con 1 salida y la función sigmoid, debido a que esta devuelve valores entre 0 y 1 relacionados con la probabilidad de la entrada a pertenecer o no a la clase principal. Es por este motivo que al tratarse de una clasificación binaria se escoge esta función, al ser también habitual para la resolución de problemas de clasificación binarios [24]. Se mantiene el optimizador ADAM por los mismos motivos anteriormente mencionados. Por otra parte, para la función de coste se escoge binary crossentropy ya que se trabaja con un clasificador binario.

Tabla 6. Parametrización de los modelos iniciales.

	Estudio de especiación	Estudio de adulteración
RF	**-Árboles:** 500. **-Número de variables seleccionadas aleatoriamente como candidatas:** √600. Para los conjuntos de datos en los que se han reducido dimensiones se emplearon √27 y √19 variables.	**-Árboles:** 500. **-Número de variables seleccionadas aleatoriamente como candidatas:** √1600. Para los conjuntos de datos en los que se han reducido dimensiones se emplearon √101 y √100 variables.
SVM	**-Kernel:** Lineal (*vanilladot*).	**-Kernel:** Lineal (*vanilladot*).
NN	**-**Capa de entrada de 600 neuronas. Para los conjuntos de datos después de aplicar SMOTE y SMOTE+PCA se emplearon 27 y 19 neuronas respectivamente. **-**1 capa densa de 10 neuronas con función de activación rectificadora (*ReLu*). **-**Capa final con 4 neuronas con función de activación *softmax*. **-Optimizador:** ADAM. **-Función de coste:** *Categorical crossentropy*. **-**50 épocas. **-**Minilotes de 32 muestras. **-**Subconjunto de validación del 10% del tamaño del conjunto de entrenamiento.	**-**Capa de entrada de 1600 neuronas. Para los conjuntos de datos después de aplicar SMOTE y SMOTE+PCA se emplearon 101 y 100 neuronas respectivamente. **-**1 capa densa de 20 neuronas con función de activación rectificadora (*ReLu*). **-**Capa final con 1 neurona con función de activación *sigmoid*. **-Optimizador:** ADAM. **-Función de coste:** *Binary crossentropy*. **-**50 épocas. **-**Minilotes de 32 muestras. **-**Subconjunto de validación del 10% del tamaño del conjunto de entrenamiento.

Tabla 7. Parametrización de los modelos alternativos para el algoritmo RF.

	Estudio de especiación	Estudio de adulteración
RF alternativa 1	**-Número de árboles:** 500. **-Control de entrenamiento:** Validación cruzada. **-Número de pliegues:** 5. **-Número de variables seleccionadas aleatoriamente como candidatas:** No se especifica. La función automáticamente prueba diversos valores y escoge el que mejores resultados ofrece.	**-Número de árboles:** 500. **-Control de entrenamiento:** Validación cruzada. **-Número de pliegues:** 5. **-Número de variables seleccionadas aleatoriamente como candidatas:** No se especifica. La función automáticamente prueba diversos valores y escoge el que mejores resultados ofrece. Debido al alto coste computacional de esta técnica, se ha desestimado su aplicación en el escenario de trabajo con muestras aumentadas (SMOTE), ya que, debido a la magnitud del conjunto de datos, se requería de largos tiempos de cálculo para la obtención de resultados.
RF alternativa 2	**-Número de árboles:** 1000. **-Control de entrenamiento:** Validación cruzada. **-Número de pliegues:** 5. **-Número de variables seleccionadas aleatoriamente como candidatas:** No se especifica. La función automáticamente prueba diversos valores y escoge el que mejores resultados ofrece.	**-Número de árboles:** 1000. **-Control de entrenamiento:** Validación cruzada. **-Número de pliegues:** 5. **-Número de variables seleccionadas aleatoriamente como candidatas:** No se especifica. La función automáticamente prueba diversos valores y escoge el que mejores resultados ofrece. Debido al alto coste computacional de esta técnica, se ha desestimado su aplicación en el escenario de trabajo con muestras aumentadas (SMOTE), ya que, debido a la magnitud del conjunto de datos, se requería de largos tiempos de cálculo para la obtención de resultados.

Tabla 8. Parametrización de los modelos alternativos para el algoritmo SVM.

	Estudio de especiación	Estudio de adulteración
SVM alternativa 1	**-Kernel:** Lineal (*vanilladot*) **-Control de entrenamiento:** Validación cruzada. **-Número de pliegues:** 5.	**-Kernel:** Lineal (*vanilladot*) **-Control de entrenamiento:** Validación cruzada. **-Número de pliegues:** 5.
SVM alternativa 2	**-Kernel:** Gausiano (*svmRadial*) **-Control de entrenamiento:** Validación cruzada. **-Número de pliegues:** 5.	**-Kernel:** Gausiano (*svmRadial*) **-Control de entrenamiento:** Validación cruzada. **-Número de pliegues:** 5. **-Hiperparámetos:** La propia función prueba distintos valores del hiperparámetro "coste" y escoge el que mejor resultados presente.
SVM alternativa 3	**-Kernel:** Gausiano (*svmRadial*) **-Control de entrenamiento:** Validación cruzada. **-Número de pliegues:** 5. **-Valores sigma:** 0.008, 0.01 y 0.013. **-Valores de coste:** 0.9, 1.1, y 1.35.	**-Kernel:** Polinomial (*svmPoly*) **-Control de entrenamiento:** Validación cruzada. **-Número de pliegues:** 5. **-Hiperparámetos:** La propia función prueba distintos valores de los hiperparámetros "coste" y "escala" y escoge la combinación que mejor resultados presente.

Tabla 9. Parametrización de los modelos alternativos para el algoritmo NN.

	Estudio de especiación	Estudio de adulteración
NN alternativa 1	-Capa de entrada de 600 neuronas. Para los conjuntos de datos después de aplicar SMOTE y SMOTE+PCA se emplearon 27 y 19 neuronas respectivamente. -2 capas densas de 20 y 10 neuronas respectivamente con función de activación rectificadora (*ReLu*). -Capa final con 4 neuronas con función de activación *softmax*. **-Optimizador:** ADAM. **-Función de coste:** *Categorical crossentropy*. -50 épocas. -Minilotes de 32 muestras. -Subconjunto de validación del 10% del tamaño del conjunto de entrenamiento.	-Capa de entrada de 1600 neuronas. Para los conjuntos de datos después de aplicar SMOTE y SMOTE+PCA se emplearon 101 y 100 neuronas respectivamente. -2 capas densas de 30 y 15 neuronas respectivamente con función de activación rectificadora (*ReLu*). -Capa final con 1 neurona con función de activación *sigmoid.* **-Optimizador:** ADAM. **-Función de coste:** *Binary crossentropy*. -50 épocas. -Minilotes de 32 muestras. -Subconjunto de validación del 10% del tamaño del conjunto de entrenamiento.

	Estudio de especiación	Estudio de adulteración
NN alternativa 2	-Capa de entrada de 600 neuronas. Para los conjuntos de datos después de aplicar SMOTE y SMOTE+PCA se emplearon 27 y 19 neuronas respectivamente. -1 capa densa de 5 neuronas con función de activación rectificadora (*ReLu*). -Capa final con 4 neuronas con función de activación *softmax*. **-Optimizador:** ADAM. **-Función de coste:** *Categorical crossentropy*. -50 épocas. -Minilotes de 32 muestras. -Subconjunto de validación del 10% del tamaño del conjunto de entrenamiento.	-Capa de entrada de 1600 neuronas. Para los conjuntos de datos después de aplicar SMOTE y SMOTE+PCA se emplearon 101 y 100 neuronas respectivamente. -1 capa densa de 55 neuronas con función de activación rectificadora (*ReLu*). -Capa final con 1 neurona con función de activación *sigmoid.* **-Optimizador:** ADAM. **-Función de coste:** *Binary crossentropy*. -50 épocas. -Minilotes de 32 muestras. -Subconjunto de validación del 10% del tamaño del conjunto de entrenamiento.

3.5 Técnicas de aumento de datos

Sobre los problemas mencionados con anterioridad, hay diferentes formas de abordarlos. En primer lugar, sobre la problemática de la escasez de muestras:

- **Data augmentation:** Es una forma de generar nuevos datos de forma artificial. Las referencias que existen hacen alusión habitualmente a la aplicación destinada a aumento de datos en texto, imágenes y otros medios audiovisuales, realizando pequeñas transformaciones, añadiendo ruido o eliminando parte de los datos. Aunque esta misma técnica es difícil de extrapolar a datos tabulados como los que se disponen, existen otras opciones basadas en la aplicación de k nearest neighbors, como SMOTE, o basadas en la aplicación de redes neuronales con *autoencoders* (*VAEs*) o redes generativas adversarias (*GANs*) para la generación de nuevos datos manteniendo las clases originales [23,28].
- **Transfer learning:** Trata de aprovechar la experiencia de aprendizaje de un modelo previamente entrenado para otra tarea de clasificación distinta a la que se quiere llevar a cabo [29]. Para este caso sería necesario encontrar un modelo de acceso libre al tipo de datos se quiere trabajar para poder adaptarlo.
- **N-shot learning:** Se trata de entrenar un modelo a partir de una cantidad de muestras que puede variar de una sola muestra a unas pocas. El concepto de este entrenamiento no se fundamenta en la probabilidad de que una nueva instancia de datos pertenezca a una clase u otra, se fundamenta en el grado de similitud (distancia) comparando el vector de características generado por una red neuronal del elemento de referencia

con el vector de características de la muestra que se quiere clasificar. Es de habitual aplicación en redes neuronales, siendo popular por reportar buenos resultados las redes neuronales siamesas [30].

Para lidiar con este problema se escoge la técnica de data augmentation para generar más muestras artificialmente. De las opciones disponibles que hay para aplicar data augmentation se escoge SMOTE, por ser de sencilla aplicación sin tener que ajustar la cantidad de parámetros que implica una red neuronal, más aún para la cantidad de variables que los conjuntos de datos disponen. Resulta difícil encontrar un modelo que se adapte los requerimientos específicos del estudio. Además, hay que tener en cuenta que uno de los objetivos del estudio es encontrar el modelo con la técnica que ofrezca mejores resultados, por lo que habría sido necesario encontrar un modelo pre-entrenado para cada uno de los algoritmos escogidos.

SMOTE es una técnica basada en el algoritmo k nearest neigbours para el aumento de datos. En su aplicación se estableció como 5 el número de vecinos próximos para realizar la síntesis de cada nueva muestra. Los aumentos se han realizado por cada clase, aumentando iterativamente cada clase en una tasa de 25 veces la clase minoritaria sobre la clase mayoritaria para el estudio de especiación y 20 veces para el de adulteración. Se emplearon las funciones *SMOTE* y *ADAS* del paquete *smotefamily* para su implementación. La técnica se ha aplicado por separado a los subconjuntos de entrenamiento y de validación surgidos de la división del conjunto de datos original.

3.6 Técnicas de reducción de la dimensionalidad

Por otra parte, el problema de la alta dimensionalidad de los datos puede ser resuelto analizando las variables y seleccionando aquellas que evidentemente tengan importancia en el conjunto de datos. Aquí hay una lista no exhaustiva de las opciones más habituales de reducción:

- **Principal component análisis (PCA):** Reduce la dimensionalidad de los datos identificando el hiperplano más próximo a los datos y proyectando sobre este las observaciones. Posteriormente se escogen las componentes principales, es decir los ejes en los que se proyectan los datos, siendo el primero aquel que preserve la máxima varianza, el segundo el siguiente a preservar más varianza y que a su vez sea perpendicular al primero y así sucesivamente [24]. De esta manera se crean nuevas variables, combinación lineal de las originales pero incorrelacionadas entre ellas [31], permitiendo así eliminar los problemas de multicolinealidad y en consecuencia reduciendo la dimensión [32].
- **Multidimensional scaling:** Se trata de un caso similar con un caso similar al anterior. Trata de encontrar la aproximación lineal que encaje los datos

en una dimensión menor. A diferencia de las componentes principales, en el este caso se trata de conservar las distancias entre dos puntos [33].

- **Best subset selection:** Trata de construir un submodelo por separado combinando todos los pares de variables posibles y posteriormente construir el modelo seleccionando el mejor de cada submodelo, lo que permite seleccionar el mejor modelo de entre todas las combinaciones de variables posibles. Todo y su simplicidad conceptual, este método es difícil de implementar por su alto coste computacional, especialmente en conjuntos de datos con gran número de variables. Hay que tener en cuenta que todas las combinaciones posibles con 2 variables (*p)* se calcularían como 2^p, dando lugar así a un sinfín de combinaciones [34].
- **Stepwise selection (forward y backward):** Trata de construir iterativamente un modelo de regresión. Para el caso de backward selection se construye un modelo con todas las variables en el que en cada paso se eliminará la variable con un p-valor más alto y a la vez superior al nivel de confianza previamente fijado, hasta llegar al punto en que no hay más variables que superen el nivel de confianza. De forma contraria, forward selection parte de un modelo sin variables en que en cada paso se añade la variable con el *p-valor* más bajo que quede por debajo del nivel de confianza, y así sucesivamente hasta que no queden variables que puedan ser añadidas [35]. También existen soluciones híbridas de las dos variantes en que en cada paso se añade y se elimina una variable conforme a los criterios anteriormente descritos [34]. Tiene la ventaja de ser computacionalmente menos exigente que otros algoritmos, aunque su resultado no siempre deriva en el mejor modelo [34], y más aún cuando la eliminación de variables implica aumentos en el p-valor de las variables restantes [35]. Por otra parte, y en concreto la variante forward, es adecuado para ser usada cuando el número de variables es grande y supera al de las muestras disponibles [34].
- **Random forests:** Si bien es un algoritmo de machine learning como ya hemos comentado con anterioridad para resolver problemas de clasificación, también puede usarse para reducir la dimensionalidad. En este caso, se aprovecha la métrica *Gini* que mide la calidad de la división durante el entrenamiento. Variables con un bajo valor de *Gini* indicarán que su importancia es menor al resto a la hora de establecer divisiones y por lo tanto tienen poca influencia en el modelo [36].

Para resolver el problema de la dimensionalidad se ha optado por escoger la técnica de principal component analisys. Los motivos son principalmente el alto tiempo de computación que otras técnicas, como la eliminación/adición recursiva de variables e incluso random forests, tienen para trabajar en conjuntos de variables de alta dimensionalidad como los que en este trabajo se emplean. Por otra parte, PCA, ofrece la ventaja de no eliminar variables y conservarlas combinadas en las nuevas componentes, priorizando aquellas que más

variabilidad ofrecen. De esta forma no se omite información que podría ser relevante para algunos casos de discriminación, resultando en la consecución de un modelo más fiable que el que pudieran conseguir el resto de las técnicas. Hay que comentar que se escoge PCA frente al escalado multidimensional, dado que para el problema que se quiere resolver, más que las distancias, es necesario conservar la varianza de los datos para discriminar, dado que no se está resolviendo un problema de similitudes.

Para la implementación de la reducción de dimensiones mediante PCA se utilizó la función *prcomp* del paquete *stats* de R. Esta función realiza el cálculo de las componentes principales mediante la descomposición en valores singulares de la matriz de datos [37]. La selección del número óptimo de componentes a emplear se ha llevado a cabo mediante el criterio de *Kaiser,* donde se escogen aquellas componentes cuya varianza explicada sea mayor que 1 [38]. La técnica se ha aplicado por separado a los subconjuntos de entrenamiento y de validación surgidos de la división del conjunto de datos original. En el escenario de trabajo donde se aplica la combinación de aumento de muestras y reducción de la dimensionalidad, se ha aplicado PCA por separado sobre los conjuntos de entrenamiento y validación previamente aumentados con SMOTE.

3.7 Métricas para la evaluación del rendimiento de los modelos

Para la evaluación del rendimiento se ha optado por las métricas que a continuación se muestran, las cuales han sido obtenidas mediante la función *confusionMatrix* que integra el paquete *caret*:

- **Exactitud (*accuracy*):** Indica la proporción de predicciones correctas del modelo frente al total de predicciones ejecutadas, es decir la cantidad de verdaderos positivos (TP) y verdaderos negativos (TN) frente a la suma de estos dos más la cantidad de falsos positivos (FP) y falsos negativos (FN). En otras palabras, indica la tasa de éxito del modelo: [9]

$$Exactitud = \frac{TP + TN}{TP + TN + FN + FP}$$

- **Estadístico *Kappa*:** Es similar a la exactitud, pero ajusta el valor de esta teniendo en cuenta la posibilidad de realizar una predicción correcta por azar: [9]

$$kappa = \frac{\Pr(a) + \Pr(e)}{1 - \Pr(e)}$$

  Pr(a) hace referencia a la tasa de acierto obtenida con el modelo, mientras que *Pr(e)* hace referencia a la proporción de aciertos esperado si se escogieran los resultados por azar [9]. Su valor oscila de 0 a 1. En Lantz, 2015 [9] se propone la siguiente interpretación:
  - Concordancia pobre: por debajo de 0.20.

  - Concordancia media: entre 0.20 y hasta 0.40.
  - Concordancia moderada: entre 0.40 y hasta 0.60.
  - Concordancia buena: entre 0.60 y hasta 0.80.
  - Muy buena concordancia: entre 0.80 y hasta 1.00.
- **Sensibilidad (*sensitivity*):** Tasa de verdaderos positivos. Mide la proporción de verdaderos positivos sobre el total de positivos (tanto verdaderos como falsos negativos): [9]

$$sensibilidad = \frac{\mathrm{TP}}{TP + FN}$$

  En el caso del estudio de adulteración se usan las muestras adulteradas como clase positiva, mientras que en el de especiación se calcula esta métrica por cada una de las clases, asumiendo cada vez una distinta la clase positiva.
- **Especificidad (*specificity*):** Tasa de verdaderos negativos. Mide la proporción de verdaderos negativos sobre el total de negativos (tanto verdaderos como falsos positivos):

$$especificidad = \frac{\mathrm{TN}}{TN + FP}$$

  En el caso del estudio de adulteración se usan las muestras adulteradas como clase positiva, mientras que en el de especiación se calcula esta métrica por cada una de las clases, asumiendo cada vez una distinta la clase positiva.

3.8 Flujo de trabajo para el diseño y la mejora de modelos

El flujo de trabajo para el diseño de los diferentes modelos se muestra a continuación y está basado en el esquema de Lantz 2015 [9] que se ha mencionado en el apartado *1.4 Enfoque y método seguido*:

1. Se han planteado cuatro escenarios de trabajo distintos:
   a. Escenario con el conjunto de datos sin transformar (escenario original).
   b. Escenario con datos aumentados (mediante SMOTE).
   c. Escenario con dimensiones reducidas (mediante PCA).
   d. Escenario combinado, con datos aumentados y dimensiones reducidas (aplicando SMOTE y seguidamente PCA al conjunto de datos aumentado).
2. Dentro de cada escenario de trabajo se han construido diferentes modelos iniciales empleando los algoritmos propuestos. A estos modelos se les ha aplicado una configuración inicial sencilla, modificando los parámetros indispensables para su puesta en marcha.
3. A continuación, se ha evaluado el rendimiento de cada modelo inicial diseñado con cada algoritmo propuesto mediante las métricas de exactitud, Kappa, sensibilidad y especificidad.

4. A partir de los resultados de rendimiento se lleva a cabo la mejora de los modelos. Se modifican los parámetros de cada modelo inicial, añadiendo control de entrenamiento mediante validación cruzada en algunas de las variaciones.
5. Se realizan varias iteraciones de los puntos 3 y 4 para disponer de diversidad de modelos.
6. Terminada la construcción, evaluación y mejora de modelos en uno de los escenarios de trabajo, se repite el proceso entero en el siguiente escenario de trabajo.

En la figura 5 se puede apreciar un esquema del procedimiento seguido para la construcción de los modelos.

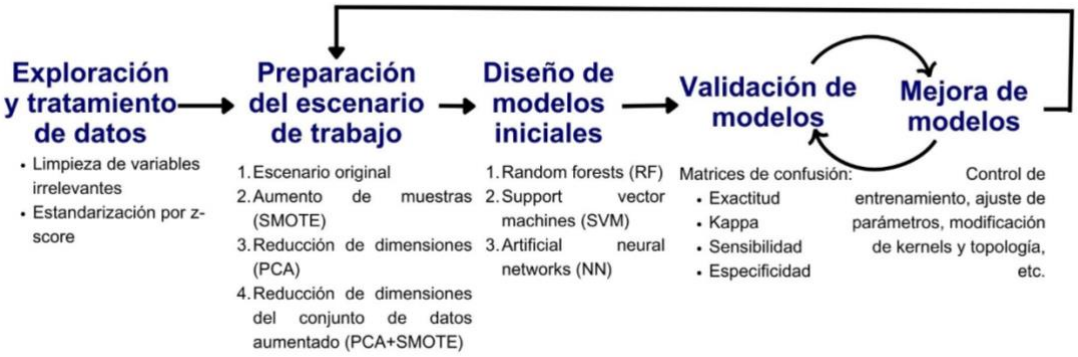

Figura 5. Flujo seguido para el diseño y la mejora de los diferentes modelos construidos.

3.9 Criterios para la selección del mejor modelo

Para la determinación del mejor modelo a emplear se ha establecido como primer criterio de selección la obtención de las mejores métricas, poniendo especialmente foco en la exactitud. En caso de obtener varios modelos con resultados iguales o muy similares, el siguiente criterio para escoger el mejor modelo está basado en la complejidad de este, teniendo preferencia siempre por aquel modelo más simple que facilite la comprensión de su trabajo. Además, en este último criterio también se valora la necesidad de ajuste de hiperparámetros, así como el coste computacional del modelo en su uso predictivo.

Por otra parte, el criterio de selección del mejor escenario de trabajo está basado en la obtención de mejores métricas de rendimiento.

3.10 Aplicación web

Dentro del objetivo general del presente trabajo y como objetivo específico, estaba el desarrollo de una aplicación que permitiera a usuarios externos poder ejecutar una predicción de muestras no previamente vistas de una forma rápida y sencilla. Es por ese motivo que el desarrollo de la aplicación se ha hecho mediante web, de tal forma que el usuario puede acceder desde cualquier

dispositivo y sin tener que interactuar directamente con el código del modelo. Hay que comentar que para el desarrollo de dicha aplicación se han empleado los trabajos de Santiago G. Berruga [39], Damaris Alarcón [40] y Chanin Nantasenamat [41] como modelos de referencia.

3.10.1 Interfaz de usuario de la aplicación

Por lo que respecta a la estructura de la página web, es decir la interfaz de usuario, se ha optado por emplear dos paneles. El primer panel, el lateral, tiene la función de recogida de datos, y es donde se encuentran las diversas opciones y espacios para esta finalidad. El segundo panel, el central, tiene la función de comunicación con el usuario y es donde se proyectan los resultados y otra información como instrucciones o información sobre la aplicación. Además, se incluye también una pestaña donde se pueden encontrar conjuntos de datos para probar la aplicación y las indicaciones de cómo usar cada uno. Para este último panel se ha optado hacer una distribución por pestañas para comodidad del usuario, dejando la pestaña de resultados abierta por defecto al iniciar la aplicación.

Toda la interfaz de usuario se ha implementado en la sección *UI* del código, empleando las funciones *sidebarPanel* para el diseño del panel lateral de captación de datos y *tabsetPanel* para el diseño del panel central donde se proyecta información y resultados para el usuario. Parte de la interfaz de usuario fue implementada en el lenguaje HTML, en concreto aquellas que requerían de texto, como el pie de página, la pestaña de instrucciones y el apartado de información sobre la aplicación.

3.10.2 Servidor de la aplicación

La sección *server* contiene el código correspondiente a la parte ejecutiva de la aplicación, la cual usa los datos entrados en la interfaz de usuario. Cuenta con las siguientes funcionalidades:

- **Elección del tipo de predicción a realizar:** Es decir, permite seleccionar al usuario si llevar a cabo una predicción de especie o bien una predicción de adulteración de sus muestras, e indicar si estas son individuales o bien múltiples.
- **Elección del modo de subida de datos:** En el caso de la predicción individual, la aplicación permite escoger entre subir un archivo *.csv* o bien pegar en un cuadro de texto la muestra a analizar. Para el análisis múltiple solo existe la opción de carga de archivo.
- **Elección del formato de los datos:** Permite al usuario indicar las diversas opciones del formato en el que se introducen los datos: si los datos contienen encabezados, si la muestras están identificadas y en qué columna se encuentra su identificador, el carácter separador de

columnas (coma o punto y coma) y el carácter separador de decimales (punto o coma). En el caso del identificador, si se indica que los datos no llevan identificador, la aplicación asigna uno automáticamente a cada muestra de los datos introducidos.

- **Validaciones de datos:** El propio formulario de recolección de datos integra validaciones de datos indicando al usuario cuando los datos introducidos no cumplen con los requisitos. Por otra parte, se realizan diferentes validaciones a nivel interno: que el conjunto de datos tenga todas las variables necesarias para el tipo de predicción que se quiere llevar a cabo, que el número de columna de identificador indicado tenga valores lógicos y que los conjuntos de datos no tengan valores nulos, perdidos o no numéricos. Esta parte se ha implementado mediante la función *InputValidator* del paquete *shinyvalidate*. La forma de comunicar los errores en las validaciones de datos es mediante alertas en el propio formulario y ventanas emergentes indicando el tipo de error cuando se intenta ejecutar la predicción con valores no adecuados. Esta funcionalidad se ha implementado mediante el paquete *shinyalert* con su función *shinyalert.*
- **Predicciones:** La aplicación tiene un panel dedicado exclusivamente a la muestra de resultados. Estos se muestran tabulados con su identificador original o bien con el asignado por la aplicación. La tabla de resultados es interactiva, incluye un paginador con la opción de escoger la cantidad de resultados por página para los casos en que la cantidad de muestras analizadas sea extensa. También incorpora un buscador que permite la búsqueda de valores o muestras.

3.11 Software empleado

3.11.1 Tratamiento de datos y desarrollo de los modelos de aprendizaje automático

Se ha decidido programar los algoritmos del presente trabajo en el lenguaje de programación R principalmente, por la familiaridad con este y la disponibilidad de librerías, mostradas en la tabla 10, que dispone para machine learning.

El código ha sido escrito y ejecutado en la versión más reciente de R a la fecha de esta memoria, la versión 4.3.0, tanto cuando se ha trabajado en entorno Windows como en entrono macOS. En el caso de la versión empleada para entornos macOS se usó la versión 4.3.0-x86_64 mediante el traductor dinámico binario Rosetta que incorpora este sistema operativo. Esta versión está dedicada a los ordenadores Mac que usan chips Intel y se ha decidido emplear para evitar problemas de compatibilidad con algunos paquetes.

Para la escritura y ejecución del código se ha empleado el IDE (entorno integrado de desarrollo por sus siglas en inglés) RStudio versión 2023.03.1+446 de la compañía Posit PBC, tanto cuando se ha trabajado en entorno Windows como en entrono MacOs X.

Tabla 10. Librerías de R a emplear para el desarrollo del trabajo y su función.

Librería	Aplicación
dplyr	Para el uso de operadores especiales y simplificar la escritura del código
smotefamily	Para la aplicación de SMOTE para data augmentation.
keras	Framework para aplicar de forma sencilla modelos de artificial neural networks.
kernlab	Para la aplicación de support vector machines.
randomForest	Para la aplicación de random forests.
caret	Preprocesamiento de datos y división de los conjuntos de datos en los subconjuntos de entrenamiento y validación. También para la obtención de las matrices de confusión y métricas de rendimiento. Construcción de modelos de SVM y RF con validación cruzada y ajuste de hiperparámetros.
kableExtra	Presentación de tablas de resultados para los informes parciales y el presente informe.
shiny	Para el desarrollo de la aplicación web.
rsconnect	Para la conexión entre el IDE y el servidor web de Shiny Apps donde se aloja la aplicación.
shinythemes	Para establecer con facilidad un tema preconfigurado en la interfaz de usuario de la aplicación web.
shinyvalidate	Para la validación de datos de entrada de la aplicación web y avisos en el propio formulario.
shinyjs	Para la ejecución de funcionalidades dentro la aplicación web.
shinyalert	Para la implantación de alertas de error dentro de la aplicación web.
MVN	Aplicación del test de Shapiro-Wilk para comprobar la normalidad de los datos.

3.11.2 Desarrollo de la aplicación web

Para el desarrollo de la aplicación web se ha decidido emplear la herramienta Shiny Apps de la compañía Posit PBC. Esta herramienta se integra dentro del IDE de RStudio, con lo cual permite emplear R para el desarrollo de aplicaciones y facilita su implementación al no tener que cambiar de lenguaje para ello. Adicionalmente, la compañía permite el alojamiento de las aplicaciones creadas en sus servidores de forma gratuita y asociados a una cuenta de usuario. Este hecho también facilita la publicación de la aplicación y la hace

accesible a todas las personas que la quieran emplear a través de un enlace web.

3.12 Hardware empleado

La realización del trabajo se ha llevado a cabo mediante ordenador personal. Ha sido necesario emplear dos ordenadores con sistemas operativos distintos, ya que algunas librerías de R presentaban problemas de compatibilidad en los modelos más recientes de MacBook que emplean chips Apple Silicon:

- **MacBook Air M1 2020:** Chip Apple M1 8 núcleos (4 x 3.2GHz & 4 x 2GHz), 16gb RAM, SO macOS Ventura 13.3.1.
- **HP Laptop 15-dw1023ns:** Chip Intel® Core™ i5-10210U (frecuencia base de 1,6 GHz, hasta 4,2 GHz con tecnología Intel® Turbo Boost, 6 MB de caché L3, 4 núcleos), 8gb RAM, Sistema de 64 bits, Sistema operativo Windows 11 Pro 22H2 22621.1555.

4 Resultados

A continuación, se describen los resultados obtenidos por cada tipo de estudio. Por cada modelo se evalúa su rendimiento general y en cada escenario de trabajo.

4.1 Estudio de especiación

Se pueden apreciar los resultados en las tablas 11 y 12. La tabla 11 muestra los resultados de exactitud y de kappa mientras que en la tabla 12 se proyectan los resultados de sensibilidad y especificidad.

Tabla 11. Resultados de exactitud (acc) y kappa (kp) para el estudio de especiación.

	Original		SMOTE		PCA		SMOTE + PCA	
	acc	**kp**	**acc**	**kp**	**acc**	**kp**	**acc**	**kp**
RF	89.3	85.7	96.7	95.6	92.9	90.4	97.3	96.3
RF + CV	89.3	85.7	97.3	96.3	92.9	90.4	97.4	96.5
RF + CV y 1000 árboles	89.3	85.7	97.6	96.7	96.5	95.2	97.4	96.5
SVM lineal	100	100	100	100	100	100	100	100
SVM lineal + CV	100	100	100	100	100	100	100	100
SVM radial + CV	96.4	95.2	100	100	92.9	90.5	99.3	99.1
SVM radial + CV y hiper.	89.3	85.7	85.6	80.8	96.4	95.2	100	100
NN 10	100	100	100	100	50.0	33.3	100	100
NN 20+10	96.4	95.2	100	100	64.3	52.4	99.9	99.8
NN 5	89.3	85.7	98.8	98.4	39.3	19.1	98.2	97.6

Tabla 12. Resultados de sensibilidad (sen) y especificidad (esp) para el estudio de especiación. Clase positiva: Camello (Cam), vaca (Vac), cabra (Cab) y oveja (Ov).

			RF	RF + CV	RF + CV y 1000 árboles	SVM lineal	SVM lineal + CV	SVM radial + CV	SVM radial + CV y hiper.	NN 10	NN 20+10	NN 5
Original	**sen**	**Cam**	100	100	100	100	100	100	87.5	100	100	71.4
		Vac	85.7	85.7	85.7	100	100	100	100	100	85.7	85.7
		Cab	85.7	85.7	85.7	100	100	87.5	83.3	100	100	100
		Ov	87.5	87.5	87.5	100	100	100	87.5	100	100	100
	esp	**Cam**	95.5	95.5	95.5	100	100	100	100	100	95.2	100
		Vac	95.2	95.2	95.2	100	100	95.5	95.5	100	100	90.5
		Cab	95.2	95.2	95.2	100	100	100	90.9	100	100	95.2
		Ov	100	100	100	100	100	100	100	100	100	100
SMOTE	**sen**	**Cam**	100	100	100	100	100	100	63.9	100	100	95.1
		Vac	88.4	100	100	100	100	100	98.7	100	100	100
		Cab	100	92.6	93.2	100	100	100	100	100	100	100
		Ov	100	96.8	97.3	100	100	100	100	100	100	100
	esp	**Cam**	95.8	100	100	100	100	100	100	100	100	100
		Vac	100	97.5	97.7	100	100	100	94.8	100	100	98.4
		Cab	100	98.9	99.1	100	100	100	90.1	100	100	100
		Ov	100	100	100	100	100	100	97.3	100	100	100
PCA	**sen**	**Cam**	100	100	100	100	100	87.5	87.5	42.9	71.4	0.0
		Vac	87.5	87.5	87.5	100	100	100	100	71.4	42.9	71.4
		Cab	100	100	100	100	100	85.7	100	0.0	71.4	14.3
		Ov	87.5	87.5	100	100	100	100	100	85.7	71.4	71.4
	esp	**Cam**	95.5	95.5	95.5	100	100	100	100	81.0	100	100
		Vac	100	100	100	100	100	95.5	100	71.4	81.0	71.4
		Cab	95.5	95.5	100	100	100	95.2	95.5	95.2	76.2	76.2

			RF	RF + CV	RF + CV y 1000 árboles	SVM lineal	SVM lineal + CV	SVM radial + CV	SVM radial + CV y hiper.	NN 10	NN 20+10	NN 5
		Ov	100	100	100	100	100	100	100	85.7	95.2	71.4
SMOTE + PCA	**sen**	**Cam**	100	100	100	100	100	97.3	100	100	100	100
		Vac	97.3	97.9	97.9	100	100	100	100	100	99.5	92.9
		Cab	100	100	100	100	100	100	100	100	100	100
		Ov	92.4	92.4	92.4	100	100	100	100	100	100	100
	esp	**Cam**	100	100	100	100	100	100	100	100	100	100
		Vac	100	100	100	100	100	100	100	100	100	100
		Cab	96.5	96.6	96.6	100	100	99.1	100	100	99.8	100
		Ov	100	100	100	100	100	100	100	100	100	97.6

La gran mayoría de resultados ha conseguido rebasar el 90% de exactitud como eficiencia objetivo marcada. Este hecho se comenta a continuación y puede apreciarse en la tabla 11.

Poniendo foco en SVM, se puede apreciar que el empleo del kernel lineal, ya sea con validación cruzada o sin ella consigue lograr un 100% de exactitud en todos los escenarios de trabajo, y así lo hace también el valor de kappa. A la hora de aplicar un kernel radial, incluso probando diferentes combinaciones de sus hiperparámetros, se obtiene una eficiencia superior al 90%, pero notablemente menor con respecto al kernel lineal. Dado que el kernel lineal ha proporcionado una eficiencia alta, hay que fijarse en los modelos con kernel radial para ver cómo afecta el escenario de trabajo, aumentando en los casos en los que se aplica SMOTE solo y junto a PCA. Sin embargo, cuando se aplica PCA a solas se sigue consiguiendo un resultado por encima del 90% de exactitud, pero disminuyendo esta respecto al escenario original.

Para el algoritmo NN se han obtenido eficiencias y valores de kappa del 100% con una capa densa de 10 neuronas, a excepción del escenario donde solo se aplica PCA que se queda en un 50% de precisión y un 33.3% de kappa. Al aumentar las capas del modelo se ha seguido rebasando el 90% de eficiencia, pero quedando por debajo del modelo con una sola capa y 10 neuronas, a excepción del escenario donde se aplica SMOTE que sí que consigue llegar al 100% en exactitud y kappa. La disminución a 5 neuronas y una sola rebaja los valores de kappa respecto a los otros modelos en pequeña medida, pero haciendo que, en ocasiones, como en el escenario original o el escenario donde se aplica PCA, la exactitud quede por debajo del 90%. En cuanto al escenario de trabajo, todos siguen una misma tendencia, donde aquellos que emplean SMOTE cosechan los resultados más altos. Sin embargo, el escenario de trabajo donde únicamente se aplica PCA cosecha los resultados más bajos, siendo 64,3% la mayor exactitud alcanzada.

Los modelos elaborados con el algoritmo RF son los que resultados más bajos han alcanzado por lo general, pero mayoritariamente también han rebasado el 90% de eficiencia fijado, a excepción de los que se han llevado a cabo en el escenario original. En rasgos generales, los valores de exactitud y kappa han sido muy pareados en todas las combinaciones. Alcanzan la exactitud máxima en 97.5%, en el modelo con 1000 árboles y validación cruzada, y así lo hace también el valor de kappa alcanzando un máximo en 96.7%. Por lo que respecta al escenario de trabajo, en los casos que se emplea SMOTE ya sea solo o junto a PCA, se obtienen cotas de exactitud y kappa. Sin embargo, es con el escenario original con el que se cosechan los resultados más bajos, no alcanzando el 90% de eficiencia en ningún caso.

Poniendo foco en la tabla 12. Se pueden apreciar resultados altos en cuanto a sensibilidad y especificidad se refiere, alcanzando en muchas

ocasiones el 100% para las dos métricas. Es evidente que los mayores resultados de especificidad son logrados por el algoritmo SVM, en especial para aquellos modelos con kernel lineal.

4.2 Estudio de adulteración

Se pueden apreciar los resultados en las tablas 13 y 14. La tabla 13 muestra los resultados de exactitud y de kappa mientras que en la tabla 14 se proyectan los resultados de sensibilidad y especificidad.

Tabla 13. Resultados de exactitud (acc) y kappa (kp) para el estudio de adulteración.

	Original		SMOTE		PCA		SMOTE + PCA	
	acc	**kp**	**acc**	**kp**	**acc**	**kp**	**acc**	**kp**
RF	80.0	47.1	78.7	57.6	70.0	8.3	60.9	22.2
RF + CV	90.0	76.6	-	-	78.0	40.6	58.2	16.9
RF + CV y 1000 árboles	88.0	71.5	-	-	78.0	40.6	56.6	13.6
SVM lineal	90.0	78.1	95.4	90.7	92.0	82.8	96.3	92.5
SVM lineal + CV	90.0	78.1	95.4	90.7	92.0	82.8	96.3	92.5
SVM radial + CV	68.0	0.0	74.0	48.0	68.0	0.0	75.6	51.1
SVM polinomial + CV	90.0	78.1	95.4	90.7	68.0	0.0	96.3	92.5
NN 20	76.0	46.6	83.7	67.4	76.0	40.9	81.0	62.0
NN 30+15	82.0	56.5	88	76.1	76.0	46.6	90.6	81.1
NN 55	88.0	72.4	89.8	79.7	78.0	46.8	82.3	64.7

Tabla 14. Resultados de sensibilidad (sen) y especificidad (esp) para el estudio de adulteración. La clase positiva son las muestras adulteradas.

	Original		SMOTE		PCA		SMOTE + PCA	
	sen	**esp**	**sen**	**esp**	**sen**	**esp**	**sen**	**esp**
RF	78.6	87.5	70.9	95.8	69.4	100	56.0	97.5
RF + CV	91.4	86.7	-	-	76.7	85.7	54.4	91.3

	Original		SMOTE		PCA		SMOTE + PCA	
	sen	**esp**	**sen**	**esp**	**sen**	**esp**	**sen**	**esp**
RF + CV y 1000 árboles	88.0	85.7	-	-	76.7	85.7	53.5	88.3
SVM lineal	96.8	79.0	97.2	93.7	100	80.0	100	93.1
SVM lineal + CV	96.8	79.0	97.2	93.7	100	80.0	100	93.1
SVM radial + CV	68.0	0.0	66.6	92.3	68.0	0.0	100	67.3
SVM polinomial + CV	96.8	79.0	97.2	93.7	68.0	0.0	100	93.1
NN 20	79.4	68.8	95.5	72.0	88.2	50.0	86.5	75.6
NN 30+15	91.2	62.5	98.5	77.7	79.4	68.8	85.2	95.8
NN 55	91.2	81.3	97.6	82.1	88.2	56.3	97.6	67.3

El algoritmo SVM ha conseguido alcanzar el 90% de exactitud en la mayoría de sus aplicaciones, e incluso rebasarlo. El kernel radial es el único que no ha alcanzado el umbral de exactitud, con valores del 69% hasta el 75.6%, mientras que las configuraciones de kernel lineal simple y con validación cruzada y la configuración con kernel polinomial, lo ha conseguido de forma bastante igualada entre ellas con un arco que va del 68% al 96.3%. En cuanto al escenario de trabajo, se puede apreciar que tanto el aumento de muestras como la reducción de dimensiones consiguen mejores resultados, siendo el escenario combinando SMOTE y PCA el que mejor resultados ha obtenido. En cuanto el valor de kappa respecta, ha ido en aumento a medida que se aplicaban transformaciones en el escenario original hasta el 92.5% del escenario combinado, siendo éste el valor más alto obtenido.

Para el algoritmo NN, se puede ver que los resultados mejoran a medida que se han ido aumentando las capas o el número de capas por neurona. El modelo con 20 capas es el que eficiencias más bajas cosecha de entre el 76% y el 83.7%. En contraposición, aumentar el número de capas ha aumentado la eficiencia, obteniendo una horquilla de exactitud del 76% al 90%. El aumento del número neuronas también aumenta la eficiencia, como es el caso del modelo con 55 neuronas donde se ha obtenido una horquilla de exactitud de entre 78% y 89.9%. Poniendo foco en el escenario de trabajo, empleando SMOTE y un modelo de una sola capa con 55 neuronas se cosechan resultados próximos al 90%, aunque es en el escenario combinado de SMOTE y PCA donde se alcanza

el pico de exactitud en 90.6% para el modelo NN 30+15. Los valores de kappa obtenidos consiguen llegar en un par de ocasiones al 79.7% y al 81.1% para los modelos NN 55 y NN 30+10 respectivamente.

En cuanto al algoritmo RF, empleando una configuración de 500 árboles y empleando la validación cruzada, se ha alcanzado el 90% de exactitud en el escenario de trabajo original, su máximo valor. No se puede valorar el rendimiento de este algoritmo en el escenario de trabajo SMOTE, ya que como se ha dicho con anterioridad, los largos tiempos de computación han impedido su aplicación. En el escenario con PCA y en el escenario combinado apenas se consigue un 60% de eficiencia en el mejor de los casos. Los valores de kappa cosechados son bajos, siendo del 76.6% en el mejor de los casos.

Por otra parte, es importante fijar la vista en los resultados de la tabla 2 donde aparecen los valores de sensibilidad y especificidad obtenidos y con los cuales se muestra la capacidad del modelo para predecir una y otra clase. El patrón es similar a lo comentado con la exactitud y kappa, se pueden apreciar que los resultados de sensibilidad y especificidad cosechados por SVM son los más altos, especialmente para el escenario combinando SMOTE y PCA donde alcanza el 100% para la sensibilidad y el 93.7% para sensibilidad.

4.3 Aplicación web

A continuación, se muestran los resultados más relevantes del desarrollo de la aplicación web “Detector de fraude en leche” como uno de los productos finales del trabajo. Para la parte del servidor de la aplicación se ha implementado un modelo de SVM lineal tanto para la función de predicción de especie como la de adulteración. En los apartados *5. Discusión* y *6. Conclusiones* se justifica la elección de dicho modelo frente al resto de opciones. Puede accederse a la aplicación a través del siguiente enlace:

https://miguelangellg.shinyapps.io/detectorFraudeLeche/

La figura 6 muestra la interfaz gráfica de la aplicación abierta en su apartado de instrucciones. A la izquierda se puede ver el panel de capitación de datos, donde el usuario selecciona el tipo de predicción a llevar a cabo, cómo llevarla a cabo, un espacio también para la subida de archivos si aplica, y unos selectores para indicar el formato en que se encuentran sus datos. Hay que comentar que el panel de captación de datos es dinámico, mostrando nuevas opciones en función de las opciones que se vayan seleccionado.

Figura 6. Interfaz de usuario de la aplicación web.

La figura 7 muestra un ejemplo de alerta cuando en el formulario se ha introducido un dato no válido. El propio cuadro de texto alerta del error. La figura 8 muestra un ejemplo de ventana emergente derivada de la carga de un conjunto no válido para la predicción que se quiere llevar a cabo.

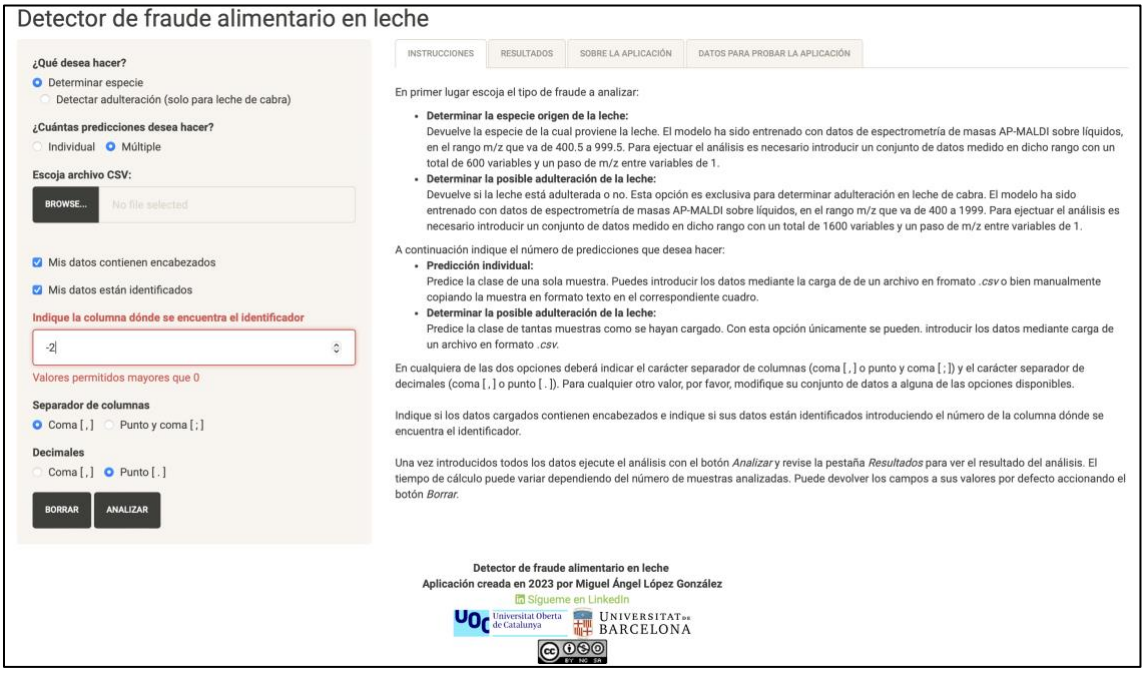

Figura 7. Alerta de datos erróneos en el formulario de la aplicación web.

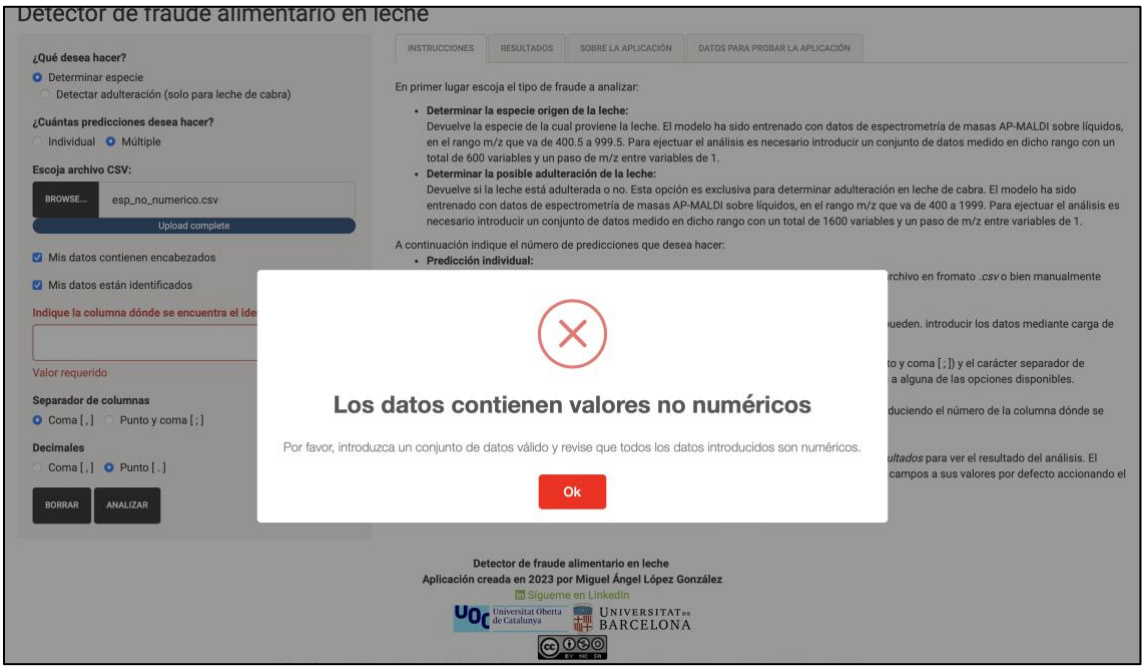

Figura 8. Ventana emergente por conjunto de datos no válido en la aplicación web.

La figura 9 muestra un ejemplo de predicción satisfactoria, donde se puede ver una tabla con paginador y buscador que muestra los resultados obtenidos.

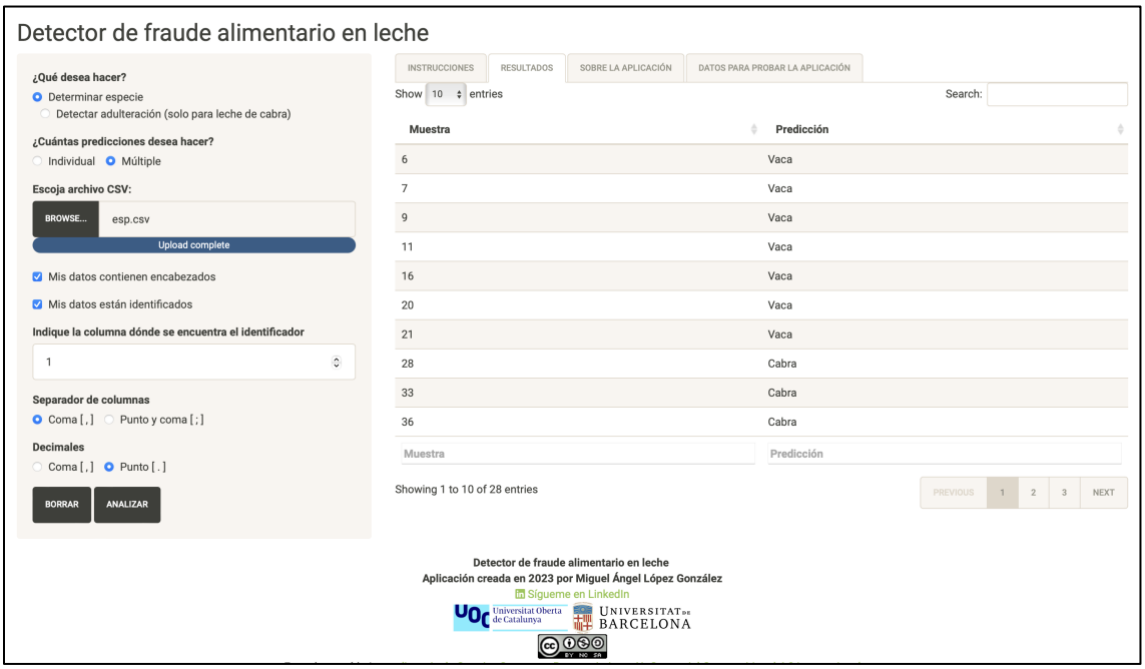

Figura 9. Resultados para predicción múltiple en la aplicación web.

En la figura 10 se puede apreciar como la aplicación lanza una predicción individual mediante carga manual.

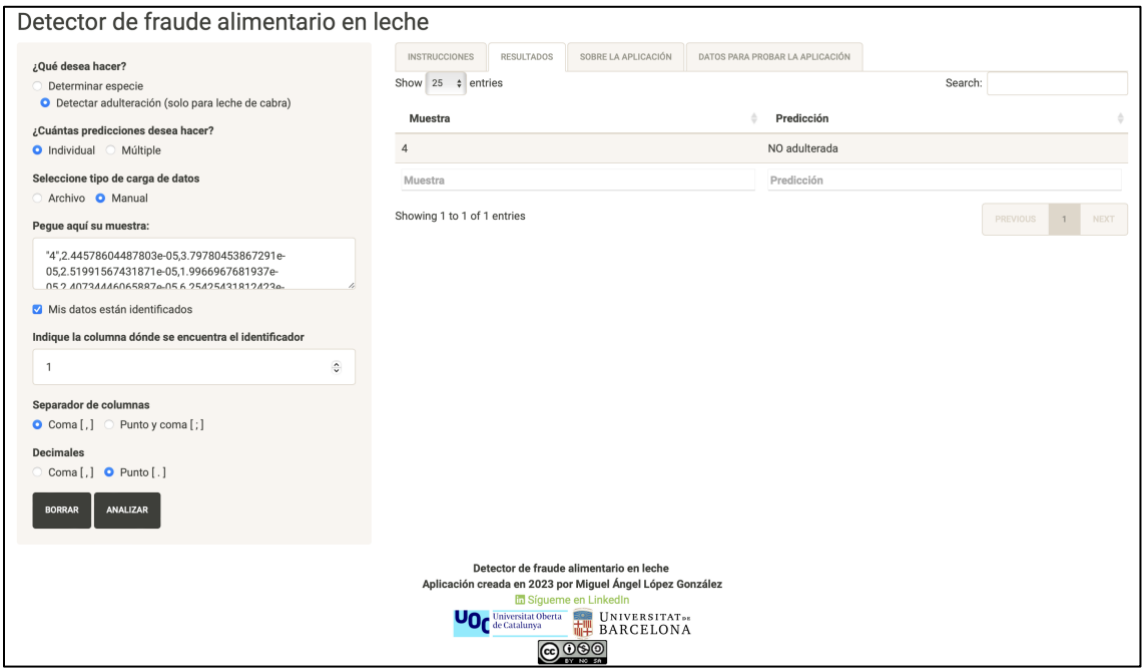

Figura 10. Resultados para predicción individual en la aplicación web.

5 Discusión

5.1 Estudio de especiación

Dados los resultados mostrados en el anterior apartado, salta a la vista que el algoritmo SVM es el que mejor rendimiento obtiene, ya que ha logrado alcanzar unas métricas del 100% para diferentes variaciones de los modelos, pero también en los diferentes escenarios de trabajo. A continuación, el algoritmo NN logra unos resultados tan buenos como SVM, aunque no todas las variaciones de los modelos han tenido tanto éxito ni tampoco han actuado bien en todos los escenarios. Por ejemplo, en el escenario aplicando PCA, donde tan solo logra alcanzar un 39.2% y 50% de exactitud con una kappa del 33.3% y 29.05% respectivamente. Los peores resultados de todos los modelos. Por lo general, los resultados de NN son buenos, superando el 90% en la gran mayoría de variaciones practicadas. También aporta buenos resultados el algoritmo RF, aunque en el escenario original no ha sido capaz de alcanzar el 90% de eficiencia. Las sucesivas aplicaciones de diferentes escenarios de trabajo han conseguido mejorar su eficiencia, pero en ninguna variación se ha logrado aportar métricas del 100% como lo han hecho el resto de los algoritmos.

Poniendo atención al algoritmo SVM, se debe destacar que han sido los kernels lineales, tanto en su aplicación sin control de entrenamiento como con la aplicación de validación cruzada, los que mejor resultado han obtenido, con tasas del 100% en todas las métricas observadas en las tablas 11 y 12.

Por lo que respecta al escenario de trabajo, es evidente que con el algoritmo SVM ninguno de ellos resulta un problema. Hay que fijarse por lo tanto en el resto de las modelos que no ha conseguido tan buena métrica. Por ejemplo, en RF se aprecia que la aplicación de SMOTE y SMOTE junto con PCA mejora las métricas a máximos para dicho algoritmo, especialmente en este último escenario de trabajo. De manera similar ocurre con todos los modelos de NN y los modelos SVM con kernel radial, donde se aprecia ese salto cualitativo a la hora de usar un escenario de trabajo donde se aumentan datos, pero viéndose perjudicados en el escenario en el que se aplica únicamente PCA donde se obtienen las métricas más bajas.

Comparando los resultados obtenidos con los obtenidos por Piras et al. [7], en rasgos generales y con la excepción de algunos modelos, se puede apreciar que se ha logrado igualar la eficiencia obtenida dado que se ha alcanzado el 100% de exactitud en la mayoría de modelos, resultado que este alcanza en su estudio empleando análisis discriminante lineal.

5.2 Estudio de adulteración

A la vista de los resultados, es evidente que el mejor algoritmo es SVM, seguido de NN y RF. Es con los modelos construidos con SVM con los cuales se alcanzan las cotas de eficiencia y se supera el umbral del 90% fijado en los objetivos. Por otra parte, NN le precede, aunque en una sola ocasión alcanza una exactitud del 90%, los resultados en general son buenos, manteniéndose en torno al 80-90% la exactitud. Por otra parte, RF queda lejos en cuanto a buena eficiencia comparado con los otros modelos. Solo en una ocasión alcanza un 90% pero en general los valores oscilan por debajo del 80% en la exactitud. Además, RF supone un problema al tratar datos de alta dimensionalidad por el tiempo de computación necesario para el cálculo de los modelos, por lo que automáticamente le resta preferencia a la hora de ser escogido para la implantación del modelo.

Dentro de SVM, es necesario poner foco en los kernels lineales. Son los que con creces han obtenido mejores resultados, tanto en exactitud como en kappa, llegando hasta el 96.3% en el primer parámetro. En este grupo también entraría el kernel polinomial, aunque con carencias en el escenario donde se trabaja con PCA.

Para finalizar y poniendo foco exclusivamente en los diferentes escenarios de trabajo, se aprecia que en los casos que se han aumentado las muestras con SMOTE las métricas mejoran, especialmente en el escenario combinando SMOTE y PCA. Las exactitudes crecen en prácticamente todos los algoritmos, a excepción de RF. Así lo hace también el valor de kappa. Este hecho indica que un entrenamiento con más muestras mejora la capacidad de predicción. También se consigue un modelo más fiable, en tanto que los valores de kappa crecen junto con el aumento de muestras. Este hecho es debido a que empleando SMOTE se balancean las clases igualando las probabilidades de obtención de una clase u otra por azar.

Observando los resultados de Piras et al. [7] se puede apreciar que para dicho estudio no emplea la métrica de exactitud, por lo que se compara únicamente en sensibilidad y especificidad. Este consigue una sensibilidad y especificidad del 92.5% y 94.5% para la leche adulterada al 5% y del 99.2% y 99.1% para la leche adulterada al 10%. Para el presente trabajo se han unificado los dos grados de adulteración para construir un único modelo predictivo. Aun así y pese a la diversidad de los resultados conseguidos en los diferentes algoritmos, se puede apreciar que la tendencia es similar a la obtenida por Piras et al., aproximándose o superando el 92% de sensibilidad e incluso llegando al 100%, en los mejores modelos. Especialmente para SVM lineal y las diferentes configuraciones de neural networks en los escenarios en los que se ha empleado SMOTE solo o junto con PCA. Por otra parte, los resultados de especificidad, todo y ser bastante buenos y cercanos a los de Piras et al., no se aprecian tan

igualados ni con tanta frecuencia, consiguiendo su máximo en 93.7% en modelos SVM con kernels lineales y polinomial y 95.8% para el modelo NN 30+15, en comparación al 94.5% y 99.1% que consigue Piras et al, usando análisis discriminante lineal.

En resumen, SVM en sus configuraciones lineal, lineal con validación cruzada y polinomial y bajo el escenario de trabajo donde se ha aplicado SMOTE y PCA al conjunto de datos original, es el algoritmo cuyos modelos obtienen los mejores resultados.

6 Conclusiones

6.1 Estudio de especiación

Como conclusión para el estudio de especiación, se afirma que el algoritmo SVM junto con NN, en concreto los modelos con kernel lineal simple, kernel lineal con validación cruzada y las redes neuronales con 10 neuronas y 2 capas de 20 y 10 neuronas, son las mejores opciones para la clasificación de muestras para especiación. Estas obtienen resultados del 100% en exactitud, kappa, sensibilidad y especificidad. A la vista salta que estos modelos obtienen exactamente el mismo resultado de eficiencia en cuanto a exactitud se refiere, por lo que aplicando el criterio de simplicidad se decide escoger como mejor modelo el de SVM con kernel lineal sin aplicar controles de entrenamiento ni ajuste de hiperparámetros.

Por otra parte, por lo que respecta al escenario de trabajo, y aunque el modelo de SVM lineal simple ha trabajado bien en todos los escenarios propuestos, se decide como escenario óptimo aquel en el cual se aplica SMOTE y PCA al conjunto de datos original. Así se decide dado que el aumento de muestras y reducción de dimensiones hace que sea un modelo más robusto frente a datos no vistos previamente por el modelo. A la vista está también la mejora de los resultados en dicho escenario, marcando valores del 100% o muy cercanos en cada algoritmo empleado.

6.2 Estudio de adulteración

Se puede concluir que el mejor modelo para crear un modelo predictivo de adulteración en leche de cabra es el creado con el algoritmo SVM. Es el que, con holgura, permite alcanzar el 90% de eficiencia fijado en los objetivos. Si se aprecian los resultados de las tablas 13 y 14, puede observarse que entre las dos configuraciones de kernel lineal (con y sin validación cruzada) y la configuración con kernel polinomial se obtienen los exactamente los mismos resultados de exactitud. En este caso, y según el criterio de simplicidad anteriormente mencionado, se escoge el modelo más simple como el mejor, que en el caso para el estudio de adulteración es SVM con kernel lineal sin aplicar ningún control de entrenamiento ni ajuste de hiperparámetros.

Por otra parte, a la vista de los resultados, es evidente que el mejor escenario con el que trabajar es el de datos aumentados y dimensiones reducidas, es decir, aquel donde al conjunto de datos original se le ha aplicado SMOTE y PCA. Es el escenario donde mayor valor de exactitud se alcanza y, además, de mejorar y alcanzar cotas en los valores de kappa, sensibilidad y especificidad.

6.3 Conclusiones generales

Con este estudio se ha podido demostrar:

- La capacidad de algoritmos alternativos a los propuestos por Piras et at. [7], como son SVM, RF y NN, para la detección de especie de muestras de leche y para la detección adulteración a concentraciones de un 0 y 10% en leche de cabra, obteniendo un alto grado de exactitud y fiabilidad.
- El uso de conjuntos de datos de alta dimensionalidad y con bajo número de muestras para el desarrollo de modelos predictivos influye de forma negativa en la capacidad de predicción y la fiabilidad de estos modelos. Por otra parte, se ha podido demostrar a su vez que se puede conseguir mejorar la eficiencia y fiabilidad de los modelos aplicando técnicas sencillas de aumentos de datos como SMOTE y técnicas de reducción de la dimensionalidad como PCA.

Por lo que respecta a los objetivos del trabajo, han podido cumplirse todos. Se ha podido reducir la dimensión del conjunto de datos seleccionado, así como incrementar el número de muestras artificialmente. Este hecho ha permitido comparar los modelos en diferentes escenarios de trabajo, con la pequeña excepción en el estudio de adulteración con el algoritmo NN, con el cual no ha sido posible la construcción de modelos bajo el escenario de muestras aumentadas mediante SMOTE por el alto coste computacional que implicaba. Se ha podido escoger el mejor escenario de trabajo. También se ha podido determinar cuál de los modelos es el que mejor se adapta a lo datos con los cuales se trabaja y al problema a resolver, cumpliendo además con el requisito de eficiencia en exactitud del 90% en el modelo seleccionado. Con todo se ha podido implementar cada uno de los algoritmos escogidos, para los dos estudios realizados, en una aplicación web para que usuarios externos puedan realizar predicciones sobre sus propias muestras de manera rápida y sencilla. En resumen, con este proyecto se ha logrado crear modelos de machine learning para detectar fraude en leche (mediante detección de especie y adulteración de leche) e implementarlos en una herramienta interactiva.

En cuanto al seguimiento de la planificación, esta ha podido seguirse en su mayoría, aunque ha sido necesario introducir las siguientes variaciones en la metodología para garantizar el éxito del trabajo:

- Inicialmente se planteó la reducción de la dimensionalidad mediante el uso del método stepwise selection. Se desestimó su uso después de que varios intentos fracasaran por los largos tiempos de computación y malos resultados que se obtenían. Se optó por PCA cuyos tiempos de computación eran notablemente menores. Además, no hay eliminación de

variables, sino que estas se conservan en nuevas componentes combinación de las variables originales [31], por lo que no hay una pérdida de información que puede ser relevante en comparación stepwise selection.

- Se decidió valorar el rendimiento de los algoritmos en diferentes escenarios de trabajo, y no directamente tras reducir dimensiones y aumentar datos tal y como se había planteado inicialmente. Este hecho responde a la necesidad de comparar el rendimiento de los modelos en diferentes escenarios y así poder justificar la elección llevada a cabo. Este hecho supuso añadir un objetivo más al trabajo.
- En un principio no se había planificado el desarrollo de la aplicación web, pero decidió llevarse a cabo al disponer de tiempo para ello. De esta forma se le ha logrado dar más sentido al presente trabajo. Eso ha supuesto añadir un objetivo más al trabajo.
- En general, los objetivos planificados en un inicio se han mantenido, añadiendo los anteriormente mencionados. Algunos de ellos se han modificado para ser más claros en la finalidad que se persigue.
- El orden de la ejecución de las tareas en ocasiones no ha sido el que en un inicio se planteó, pero este hecho no ha supuesto ningún inconveniente en el desarrollo de las tareas.

Se ha comentado con anterioridad que el alcance del trabajo tiene repercusión sobre los ODS 2 y 3 correspondientes a hambre y seguridad alimentaria y salud respectivamente. Se espera a que el presente trabajo impacte de forma positiva en estos ODS por el hecho de haber podido crear modelos de detección de especie y adulteración en leche con una buena fiabilidad, y más aún por el hecho de haber podido crear una aplicación web implementando dichos modelos en un entorno de fácil y rápida ejecución por personal no necesariamente técnico. Este hecho facilita, en primer lugar, a laboratorios y empresas dedicadas al sector alimentario, cierta independencia para analizar las muestras, sin tener que depender de terceros para la obtención de un veredicto ni disponer de personal especializado para ello. Además, establece un criterio común para la criba de muestras fraudulentas de las que no lo son, por lo que todos los laboratorios y empresas de un mismo grupo podrían trabajar de la misma forma. Si bien, como se verá en las líneas de futuro, el presente trabajo admite mejoras para hacer aún más accesible, económica y de fácil ejecución la detección del fraude en leche, con este trabajo se han sentado las bases para la consecución de modelos más precisos y versátiles. También ejemplifica el machine learning como técnica válida para la detección de fraude, hecho que marca el camino para la extrapolación de los métodos aquí usados para ser aplicados en la detección de otros tipos de fraude, en otros alimentos y para detectar otros riesgos que puedan comprometer la seguridad de los consumidores y la confianza en los operadores alimentarios a su vez. En resumen, este trabajo es un aporte positivo al concepto de seguridad alimentaria

en su vertiente más sanitaria, dado que facilita el acceso de manera uniforme a la mitigación del riesgo de fraude y a la vez, a la fabricación de alimentos seguros para la población general.

6.4 Líneas de futuro

Aunque se ha logrado cumplir con los objetivos del proyecto y en general se han obtenido buenos resultados en los modelos construidos empleando variaciones simples, a continuación, se muestran los futuros pasos a seguir para una mejora de los resultados y de los productos obtenidos:

- Sería conveniente repetir ambos estudios empleando conjuntos de datos con más muestras para el entrenamiento. En su defecto, sería interesante probar una técnica de aumento de datos alternativa, ya que el procedimiento de SMOTE, al basarse en una interpolación mediante el método k nearest neighbors, proporciona nuevas muestras linealmente dependientes de las originales. El uso de redes neuronales con *autoencoders* (*VAEs*) [23,28] o redes generativas adversarias (*GANs*) [23] dedicadas para datos tabulados, pueden ser buenas alternativas para sintetizar muestras independientes de las originales. Con estas propuestas se podrían mejorar los resultados de la fase de entrenamiento.
- Sería necesario disponer de muestras dedicadas exclusivamente a la fase de validación del modelo y en la medida de lo posible, que estas fueran analizadas en laboratorios o en secuencias distintas a las de los datos de entrenamiento. Disponer de muestras independientes del conjunto de entrenamiento para la fase de validación puede ayudar a mejorar el modelo frente a datos no vistos, ya que previene la introducción de sesgos en este. En este sentido, también sería adecuado repetir el estudio con muestras que no fueran réplicas, la cual cosa repercutiría positivamente en la reducción del sobreajuste de los modelos y a mejorar la predicción en datos no vistos.
- Sería necesario ampliar el estudio de adulteración con leches de otras especies. También sería adecuado ampliar este estudio con muestras adulteradas a un amplio rango de grados de dilución, lo que haría posible crear un modelo que fuera capaz de detectar la adulteración y el grado en el que esta se da.
- La aplicación admite más funcionalidades, como, por ejemplo, la representación gráfica de las muestras en las componentes principales de los datos de entrenamiento como soporte visual a los resultados o la generación de informes, tal y como ha llevado a Santiago G. Berruga en su trabajo [39]. También sería una mejora incorporar la conservación de datos asociados a cuentas de usuario, o la aceptación de distintos formatos de datos. Sería interesante también trabajar conjunto a los

operadores alimentarios y laboratorios que trabajen lácteos para introducir mejoras en la aplicación que cubran sus necesidades reales.

- Sería de especial interés repetir el estudio, pero empleando técnicas analíticas más rápidas y accesibles al personal no técnico y a cualquier empresa, como por ejemplo la espectroscopía de infrarrojo cercano.

7 Glosario

Adulteración alimentaria: Tipo de fraude alimentario basado en la sustitución parcial de la composición de un alimento con el objetivo de ganar beneficio económico.

Aprendizaje automático (*machine learning*): Campo de la computación, y en concreto de la inteligencia artificial, que trata de programar ordenadores para que aprendan a partir de datos [24] y en base a experiencias pasadas [42]. Por ejemplo: desarrollando técnicas de observación de datos y estudiando su estructura subyacente para el propio aprendizaje, entendido como la capacidad de predicción.

Artificial neural networks (NN): Algoritmo de aprendizaje automático inspirado en el mecanismo de comunicación de las neuronas biológicas. Se basa en la interconexión de un conjunto de unidades elementales (neuronas), la cual aplica una función de transformación determinada (sigmoide, ReLu, lineal, etc.) [42]. Estas redes se organizan en estructuras de capas [42], entrelazándose a través de las neuronas, donde el valor de salida de una supone el valor de entrada de otra y así hasta crear una red como producto final de la interconexión de todas las capas y neuronas.

ADAM: *Adaptive moment estimation*. Algoritmo de optimización o entrenamiento para redes neuronales. [42]

Binary crossentropy: Función de coste en redes neuronales artificiales para clasificación binaria.

Categorical crossentropy: Función de coste en redes neuronales artificiales para clasificación multiclase.

Época: Iteración en el entrenamiento de una red neuronal.

Especificidad: Métrica que mide la proporción de verdaderos negativos sobre el total de negativos en un modelo predictivo. [9]

Espectrometría de masas: Técnica de análisis químico que se basa en la ionización de moléculas orgánicas en fase gaseosa y en su almacenamiento en un campo eléctrico que permite identificarlas según su relación masa/carga (m/z). [43]

Exactitud (accuracy): Conocida como tasa de éxito, indica la proporción de predicciones correctas frente al total de predicciones llevadas a cabo. [21]

FN: Falsos negativos.

FP: Falsos positivos.

Fraude alimentario: Engaño premeditado en alimentos, por sustitución, adición o sustracción de ingredientes no declarados, manipulación de su etiquetado o información al consumidor, o sobre el cual se han realizado declaraciones falsas, con el objetivo final de obtener un beneficio económico.

HTML: *HyperText Markup Language*. Se trata de un lenguaje dedicado al diseño de páginas web.

IDE: *Integrated Development Envitoment*. Entorno de desarrollo integrado. Sowftware dedicado a la escritura de código de programación y su ejecución.

IQR: *Interquartile Range*. Rango intercuartílico. Medida estadística que indica el valor en una distribución entre el tercer y el primer cuartil.

k nearest neighbors (kNN): Algoritmo de machine learning que clasifica nuevas instancias de datos en función de los *k* vecinos más próximos que estas tienen [9]. Es decir, al representarlos en el mismo espacio que los datos con los que se ha entrenado el modelo, la nueva instancia de datos se clasifica según la clase mayoritaria de las *k* muestras más próximas.

Kappa: Estadístico para valorar la tasa de éxito de un modelo predictivo que ajusta el valor de la exactitud teniendo en cuenta la probabilidad que hay en el conjunto de datos de realizar una predicción correcta por azar. [9]

Kernel: Tipo de transformación del espacio de multidimensional de características en el algoritmo support vector machines.

m/z: Relación masa/carga de los iones detectados en espectrometría de masas.

ODS: Objetivos de desarrollo sostenible.

Principal component analisys (PCA): Técnica de reducción de la dimensionalidad que se basa en la proyección de los datos en nuevos ejes perpendiculares entre sí y que preservan la máxima varianza posible (llamados componentes) y que son combinación lineal de las variables originales, pero incorrelacionados entre ellos.

R: Lenguaje de programación dedicado al cálculo estadístico.

Random forests (RF): Algoritmo de machine learning basado en el ensamblaje de múltiples árboles de decisión. [9]

Sensibilidad: Métrica que mide la proporción de verdaderos positivos sobre el total de positivos en un modelo predictivo. [9]

Sigmoid: Función de activación sigmoidal en redes neuronales artificiales y que es de uso habitual para clasificación binaria. Proporciona un resultado entre 0 y 1 para tantas salidas como disponga la última capa de la red neuronal, el cual se corresponde a la probabilidad de pertenecer una entrada de datos a una determinada clase [26].

SMOTE: Técnica de aumento de datos tabulados basada en el algoritmo de k nearest neighbors.

Softmax: Función de activación en redes neuronales artificiales de uso habitual en problemas de clasificación multiclase. Proporciona un resultado entre 0 y 1 correspondiente a la probabilidad de pertenecer una determinada entrada de datos a una determinada clase [26].

Stepwise selection: Método de selección de variables mediante adición o sustracción recursiva de estas.

Support vector machines (SVM): Algoritmo de machine learning que se basa en la búsqueda de uno o varios hiperplanos que separen los datos de forma homogénea en el espacio multidimensional. [9]

TN: Verdaderos negativos.

TP: Verdaderos positivos.

k fold cross validation (validación cruzada, CV): Técnica para la evaluación de modelos predictivos basada en la partición aleatoria del conjunto de datos de validación en un total de *k* pliegues [42] y el posterior cálculo del promedio de los resultados obtenidos, es decir, promedio de los valores de la métrica empleada para la evaluación, como por ejemplo: exactitud, error, etc.

Valores atípicos (*outliers*): Valores que se alejan en gran medida de la mayoría de datos del conjunto [9] (en estimadores como media o mediana) y es susceptible de ser erróneo.

Z-score: Técnica de estandarización de datos que centra la media del conjunto de datos a 0 y la desviación estándar a 1. Tiene la propiedad de no comprimir los datos, dado a que no tiene un mínimo ni un máximo predefinidos. [9]

8 Bibliografía

[1] Food and Agriculture Organization of the United Nation. Food fraud – Intention, detection and management. Food safety technical toolkit for Asia and the Pacific. 2021.
[2] Choudhary A, Gupta N, Hameed F, et al. An overview of food adulteration: Concept, sources, impact, challenges and detection. Int J Chem Stud. 2020;8:2564–2573.
[3] Momtaz M, Bubli SY, Khan MS. Mechanisms and Health Aspects of Food Adulteration: A Comprehensive Review. Foods. 2023;12:199.
[4] European Comission. Monthly Food Fraud Summary Reports [Internet]. Knowl. Cent. Food Fraud Qual. 2023 [cited 2023 Mar 12]. Available from: https://knowledge4policy.ec.europa.eu/food-fraud-quality/monthly-food-fraud-summary-reports_en#year2022.
[5] Food and Agriculture Organization of the United Nations, editor. La Seguridad Alimentaria: información para la toma de decisiones. Guía práctica. 2011.
[6] World Health Organization, editor. FOOD SAFETY: What you should know. 2015.
[7] Piras C, Hale OJ, Reynolds CK, et al. Speciation and milk adulteration analysis by rapid ambient liquid MALDI mass spectrometry profiling using machine learning. Sci Rep. 2021;11:3305.
[8] Piras C, Cramer R. Speciation and adulteration analysis of milk by liquid AP-MALDI MS profiling [Internet]. University of Reading; 2020. Available from: https://researchdata.reading.ac.uk/232/.
[9] Lantz B. Machine learning with R: discover how to build machine learning algorithms, prepare data, and dig deep into data prediction techniques with R. Second edition. Birmingham Mumbai: Packt Publishing; 2015.
[10] Holland JK, Kemsley EK, Wilson RH. Use of Fourier transform infrared spectroscopy and partial least squares regression for the detection of adulteration of strawberry purées. J Sci Food Agric. 1998;76:263–269.
[11] Cui C, Xu Y, Jin G, et al. Machine learning applications for identify the geographical origin, variety and processing of black tea using 1H NMR chemical fingerprinting. Food Control. 2023;148:109686.
[12] Momeny M, Neshat AA, Jahanbakhshi A, et al. Grading and fraud detection of saffron via learning-to-augment incorporated Inception-v4 CNN. Food Control. 2023;147:109554.
[13] Li M, Aheto JH, Rashed MMA, et al. Tracing models for checking beef adulterated with pig blood by Fourier transform near-infrared paired with linear and nonlinear chemometrics. Food Sci Technol. 2023;43:e104622.
[14] Yakar Y, Karadağ K. IDENTIFYING OLIVE OIL FRAUD AND ADULTERATION USING MACHINE LEARNING ALGORITHMS. Quím Nova [Internet]. 2022 [cited 2023 Jun 5]; Available from: http://quimicanova.sbq.org.br/audiencia_pdf.asp?aid2=9497&nomeArquivo=AR2022-0116.pdf.
[15] Lima JS, Ribeiro DCSZ, Neto HA, et al. A machine learning proposal method to detect milk tainted with cheese whey. J Dairy Sci. 2022;105:9496–

9508.
[16] Tian H, Wu D, Chen B, et al. Rapid identification and quantification of vegetable oil adulteration in raw milk using a flash gas chromatography electronic nose combined with machine learning. Food Control. 2023;150:109758.
[17] Natarajan S, Ponnusamy V. Classification of Organic and Conventional Vegetables Using Machine Learning: A Case Study of Brinjal, Chili and Tomato. Foods. 2023;12:1168.
[18] Calle JLP, Ferreiro-González M, Ruiz-Rodríguez A, et al. Detection of Adulterations in Fruit Juices Using Machine Learning Methods over FT-IR Spectroscopic Data. Agronomy. 2022;12:683.
[19] Huang W, Guo L, Kou W, et al. Identification of adulterated milk powder based on convolutional neural network and laser-induced breakdown spectroscopy. Microchem J. 2022;176:107190.
[20] Renesh Bedre. 8 methods to find outliers in R (with examples) [Internet]. Data Sci. Blog. 2022 [cited 2023 Apr 23]. Available from: https://www.reneshbedre.com/blog/find-outliers.html.
[21] Lantz B. Machine learning with R: expert techniques for predictive modeling. Packt publishing ltd; 2019.
[22] Dinov ID. Data Science and Predictive Analytics: Biomedical and Health Applications using R [Internet]. Cham: Springer International Publishing; 2018 [cited 2023 Mar 9]. Available from: https://link.springer.com/10.1007/978-3-319-72347-1.
[23] Mumuni A, Mumuni F. Data augmentation: A comprehensive survey of modern approaches. Array. 2022;16:100258.
[24] Géron A, Pineda González B. Aprende Machine Learning con Scikit-Learn, Keras y TensorFlow. Segunda. Madrid: Anaya Multimedia; 2020.
[25] Nicolás Arrioja Landa Cosio. La maldición de la dimensionalidad [Internet]. Medium. 2021 [cited 2023 Mar 19]. Available from: https://medium.com/@nicolasarrioja/la-maldici%C3%B3n-de-la-dimensionalidad-f7a6248cf9a.
[26] Sharma S, Sharma S, Athaiya A. Activaction fucntions in neural networks. Int J Eng Appl Sci Technol. 2020;04:310–316.
[27] TF, Keras, Losses, Categorical Crossentropy [Internet]. 2023 [cited 2023 Apr 24]. Available from: https://www.tensorflow.org/api_docs/python/tf/keras/losses/CategoricalCrossentropy.
[28] Mahendiran A, Subramaniam V. Data Augmentation Techniques for Tabular Data [Internet]. Mphasis Corporation; 2022 [cited 2023 Mar 20]. Available from: https://www.mphasis.com/content/dam/mphasis-com/global/en/downloads/whitepaper/Mphasis_Data-Augmentation-for-Tabular-Data_Whitepaper.pdf.
[29] Yang L, Hanneke S, Carbonell J. A theory of transfer learning with applications to active learning. Mach Learn. 2013;90:161–189.
[30] Chica J, Salamea C. Uso de técnicas basadas en one-shot learning para la identificación del locutor. Proces Leng Nat. 2020;101–108.
[31] Everitt B. An R and S-PLUS companion to multivariate analysis. London: Springer; 2005.
[32] Wei-Meng Lee. Using Principal Component Analysis (PCA) for Machine Learning [Internet]. Data Sci. 2022 [cited 2023 Mar 16]. Available from: https://towardsdatascience.com/using-principal-component-analysis-pca-for-

machine-learning-b6e803f5bf1e.
[33] Marsland S. Machine Learning: An Algorithmic Perspective, Second Edition. 2nd ed. Chapman & Hall/CRC; 2014.
[34] James G, Witten D, Hastie T, et al. An Introduction to Statistical Learning [Internet]. New York, NY: Springer New York; 2013 [cited 2023 Mar 16]. Available from: http://link.springer.com/10.1007/978-1-4614-7138-7.
[35] Faraway JJ. Linear models with R. Second edition. University of Bath, United Kingdom: Taylor and Francis; 2015.
[36] Rukshan Pramoditha. Random forests — An ensemble of decision trees [Internet]. Data Sci. 2020 [cited 2020 Oct 16]. Available from: https://towardsdatascience.com/random-forests-an-ensemble-of-decision-trees-37a003084c6c.
[37] prcomp: Principal Components Analysis [Internet]. RDocumentation. [cited 2023 Jun 7]. Available from: https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/prcomp.
[38] Cuadras CM. Nuevos métodos de análisis multivariante. CMC editions; 2014.
[39] Berruga SG. Diagnóstico de enfermedad hepática mediante técnicas de aprendizaje automático y su implementación en una aplicación web.
[40] Damaris Alarcón Vallejo. Desarrollo de una aplicación web para la predicción de la salud lumbar, aplicando técnicas de aprendizaje automático sobre las características biomecánicas de pacientes ortopédicos. [Trabajo de final de máster]. Universitat Oberta de Catalunya y Universitat de Barcelona; 2021.
[41] Chanin Nantasenamat. R shiny free code camp [Internet]. rshiny_freecodecamp. 2021 [cited 2023 May 20]. Available from: https://github.com/dataprofessor/rshiny_freecodecamp.
[42] Bosch Rué A, Casas-Roma J, Lozano Bagén T. Deep learning: principios y fundamentos. Primera edición digital. Barcelona: Editorial UOC; 2020.
[43] Louis Brown, John Herbert Beynon. Mass spectromety [Internet]. Encycl. Br. Encyclopædia Britannica, Inc; [cited 2023 Jun 13]. Available from: https://www.britannica.com/science/mass-spectrometry/Negative-ions.