SISTEMAS DE RAZONAMIENTO BASADO EN CASOS PARA APLICACIONES MÉDICAS CON ETAPAS DE ADAPTACIÓN Y RECUPERACIÓN MEJORADAS

DAVID RAMIRO BASTIDAS TORRES
CAMILO ANDRÉS PIÑEROS RODRÍGUEZ

UNIVERSIDAD DE NARIÑO
FACULTAD DE INGENIERÍA
INGENIERÍA ELECTRÓNICA
SAN JUAN DE PASTO
2018

SISTEMAS DE RAZONAMIENTO BASADO EN CASOS PARA APLICACIONES MÉDICAS CON ETAPAS DE ADAPTACIÓN Y RECUPERACIÓN MEJORADAS

DAVID RAMIRO BASTIDAS TORRES
CAMILO ANDRÉS PIÑEROS RODRÍGUEZ

Trabajo de grado para optar por el título de Ingenieros Electrónicos

ASESOR
PhD. DIEGO HERNÁN PELUFFO ORDÓÑES

UNIVERSIDAD DE NARIÑO
FACULTAD DE INGENIERÍA
INGENIERÍA ELECTRÓNICA
SAN JUAN DE PASTO
2018

NOTA DE RESPONSABILIDAD

“La Universidad de Nariño no se hace responsable por las opiniones o resultados obtenidos en el presente trabajo y para su publicación priman las normas sobre el derecho de autor.”

Acuerdo 1. Artículo 324. Octubre 11 de 1966, emanado del Honorable Consejo Directivo de la Universidad de Nariño.

Nota de aceptación

_____

_____

_____

_____

_____

_____

Firma del presidente del jurado

_____

Firma del jurado

_____

Firma del jurado

San Juan de Pasto, 1 de Agosto de 2018

RESUMEN

En este trabajo se presenta una modificación a la metodología convencional en el proceso de clasificación en un sistema de razonamiento basado en casos (CBR), el cual está fundamentado en el aprendizaje adquirido por expertos. Su objetivo es solucionar futuros problemas de características similares a las adquiridas y recopilar información de posibles nuevos casos de forma que se mantenga un aprendizaje continuo. Las aplicaciones más frecuentes están en el ámbito médico, en donde el CBR puede proveer información confiable que sirve de apoyo al diagnóstico, en especial, en situaciones donde es difícil determinar con exactitud una patología.

En la actualidad, los sistemas CBR proveen una clasificación de datos con tendencia a un porcentaje de error aceptable, y por este motivo son susceptibles a mejoras en aspectos como precisión, exactitud e interacción con el usuario de tal manera que sean más explicativos y claros en la información que entregan como respuesta.

Para lograrlo, aquí se modifica la metodología como sigue: Primero, se realiza un estudio comparativo entre métodos de selección de características y balanceo de datos (etapa conocida como Pre-procesamiento que es independiente al ciclo del CBR propuesto). Segundo, se utiliza la fusión de las etapas de Recuperación y Adaptación de un sistema CBR convencional utilizando clasificación en cascada y un sistema de estimación de probabilidad empleando Maquinas de vectores de soporte (por sus siglas en ingles SVM). Los resultados obtenidos comprueban que el CBR propuesto tiene un mayor grado de precisión y exactitud en comparación a otras aproximaciones, y que además tiene la habilidad de retener la nueva información de manera automática o según la valoración de un especialista.

Se verifica que la metodología propuesta con clasificación en cascada y estimación de probabilidad SVM mejora los resultados obtenidos con respecto a CBR convencionales, en donde se utiliza clasificadores individuales y ningún proceso de estimación de probabilidad. De igual manera, se demuestra que el sistema SVM ofrece mejores resultados como estimador de probabilidad frente a otros estimadores comunes como los basados en vecinos más cercanos (por sus siglas en ingles KNN) y ventanas de Parzen.

ABSTRACT

This paper presents a modification to the conventional methodology in the classification process in a case-based reasoning system (CBR). This system is based on learning information provided by an expert experience. Its objective is to solve future problems of similar characteristics to those acquired and collect information on new cases in order to maintain continuous learning process. The most frequent applications, for this system, are in the medical field because they provide reliable information that supports diagnosis, especially in situations where it is so difficult to accurately determine a pathology.

Currently these systems provide a classification process with a tendency to an acceptable percentage of error, and for this reason are susceptible to improvements in aspects such as precision, accuracy and interaction with the user, in such a way, that they are more explanatory and clear in the information that they deliver as an answer.

To achieve this, the methodology is modified as follows: First, a comparative study is carried out between methods of feature selection and data balancing (a stage known as Pre-processing that is independent of the proposed CBR cycle). Second, a merge between the Recovery and Adaptation stages of a conventional CBR system is used using sequential classification and probability estimation with support vector machines or SVM.
The results obtained have a greater degree of precision and accuracy in comparison to other approaches, and the retention of new information is allowed automatically or according to the assessment of a specialist.

It is verified that the proposed methodology with cascade classification and SVM probability estimation improves the results obtained with respect to conventional CBRs, where individual classifiers are used and no probability estimation process. Likewise, it is shown that the SVM system offers better results as a probability estimator compared to other common estimators such as k-nearest neighbor or KNN and Parzen windows.

TABLA DE CONTENIDO

LISTA DE FIGURAS

LISTA DE TABLAS

LISTA DE ANEXOS

GLOSARIO

Inteligencia artificial: Es un área multidisciplinaria que combina diferentes áreas como la computación y la lógica, cuyo objetivo es dar a la maquina la capacidad de resolver problemas o realizar tareas por sí mismos, utilizando algoritmos y paradigmas de comportamiento humano.

Aprendizaje automático (Machine Learning): Se refiere a una rama de la inteligencia artificial dedicada al estudio de técnicas para darle a la maquina la capacidad de decidir. Estas técnicas se basan en el uso de clasificadores y probabilidad, entre otros. En la actualidad se realiza un estudio intenso de esta área por el aumento del uso de la automatización en diferentes ámbitos de la vida diaria.

Razonamiento: Es el proceso de organizar y estructurar las ideas para obtener respuestas y resoluciones a los problemas de cualquier índole.

Caso: Es la descripción ordenada de datos que tienen como fin dar una respuesta.

Base de casos: Es la materia prima del sistema de predicción. Es el histórico de casos que se usa para entrenar al sistema que detecta los patrones. El conjunto de casos se compone de instancias o muestras, y las instancias de factores, características o propiedades.

Clase: Agrupación de objetos que tiene características comunes.

Características: Son los atributos que describen cada una de las instancias del conjunto de datos.

Clasificación: Es la asignación de un objeto a una de las diversas categorías o clases especificadas.

Combinación de clasificadores: Es un enfoque donde los clasificadores se configuran en diferentes arquitecturas o sus resultados son combinados de diversas formas para obtener una mejor clasificación.

Algoritmo: Conjunto definido de reglas o procesos que llevan a la solución de un problema en un número determinado de pasos.

Diagnóstico médico: Parte de la medicina que tiene por objetivo identificar una enfermedad basándose en los síntomas que presenta el paciente, el historial clínico y los exámenes complementarios.

Pre-procesamiento de los datos: Es una actividad donde se depuran las bases de datos, utilizando diferentes mecanismos como: la reducción de características o el balanceo de clases entre otros.

Selección de características: Hace referencia al proceso de reducir las entradas para su procesamiento y análisis, o de encontrar las entradas más significativas.

Desbalance de clases: Se presenta cuando existen conjuntos de datos que tienen una cantidad grande de datos de cierto tipo (clase mayoritaria), mientras que el número de datos del tipo contrario es considerablemente menor (clase minoritaria).

Estimación de probabilidad: Es el conjunto de técnicas que permiten dar un valor aproximado de un parámetro de una población a partir de los datos proporcionados por una muestra.

Diagrama de dispersión: Es un diagrama matemático que utiliza coordenadas cartesianas para expresar un conjunto de datos por medio de la representación de sus valores.

Matriz de confusión: Es una herramienta que permite la visualización del desempeño de un algoritmo que se emplea en aprendizaje supervisado. Cada columna de la matriz representa el número de predicciones de cada clase, mientras que cada fila representa las instancias en la clase real.

Curva ROC (**Receiver Operating Characteristic**): Es una herramienta para seleccionar los modelos posiblemente óptimos.

Gráfico de caja y bigotes: Se refiere a un diagrama que sirve para visualizar la distribución de un conjunto de datos basado en cuartiles.

ACRÓNIMOS

CBR	Case-based reasoning (Razonamiento basado en casos)
Cfs-SubselEval	Criterio de evaluación de subgrupos basado en correlación
InfoGainAttribute	Criterio de evaluación basado en ganancias
SVM	Support Vector Machine (Máquinas de soporte vectorial)
K-NN	Nearest Neighbor (Vecinos cercanos)
ROC	Receiver Operating Characteristic (Característica operativa del receptor)
SMOTE	Synthetic Minority Over-sampling Technique (Técnica sobremuestreo clase minoritaria)
KNN-U	Nearest Neighbor Under-sampling (Submuestreo usando vecinos más cercanos)
MSE	Mean Squared Error (Error cuadrático medio)
Se	Sensibilidad
Sp	Especificidad
Acc	Exactitud
MC	Matriz de Confusión

1. INTRODUCCIÓN

Uno de los retos más importantes, hoy en día, es emular el comportamiento de la inteligencia humana en la toma de decisiones y el aprendizaje continuo. Entre los enfoques existentes se ha propuesto una metodología de razonamiento basado en casos CBR (por sus siglas en inglés), que brinda a la máquina la capacidad de seleccionar una decisión apropiada con respecto a un tema en particular y además aprender acorde los resultados de la misma mediante una retroalimentación directa a su base de datos [1].

Tal y como lo proponen los autores en [2] [3],la metodología de un CBR consiste en cuatro etapas: Primero, la recuperación, en donde a partir de un nuevo caso a analizar se seleccionan casos en la memoria con características similares que permitan proponer una solución adecuada; segundo, la reutilización en donde se determina una solución a sugerir; tercero, la revisión en donde se evalúa el resultado de la sugerencia; y, por último, la etapa de retención, en la que se actualiza todo el sistema y así éste aprende de la experiencia.

Es claro que, cada una de estas etapas requiere una profundización con el fin de optimizar el proceso en general, incluso se han propuesto aproximación de etapas pre-procesamiento tales que mejoren la calidad de los datos a analizar. En este sentido, las técnicas de minería de datos, y reducción de dimensión resultan útiles cuando se trata con grandes volúmenes de información con propiedades complejas e inciertas en el contexto clínico.

Si bien es cierto que los campos de acción del CBR son muchos, hoy en día uno de los más notorios es el de la medicina [4] (debido a su relevancia), puesto que los especialistas se encuentran diariamente frente al problema de diagnóstico con base en información confusa y de características extensas. Cabe resaltar que son múltiples los desarrollos para apoyo al diagnóstico [5], no obstante la gran mayoría brinda únicamente dos opciones como resultado final: normal o patológico [6]. Esta información no resulta útil en casos donde la patología presenta sintomatología variada y debe conocerse claramente a qué está asociado cada cuadro clínico.

En este trabajo, se propone una modificación a la metodología convencional del CBR, de tal manera que se optimiza cada una de las etapas y por ende el sistema en conjunto. Para ello introduce la clasificación multi-clase en cascada y un sistema de estimación de probabilidad basado en SVM en la etapa de adaptación

y recuperación fusionadas, logrando así la automatización del proceso sin descartar la experiencia del especialista puesto que ésta es una herramienta de apoyo al diagnóstico y siempre dará la opción para que en la etapa de retención sea el experto quien evalué y estipule los procedimientos a seguir. Es de anotar que se incluye una etapa de pre-procesamiento de datos, que brinda robustez y permita atacar problemas con datos de entrada complejos y el desarrollo de una interfaz interactiva que facilita el acceso de la metodología propuesta, plasmada en una herramienta informática de fácil utilización.

1.1. PLANTEAMIENTO DEL PROBLEMA

El proceso de sistema de razonamiento basado en casos pretende simular mediante un algoritmo el modelo cognitivo humano. En el ámbito medico es utilizado con el fin de ayudar al diagnóstico de diversas patologías, analizando diferentes casos con base en un aprendizaje realizado. Los desarrollos en esta área han mostrado interés en mejorar partes del sistema de tal manera que se ofrezcan mayores herramientas que permitan al experto establecer de una manera efectiva y rápida un diagnóstico claro en relación a la gran variedad de casos referentes a una enfermedad.

Una de estas partes del sistema se conoce como clasificación, este proceso depende principalmente de la representación y el tipo de información con la que se esté trabajando, ya que en algunos casos se presenta una buena asignación y en otros, se requiere utilizar uno o varios métodos diferentes, (por ejemplo, modificando clasificadores, buscando nuevas formas de clasificación o mejorando la representación de la información dada si es posible). En algunos casos el uso de un proceso de clasificación no es suficiente, por lo tanto es necesario indicar la correcta estimación de pertenencia de un caso a todas las clases posibles, dado que la diferencia de porcentajes puede determinar un diagnóstico correcto de uno errado, en consecuencia una mejora en la clasificación de un sistema multi-clase CBR junto con una presentación de probabilidad adecuada brindará un mayor porcentaje de éxito en el correcto diagnóstico patológico en ambientes médicos actuales.

1.2. JUSTIFICACIÓN

El comportamiento de un sistema CBR se basa principalmente en el proceso de emplear la experiencia en la resolución de nuevos problemas con características similares, proporcionando generalmente un resultado con solución exitosa, producto del empleo de dichas acciones. De momento, la técnica CBR es uno de los procesos que junto con minería de datos son los más utilizados en resolución de problemas de diversa índole [7], debido a su gran eficiencia y apoyo en ámbitos

científicos para proveer una respuesta completa y rápida a problemas complejos, que generalmente toman mucho tiempo en solucionar de manera convencional.

En desarrollos médicos es habitual la existencia de numerosos casos en los que se observa (debido a su complejidad) varios factores influyentes a la hora de determinar un diagnóstico claro y conciso, como por ejemplo en enfermedades de tipo cardíaca identificada a partir de señales denominadas electrocardiogramas. De esta manera, el mínimo error en el análisis de los datos de un paciente en un estado crítico, podría conllevar a graves consecuencias principalmente al paciente y recíprocamente al experto implicado [8]. Bajo esta premisa, el hecho de contar con un sistema confiable para asistencia al diagnóstico en este tipo de escenarios, promueve el constante desarrollo y mejora progresiva de los sistemas con miras a ofrecer un servicio cada vez más acertado que ayude al profesional capacitado a ser lo más objetivo posible.

1.3. CONTRIBUCIONES DE ESTA TESIS

Es claro que el desarrollo tecnológico ha incursionado en todas las áreas del conocimiento. Digno representante de ello, son las ciencias médicas que han aprovechado el desarrollo de diferentes equipos y elementos de software para mejorar el diagnóstico de enfermedades para su posterior tratamiento. Esta investigación aporta al estudio de sistemas de razonamiento basados en casos (CBR) y máquinas de aprendizaje mediante el desarrollo de un sistema de apoyo al diagnóstico médico bajo una metodología modificada. El aporte puede referirse en tres partes:

- Desarrollo de un esquema de pre-procesamiento usando métodos de selección y balanceo para mejorar la representación de casos para problemas de diagnóstico médico.
- Extensión de un sistema de razonamiento basado en casos para la inclusión de la clasificación en cascada y un estimado de probabilidad en las etapas de recuperación y adaptación.
- Diseño e implementación de una interfaz que tenga como características un fácil manejo e interpretación de sus resultados para apoyar el diagnóstico médico.
- Presentación de un artículo resultado de la esta investigación en congreso de carácter internacional.

1.4. ORGANIZACIÓN DEL DOCUMENTO

El presente documento consta de 7 secciones las cuales se encuentran constituidas de la siguiente manera: Introducción, descripción del problema,

objetivos, marco teórico, metodología, marco experimental, resultados, conclusiones y trabajos futuros.

En la sección 1, se presenta la introducción donde se hace, la descripción del problema, la justificación y que contribuciones se presentan con el desarrollo de esta tesis.

En la sección 2, se presenta los objetivos propuestos en torno al tema de investigación planteado.

En la sección 3, se presentan definiciones respecto al tema tratado en toda la investigación, tales como: Componentes de un CBR, clasificación en cascada métodos de estimación de probabilidad con método SVM, entre otros.

En la sección 4, se muestra la metodología propuesta específica en pro del desarrollo y mejora de un sistema CBR Multi-clase.

En la sección 5, se concentran todos los experimentos o pruebas realizadas además de las diferentes bases de datos usadas, los cuales servirán de sustento para la metodología propuesta.

En la sección 6, se muestran los resultados obtenidos, mostrando mejoras en el proceso de clasificación, la estimación de probabilidad y presentación de información de carácter médico.

Finalmente, en la sección 7 se muestran las conclusiones encontradas al terminar este proyecto y se analizan trabajos futuros que ayuden a encontrar una mejor alternativa en la búsqueda de una mejor respuesta de sistemas CBR convirtiéndolo en un sistema cada vez más confiable para la medicina.

2. OBJETIVOS

En esta sección se plantea los objetivos esperados con el desarrollo de esta investigación.

2.1. OBJETIVO GENERAL

Proponer una mejora a los sistemas de razonamiento basado en casos desarrollando una estimación de las probabilidades de pertenencia de los nuevos casos y usando clasificadores multi-clase en cascada con el fin de mejorar la asistencia diagnóstica en entornos médicos**.**

2.2. OBJETIVOS ESPECÍFICOS

- Diseñar etapas de recuperación y adaptación de un sistema de razonamiento de casos a través de clasificación en cascada.

- Desarrollar una forma de estimación de las probabilidades de los nuevos casos a las clases de casos conocidos de forma coherente con el clasificador.

- Integrar la clasificación en cascada y estimación de las probabilidades de pertenencia en el ciclo de vida de un sistema de razonamiento basado en casos para proporcionar una herramienta mejorada para soporte de asistencia diagnóstica médica

3. MARCO TEÓRICO

En esta sección se da a conocer los aspectos fundamentales que definen un sistema de razonamiento basado en casos, los sistemas multi-clasificadores, los clasificadores y estimadores de probabilidad utilizados en esta investigación, dando a conocer sus aspectos teóricos más importante, los cuales sirven de sustento a este trabajo de grado.

3.1. ANTECEDENTES

El inicio de los CBR tiene como origen el trabajo realizado por el grupo de investigación de la Universidad de Yale dirigido por Robert Schank [9] en el año 1977, quienes construyeron el primer modelo cognitivo. Este tomaba como punto de partida la idea de que los conocimientos humanos sobre distintas situaciones se guardan en la mente en forma de recuerdos y éstos son utilizados para llegar o inferir conclusiones respecto al primer modelo de memoria dinámica. Partiendo de esta investigación Janet Kolodner desarrolló el primer (CBR) [10]. En este se presenta una organización de memoria conceptual implícita en la memoria reconstructiva humana, fue implementado con éxito en un sistema denominado CYRUS, que almacenaba eventos acerca de las vidas de los ex Secretarios de Estado Cyrus Vance y Edmund Muskie y respondía preguntas formuladas en inglés sobre información referente a su actividad.

Por otra parte, los sistemas basados en casos desde su invención han sido utilizados en diversas áreas del conocimiento como la salud, agricultura, ingeniería, el derecho y otras. Un trabajo destacado en el área de la salud fue el realizado por Bruce Porte en 1986 [11] de la Universidad de Texas, quien creó el sistema PROTOS como una alternativa al modelo creado por Schank. Este utiliza clasificaciones heurísticas y métodos de aprendizaje de máquina, aplicados al aprendizaje de trastornos al oído. Por su parte, en la Universidad de Salford este mismo desarrollo se aplicó a diagnósticos fallidos como apoyo al área de ingeniería civil, para rehabilitación y reparación de edificios, así como a la construcción [12].

En la actualidad existen diversos grupos que se dedican a realizar investigaciónes en este tema, cabe destacar los siguientes grupos, GAIA-Group or Artificial Intelligence Applications (Universidad Complutense, Madrid), BISITE Bioinformática. Sistemas Inteligentes, Tecnología Educativa (Universidad de Salamanca. Instituto de Investigación en Inteligencia Artificial-IIIA (Consejo Superior de Investigaciones Científicas, CSIC), entre otros.

3.2. RAZONAMIENTO BASADO EN CASOS

Un sistema de razonamiento basado en casos busca solucionar problemas haciendo uso de conocimientos pasados. Al enfrentarse a un problema el sistema entrega soluciones que funcionaron bien en situaciones similares, utilizándolas como parte inicial en la resolución de dicho problema [3]. Un ejemplo de cómo actúa un CBR es el diagnóstico médico. El medico tiene una serie de experiencias recopiladas adquiridas en el proceso de su formación, de manera tal que cuando un nuevo paciente es tratado, lo compara con los síntomas similares de otros casos tratados anteriormente, pudiendo entonces dar el mismo tratamiento y así se agrega nuevo conocimiento a su experiencia.

El propósito de un sistema CBR es resolver problemas adaptando soluciones de problemas pasados, lo que se conoce como razonamiento a través de la experiencia, debido a que se guardan problemas en la memoria con sus respectivas soluciones con el propósito de resolver los nuevos problemas haciendo referencia a ese conocimiento almacenado [2].

El funcionamiento de un sistema CBR se resume de la siguiente manera: Se presenta un nuevo problema, el sistema busca en su memoria problemas pasados en la base de casos, después intenta encontrar un problema cuyas características sean iguales o similares al del problema tratado. En caso de que el sistema encuentre un problema similar, buscará uno o varios que sean lo más parecido posible al caso actual. En situaciones donde el caso recuperado es igual al problema planteado, y asumiendo que la solución con la que se almacenó fue un éxito, ésta es propuesta como solución. De otro modo, se adapta la solución. Para este proceso se identifican las diferencias entre ambos casos y se modifica la solución del caso recuperado tomando en cuenta tales diferencias.

3.2.1. Definición de caso y Base de Casos

Dentro de este sistema CBR es importante el concepto de caso, presentado dentro de todo el ciclo de vida de este sistema. Un caso se describe como un conjunto de circunstancias ocurridas en el mundo real. Este se compone de tres partes principales: el problema, su solución y la correspondiente respuesta [10]. Un caso puede tomar muchas formas, que van desde simples datos a información o conocimiento, compuestos por un conjunto de atributos finitos que determinan los valores de las diferentes características que definen el problema [3]. Por lo tanto, un caso se puede considerar como un vector que contiene los valores que toman dichos atributos. Estos atributos pueden ser cualitativos o cuantitativos dependiendo de la naturaleza del problema, donde la solución del problema puede

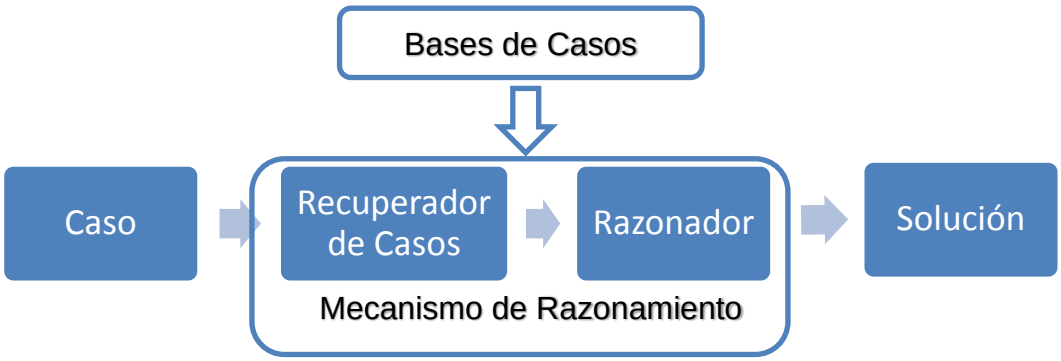
ser uno o más atributos [13].

3.2.2. Estructura y Ciclo de Vida del CBR

➢ Estructura CBR

La estructura general de un sistema de CBR está compuesta por tres partes, el caso, la base de casos que es utilizada por el mecanismo de razonamiento para obtener la solución correcta [14], esto se muestra en la Figura 1.

Figura 1. Estructura CBR. Se muestra la estructura general de un CBR y la interacción de sus tres partes, se observa que el caso y la base de casos entran al mecanismo de razonamiento para entregar una solución.



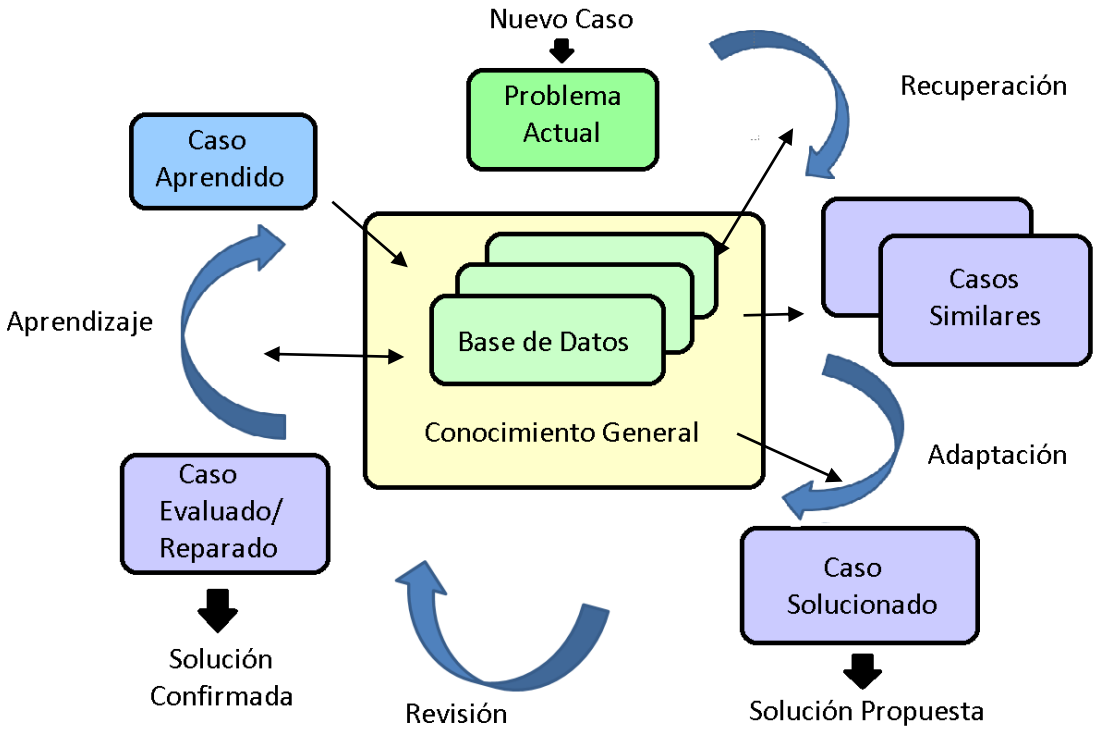
En cuanto a la estructura interna del mecanismo de razonamiento se divide en dos partes principales que son: el recuperador de casos y el razonador de casos. El primero se encarga de encontrar el caso más apropiado en la base de casos, mientras que la del segundo utiliza el caso recuperado para encontrar la solución al problema. Dada la coincidencia de un caso recuperado con otro caso, no es necesario ningún razonamiento, debido a que el caso recuperado contiene directamente la solución del caso actual, y carece de importancia en el proceso interno de funcionamiento del CBR [14].

➢ **CICLO DE VIDA DE UN CBR**

Tomando este proceso en forma cíclica el sistema basado en CBR consta de cuatro fases principales según Ammoth [3]: Primero el **recuperación**, que denota identificar el problema actual y encontrar un caso pasado similar al nuevo caso, segundo la **adaptación** que es utilizar el caso seleccionado previamente y sugerir una solución al problema actual, tercero la evaluación de la solución propuesta, conocida como etapa de **revisión** y por último y no menos importante actualizar el sistema para aprender de la experiencia, proceso conocido como **retención**. Lo

anterior se muestra en la Figura 2.

Figura 2. Ciclo de Razonamiento Basado en Caso según Aamodt. Se exponen las cuatro etapas que conforman este sistema, la recuperación, la adaptación, la revisión y el aprendizaje o también conocido como retención, y cuál es su interacción con la base de casos **Fuente [6].**



- **Recuperación *(Retrieve)***

Dentro de la base de casos se busca aquellos semejantes al caso actual, comparando sus características. En esta etapa se utilizan varios métodos desde los simples basados en distancias hasta otros mucho más complejos, basados en lógica difusa [6].

Luego de la recuperación del caso, se realiza un análisis para determinar si éste es suficientemente parecido al caso planteado. La similitud puede ser interpretada de diferentes maneras: El grado de relación de las características, la situación contextual o la dificultad que supondría su adaptación.

- **Reutilización o Adaptación *(Reuse)***

La solución recuperada es procesada para adecuarla y así solucionar el problema. Hay dos formas generales de hacer adaptación:

i) Mediante la sustitución: Los valores que aparecen en el caso recuperado de la memoria son sustituidos por aquellos valores del caso actual, de forma que la nueva solución hace uso de la situación actual que se requiere resolver.

ii) Mediante la aplicación: Al caso que hay que resolver, se le aplican el mismo conjunto de procedimientos, reglas o inferencias utilizados como solución del caso recuperado de la base de casos.

Después de elegir la solución, se debe comprobar si esta solución adaptada tiene en cuenta las diferencias entre el caso recuperado y el problema actual, si no fuera el caso, en esta etapa se debe considerar la decisión a tomar si la solución propuesta no resuelve el problema con éxito.

En el proceso de aprendizaje el sistema necesita de un criterio capaz de valorar el rendimiento de la prueba para que el sistema se renueva con la información obtenida acerca de la solución. Esta información ya almacenada aumenta la probabilidad de encontrar un caso igual o parecido convirtiendo el sistema más robusto y completo. Así, el aprendizaje puede ocurrir no sólo cuando el caso ha sido solucionado con éxito.

- **Revisión *(Revise)***

La revisión puede comprender dos fases: Una se encarga de evaluar la solución y la otra de reparar los fallos [19]. En la primera fase, la evaluación la puede hacer un experto que determine si la solución fue acertada o no, esta es la forma más común de revisión en un CBR clásico. También el sistema puede hacer la revisión de forma automática, previo a que conozca la solución acertada. La segunda, actúa si la solución no es aceptada, entonces el caso debe ser modificado, por lo tanto, se da un aprendizaje continuo en donde el sistema aprende de sus errores.

- **Aprendizaje *(Retain)***

Con la presencia de un nuevo caso se hace necesario incorporar este a la base de casos existente. Este ayudara a resolver nuevos problemas. En esta etapa se realizan las siguientes actividades: añadir el caso y revisar la base de casos o eliminar casos obsoletos o redundantes (aunque también incluye cambios en el conocimiento). El sistema debe decidir si es necesario guardar el caso con su solución, para esto debe revisar la base de casos puesto que pueden existir casos muy parecidos que al incluirlo puede generen redundancias, dando como resultado un sistema menos óptimo. Otro problema es la indexación de los casos, dependiendo de la complejidad de la estructura utilizada, este proceso puede ser más o menos complicado [11] :

- Si la organización es lineal, basta con añadir un nuevo elemento a la lista.

- Si la estructura se induce a partir de los casos, será necesario redefinir la periodicidad de la indexación. Normalmente este proceso se realiza fuera de línea para no perturbar la interacción del usuario con el sistema.
- En los modelos más complejos donde se presentan generalizaciones de los casos, es necesario aplicar técnicas de aprendizaje más sofisticadas, similares a las aplicadas en otros campos de inteligencia artificial.

3.3. DIAGNÓSTICO MÉDICO

Una vez definida la operación de un CBR, cabe resaltar que para los desarrollos tecnológicos para asistencia médica se entiende por caso a cada una de las distintas patologías asociadas al comportamiento de una señal.

Es claro que las señales biomédicas suelen contener gran cantidad de datos que revelan características importantes sobre el estado de un paciente respecto una parte de su cuerpo analizada (lo que depende del experimento y tipo de señal derivada). Los problemas inherentes que esto conlleva van ligados a la visualización de información y por ende a la interpretación de ella que consecuentemente causa dificultades al momento de establecer un diagnóstico. Técnicas como la minería de datos y los avances en interfaces para visualización e interpretación de grandes volúmenes de información, resultan eficaces en la tarea de asistir a la identificación y diagnóstico de una enfermedad determinada [5].

CBR ofrece una asistencia al diagnóstico médico recopilando información de un conjunto de datos conocido que le permite determinar una base de casos. A partir de esta, emite el grado de pertenencia de un nuevo caso analizado (señal proveniente de un nuevo paciente) a los casos comunes o registrados en su historial, mediante valores de probabilidad en función de un análisis matemático riguroso y la operación de algoritmos de selección y clasificación, como se verá en detalle en la sección 4.1 y 4.2. Es importante destacar que CBR únicamente da a conocer un resultado con base a su experiencia en clasificación y los aportes de un experto en su programación, no obstante, es directamente el profesional de la salud quien determina con base en su conocimiento y su visión de la realidad el diagnóstico final.

3.4. APRENDIZAJE DE MÁQUINA (Machine Learning)

El estudio de la inteligencia artificial busca metodologías que den a la máquina la capacidad de emular el comportamiento inteligente, similar al uso de la razón de los seres humanos con base en conductas habituales de los seres vivos, puesto

que la fuente de bio-inspiración no está únicamente en las personas. Un ejemplo claro se observa en algoritmos heurísticos que imitan el comportamiento de enjambre de hormigas, abejas y/o partículas para resolver problemas de toma de decisión particulares [15].

Sumado a lo anterior, cabe resaltar el avance matemático que aporta propuestas para el desarrollo de algoritmos capaces de trabajar con grandes volúmenes de información, con miras a la visualización y extracción de características, fundamentados en análisis de probabilidad y estadística [13].

Por otra parte, intuitivamente puede decirse que el uso de la razón de los seres humanos implica las siguientes 4 actividades: el aprendizaje, la retención, el análisis de información y la toma de decisiones. Dentro de estas, el aprendizaje juega un papel primordial ya que a partir de cómo se realice este proceso se determinará la calidad y aplicabilidad de los procesos siguientes [1].

Hay dos formas principales de aprendizajes, una llamada aprendizaje supervisado y el otro no supervisado. En el primero se busca brindar una premisa de conocimientos a partir de los cuales opera en la toma de decisiones. Estos conocimientos han sido previamente verificados por un experto en el área y garantizan óptimos resultados. El segundo pretende que la máquina sea capaz de organizar su propia base de conocimiento con base en una clasificación automatizada sustentada en un análisis matemático o heurístico que garantice buenos resultados, pero no siempre los más óptimos y en algunas ocasiones no necesariamente los correctos [16].

3.4.1. Clasificación Supervisada y No Supervisada

Dado que el aprendizaje implica una interacción entre el aprendiz y el entorno, es posible dividir las tareas de aprendizaje de acuerdo con la naturaleza de esa interacción [16]. Como ejemplo ilustrativo considere dos tareas: La primera tarea consiste en aprender a detectar una persona enferma. La segunda consiste en la detección de enfermedades con sintomatología desconocida. Para la tarea de detección de una persona enferma considere un entorno en el que el aprendiz recibe personas que únicamente pueden etiquetarse como sana o enferma. Sobre la base de dicha formación, el aprendiz debe encontrar una regla para etiquetar una persona recién llegada. Por el contrario, para la tarea de enfermedades con sintomatología desconocida, el aprendiz obtiene un entrenamiento a partir de un gran conjunto de personas (sin etiquetas) y de esta manera su trabajo está en identificar grupos que manifiesten relaciones marcadas en su sintomatología. La primera tarea es característica de un aprendizaje supervisado, por su parte la segunda de uno no supervisado.

El aprendizaje supervisado describe un escenario en el que la "experiencia", contiene información significativa (por ejemplo, sano y enfermo). La experiencia

adquirida tiene como objetivo predecir la información faltante para los datos de prueba. En tales casos, se puede pensar en el entorno como un maestro que "supervisa" al aprendiz al proporcionar la información adicional (etiquetas) [16]. En el aprendizaje no supervisado no hay distinción entre los datos de entrenamiento y prueba. Aquí el aprendiz procesa los datos de entrada con el objetivo de llegar a algún tipo de resumen o versión comprimida de esta información, agrupando un conjunto de datos en subconjuntos de objetos similares, por ejemplo el algoritmo Kmeans [17].

También existe un entorno de aprendizaje intermedio en donde, aunque los ejemplos de entrenamiento contienen más información que los ejemplos de prueba, se requiere que el aprendiz pronostique incluso más información para los ejemplos de prueba. Tales metodologías de aprendizaje son principalmente analizadas bajo el título de aprendizaje semi-supervisado [18].

3.4.2. Clasificación supervisada y No supervisada en CBR

En los sistemas de CBR se utilizan las dos formas de aprendizaje. Al existir una base de datos con soluciones reales que han sido previamente elaboradas por un experto, se utiliza la clasificación supervisada. La clasificación no supervisada es usada en etapas de pre-proceso como la técnica de "*cluster*" [6] o para generar reglas de asociación y emplearlas en la etapa de adaptación.

Existen desarrollos en clasificación supervisada aplicados al CBR que han servido para el diagnóstico de enfermedades, un ejemplo se encuentra en [19] donde se compara el resultado de una clasificación aplicando cinco técnicas diferentes para diagnosticar una enfermedad hepática, estas son: Redes neuronales (ANN, por sus siglas en inglés) de retro-propagación, arboles de regresión, regresión lineal, CBR con 10 vecinos cercanos utilizando distancia euclidiana y un modelo hibrido CBR-ANN. De los cinco sistemas comparados, la mejor respuesta se obtiene con el sistema hibrido CBR-ANN. Otras formas en que han sido utilizadas los sistemas CBR es para ayudar a la prescripción médica como se puede observar en [20] donde se utiliza en paralelo un sistema de CBR y un clasificador bayesiano. Este sistema recupera los casos más parecidos a un caso problema y en paralelo realiza una clasificación aplicando razonamiento bayesiano con los patrones de prescripción de experiencias previas. Se compara la respuesta de los dos sistemas a través de reglas IF-THEN; si los medicamentos son iguales en las dos respuestas, esa es la solución, en caso contrario pasa a otro sistema de reglas para calcular los medicamentos y las dosis correspondientes [6].

En tanto a la clasificación no supervisada en CBRs destaca el estudio expuesto en en [21] donde se propone un sistema hibrido entre CBR y árboles de decisión difusos para aplicarse en diagnóstico médico. Se realizan pruebas con bases de datos de cáncer de mama y enfermedad del hígado. En una primera fase utilizan un análisis de regresión para ponderar los pesos de cada característica, este

enfoque transforma la matriz de similitud en una matriz de equivalencia, con el fin de agrupar los casos equivalentes entre si y clusterizar. Una vez se tienen los clúster, se aplica un árbol de decisión difuso a cada clúster para construir un sistema de reglas para la toma de decisiones estudio [6].

3.5. SISTEMAS MULTICLASIFICADORES

Los clasificadores pueden ser agrupados con el fin de mejorar el resultado individual de cada clasificador. Esto debido a la existencia de señales que son difíciles de clasificar o la necesidad de mejorar la exactitud y precisión de la aplicación. Este conjunto, también conocido como sistema multiclasificador puede ser categorizado de acuerdo a diferentes criterios. Por ejemplo, si el conjunto representa la fusión o selección de clasificadores o si generan o no el ensamblado de clasificadores. Según [22] la clasificación es:

i. *Métodos basados en generación de ensamblados*. Estos sistemas fijan un esquema de combinación como por ejemplo el voto mayoritario y se encargan de generar los clasificadores que van a conformar el ensamblado. La tendencia general consiste en conformar clasificadores independientes en cuanto a sus respuestas. Ejemplos de estos métodos son el Bagging, Boosting, Bosques Aleatorios, etc.

ii. *Métodos basados en selección de clasificadores*. Sea un conjunto de L clasificadores ya entrenados. Estos métodos tratan de seleccionar cuál de los L clasificadores es el más apropiado para asignar una clase a un objeto.

iii. *Métodos basados en combinación de clasificadores*. Dado un conjunto de L clasificadores, estos métodos combinan o fusionan los L resultados de sus miembros para retornar una respuesta final.

iv. *Métodos híbridos.* Aquí se agrupan los métodos que combinan varias o todas las estrategias descritas anteriormente.

Otro aspecto importante en los sistemas multi-clasificadores es su arquitectura o topología que denota la forma en que se desea integrar un conjunto de L clasificadores para garantizar una toma de decisión. En el trabajo expuesto en [23] se categorizó tres grupos: en cascada (vertical), paralela (horizontal) e híbrida (jerárquica).

3.5.1. Arquitectura de Cascada

Este tipo de arquitectura se caracteriza porque los clasificadores van en serie (uno detrás del otro), esto implica que el resultado del siguiente clasificador se vea

afectado por la salida del anterior. Esta arquitectura puede utilizar cualquiera de los cuatro métodos expuestos anteriormente, es así como en el método de ensamblados el algoritmo Adaboost utiliza esta arquitectura colocando un calsificador (Weak classifier) en serie y lo va entrenando en toda la cadena para obtener un mejor resultado, pero en el caso de este trabajo se utiliza la metodología 3 basada en combinación de clasificadores.

Por lo general la clasificación en cascada con combinación de clasificadores presenta dos niveles. El nivel 1 se entrena con el conjunto de datos original, mientras que el nivel 2 se entrena con un conjunto de datos aumentado, el cual contiene las características del conjunto de datos original junto con la salida del clasificador del nivel 1. La salida del clasificador del nivel 1 es un vector conteniendo la distribución de probabilidad condicional $(\mathrm{p1},\ldots,\mathrm{pc})$, donde c es el número de clases del conjunto de datos original, y pi es la estimación de probabilidad calculada por el clasificador del nivel 1 de que la instancia pertenezca a la clase i. La cascada es una aproximación que se puede extender a más de dos
niveles [24].

El entrenamiento de un clasificador A con la salida de otro B hace que A se vea influenciado notablemente por B, derivando en un esquema global sobreentrenado. Sin embargo, en la cascada se reduce este problema porque:

- En cada nivel se utiliza un clasificador de naturaleza distinta al del otro.

- Porque el clasificador del nivel 2 no se entrena únicamente con la salida del clasificador de nivel 1, sino que además tiene en cuenta las características originales

En este trabajo además de utilizar la forma general de la cascada se pretende extenderla a la utilización de 3 clasificadores de diferente naturaleza, para analizar el comportamiento de los datos, a tener una presencia más fuerte de clasificadores.

3.6. PRE-PROCESAMIENTO DE DATOS

El pre-procesamiento de datos juega un papel relevante en el CBR, ya que dependiendo de la calidad con que ingrese la información al sistema se logrará realizar una correcta o una incorrecta clasificación. Esto, según los requerimientos de los algoritmos empleados que generalmente ofrecen un desempeño óptimo cuando los datos de entrada son coherentes y balanceados [6]

Lo anterior debido a que los registros de una señal biomédica son susceptibles al ruido de armónicos provenientes del equipo utilizado, incompletos dependiendo

del tipo de análisis al paciente e inconsistentes según factores externos. Por este motivo resulta primordial aplicar técnicas de limpieza, integración y transformación de datos [25] [26] [27].

3.6.1. Normalización

Esta etapa mapea el rango de valores de los atributos de su tamaño estándar a un rango normalmente de -1 a 1 o de 0 a 1.
Según [28] evita que valores altos adquieran un peso mayor que valores bajos en los resultados del modelo final. De esta manera, es una técnica útil cuando los atributos tienen órdenes de magnitud diferentes.
Para su desarrollo basta con obtener la media aritmética de los valores a normalizar para posteriormente dividir cada magnitud entre esta.

3.6.2. Selección de características.

El objetivo de esta etapa es reducir el conjunto de datos para optimizar la complejidad computacional, es decir disminuir el tiempo de ejecución y el espacio de memoria utilizada.

Consiste en seleccionar muestras representativas eliminando información repetida y ruidosa. Así, se evita el sobre aprendizaje y el incorrecto aprendizaje. Cabe anotar que existen distintas técnicas para realizar este proceso dependiendo del tipo de variable a trabajar [6]que pueden ser:

i) *Variables Ordinales:* Son de tipo cualitativo y pueden organizarse por jerarquía, por ejemplo: pertenencia a un grupo socioeconómico.

ii) *Variables Cardinales:* Son de tipo cuantitativo y a su vez pueden ser: Continuas, cuando toman cualquier valor en un intervalo, por ejemplo: Edades, Salarios, Estatura. Discretas, cuando toman algunos valores en un intervalo, por ejemplo: Hijos por familia, producción mensual de automóviles, entre otras.

Tal y como lo muestran la autora en [6] para el desarrollo de esta investigación se utiliza la selección de características basada en filtros de correlación y búsqueda de profundidad que se describen a continuación.

- **Cfs-SubsetEval** (Correlation Feature Selection),

Evalúa el valor de un subconjunto de atributos al considerar la capacidad predictiva individual de cada característica junto con el grado de redundancia entre ellas. Esto lo hace mediante un proceso de jerarquización de subconjuntos de atributos de acuerdo a su correlación basada en una función

de evaluación heurística (basada en el cálculo de la correlación estadística), buscando atributos que están muy poco correlacionados entre sí, pero tienen una buena correlación con la clase [29]. Este proceso hace que las características menos relevantes sean ignoradas debido a su correlación casi nula con la clase y las características redundantes serán penalizadas debido a alta correlación entre esas características. La función de evaluación que se utiliza para este proceso es:

$$\boldsymbol{M_{Sy}} = \frac{\boldsymbol{Y\overline{ref}}}{\sqrt{\boldsymbol{Y+Y(Y-1)\overline{\overline{rff}}}}} \quad , \qquad (\mathbf{1})$$

donde:

- M_S es el merito heurístico del subconjunto S con Y características
- $\overline{ref}$ es el valor promedio de todas las correlaciones entre clase-característica
- $\overline{\overline{rff}}$ es el valor promedio de todas las correlaciones característica-característica del subconjunto S

Una desventaja de este método, es que al asumir que las características son condicionalmente independientes dada la clase, si existe una fuerte interacción entre los distintos atributos, entonces CFS no puede garantizar que los atributos seleccionados sean relevantes [30], algo que en algunas bases de datos puede ser crítico, por esto es necesario utilizar el método complementario como el método de búsqueda Best first.

- **Best First**

El Best First es un algoritmo de búsqueda mediante un árbol. Consiste en ir eliminando atributos hasta llegar a un cierto número de estos pre- determinados por el usuario. Este algoritmo puede comenzar con el conjunto vacío de atributos y buscar hacia adelante, o comenzar con el conjunto completo de atributos y buscar hacia atrás, o comenzar en cualquier punto y buscar en ambas direcciones (considerando todas las posibles adiciones y eliminaciones de atributos en un punto dado). Cada subconjunto es evaluado cada vez que se quitan atributos, esta eliminación se hace siguiendo un esquema ordenado de eliminación de atributos (esquema de enumeración). Si algún subconjunto obtiene un valor igual o peor a una cota (resultado de la evaluación del

subconjunto de atributos mediante una métrica mono-tónica), se detiene la exploración de esa rama (es decir, se realiza una poda), debido a que continuar la exploración es inútil, conduciendo a una mejor solución. Por otro lado, si todos los subconjuntos evaluados resultan mejor que la cota, se actualiza la cota con el nuevo valor, y se repite el procedimiento hasta que no existan más ramas a explorar [30].

- **Ranker**

El método Ranker es un método de búsqueda que usa evaluaciones individuales para la clasificación de atributos [31]. Este método se evalua en conjunto con los evaluadores de atributos (ReliefF, Gain Ratio, Entropy, etc.) con el fin de generar un parámetro de ranking (verdadero o falso). Ranker proporciona una calificación de los atributos, ordenados por su puntaje al evaluado.

- **InfoGainatribbute**

El infoGainattribute es un evaluador atributos que evalúa el valor de un atributo midiendo la ganancia de información con respecto a la clase, se utiliza en la selección de características de manera predeterminada con el método de búsqueda de ranker. Una característica relevante es que la ganancia de información está sesgada hacia atributos multi-valor, el atributo a seleccionar será aquel con mayor ganancia de información.

Tomando P_i la probabilidad de que una tupla arbitraria en D pertenece a la clase C_i, estimada por $\frac{|C_i,D|}{|D|}$ Información esperada (entropía) necesaria para clasificar una tupla en D, se tiene que la ganancia de de la información se expresa como [31]:

$$\boldsymbol{Info(D) = -\sum_{i=1}^{m} P_i \log_2(P_i)} \qquad (\mathbf{2})$$

3.6.3. Métodos de Balanceo de datos

El balanceo de datos evita sobre entrenamiento con una clase mayoritaria favoreciendo la clasificación de instancias de clase minoritaria (que puede contener información desconocida relevante) sin afectar la precisión de predicción para las clases mayoritarias. Tal es el caso de datos provenientes de análisis a muchos pacientes, en donde ciertas enfermedades de baja frecuencia pueden pasar desapercibidas en comparación con la gran cantidad de registros de enfermedades comunes.

Algunas aproximaciones para realizar esta etapa de optimización a la clasificación, buscan modificar directamente los algoritmos de clasificación como en [32], no obstante los desarrollos expuestos en [33] y [34] muestran dos iniciativas adecuadas para aplicaciones de pre-procesamiento con sobre muestreo y sub muestreo.

- **Sobre-muestreo *(Oversampling)***

Su objetivo es añadir muestras a la clase minoritaria tal que la cantidad de elementos de clase mayoritaria y minoritaria este dentro de un rango aceptable para considerar los datos como balanceados. Los algoritmos representativos son SMOTE y REMUESTREO [33].En esta investigación se utiliza SMOTE por su enfoque matemático dado que el remuestreo va en función del azar.

SMOTE: (*Syntetic Minority Over-sampling Technique)*: Propuesto en [35] hoy por hoy es uno de los algoritmos más conocidos como técnica de sobremuestreo ya que presenta un buen rendimiento según [36]. Se ejecuta sobre un conjunto *P* de instancias minoritarias creando *n* muestras sintéticas de cada instancia x_i del conjunto *P*. La instancia sintética se crea teniendo en cuenta la instancia minoritaria y sus vecinos más cercanos.

El procedimiento es el siguiente: Sea $\widehat{x_i}$ una instancia aleatoria seleccionada entre los *k* vecinos más cercanos de x_i, δ un número aleatorio entre 0 y 1 y $n = {}^{b}/_{100}$ donde b es un parámetro que define el porcentaje de sobre muestreo requerido para balancear el conjunto de datos. se repite *n* veces para cada instancia x_i como el ejemplo que muestra la Figura 3 para *k*=5.

La ecuación:

$$\boldsymbol{x_{sintética} = x_i + (\widehat{x_i} - x_i)\delta} \text{ .} \qquad (\mathbf{3})$$

- **Sub-muestreo *(Undersampling)***

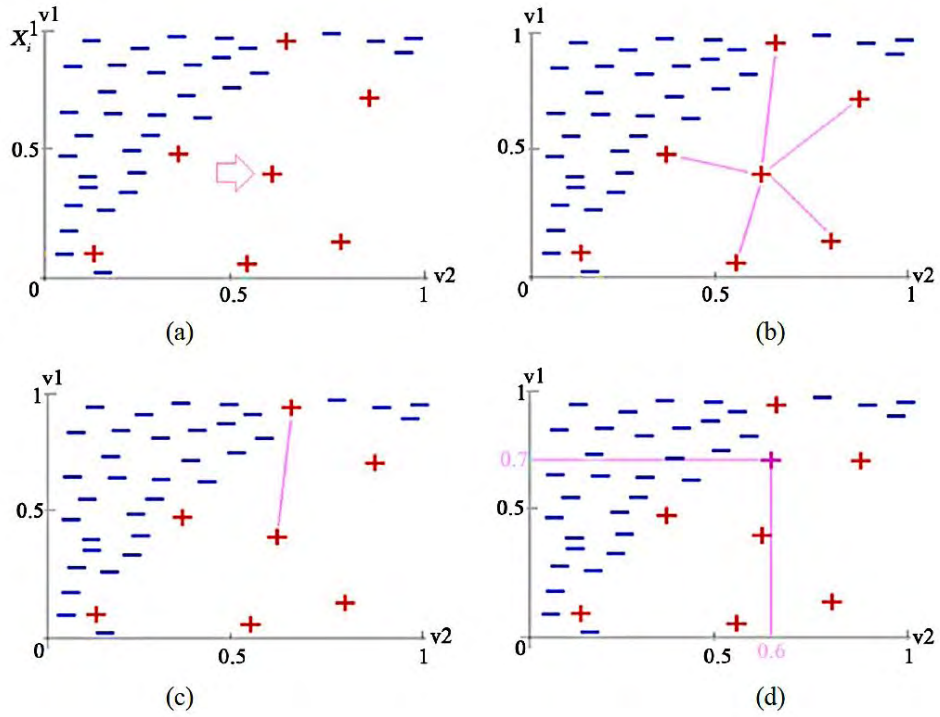
Su objetivo es disminuir la cantidad de muestras de clase mayoritaria de tal manera que la cantidad de elementos presentes en clases mayoritarias y minoritarias se encuentre dentro de un rango aceptable para ser consideradas balanceadas. En esta investigación se utiliza KNN-Und porque limpia la superficie de decisión eliminando ejemplos ruidosos.

- **KNN-U:**

Tal y como se describe en [34] este método remueve instancias de clases mayoritarias basado en sus *k* vecinos más cercanos para un sub conjunto *N* acorde los siguientes tres pasos:

- Primero se obtiene los *k* vecinos más cercanos para $x_i \in N$; segundo, x_i es removido si la cuenta de sus vecinos es mayor o igual a *t*; tercero, el proceso se repite para cada instancia mayoritaria del subconjunto *N.*

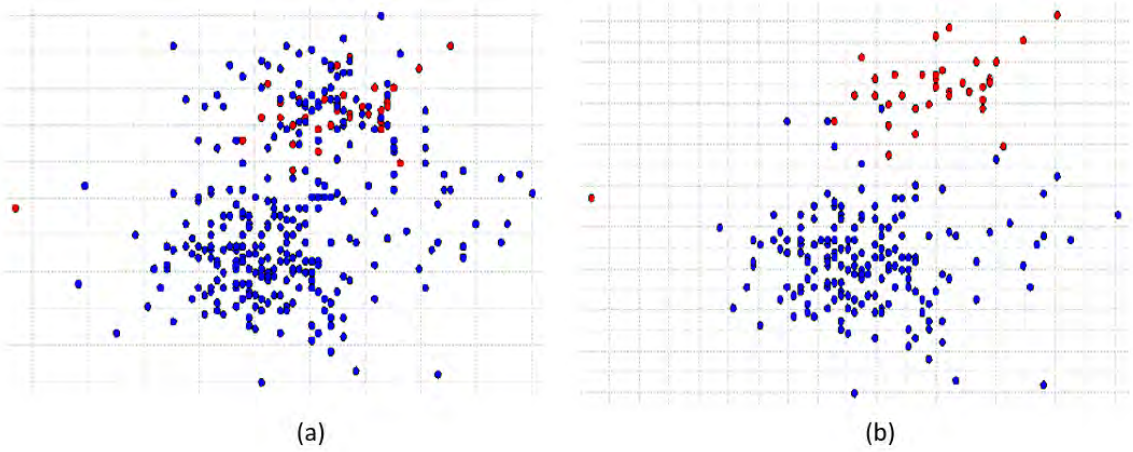
Figura 3. Proceso SMOTE con 5 vecinos cercanos. Dónde: (a) Conjunto de datos no balanceado, con instancias mayoritarias (-) y minoritarias (+), se selecciona la instancia x_i; (b) Las 5 instancias más cercanas del vecindario x_i son seleccionadas; (c) $\hat{x}_i$ es seleccionado aleatoriamente de entre los 5 vecinos; (d) Se crea una instancia sintética con los valores aleatorios de v1 y v2 entre x_i y $\hat{x}_i$. **Fuente: [34]**



- El parámetro t define la mínima cuenta de vecinos alrededor de x_i que pertenecen al conjunto minoritario *P*. Si la cuenta es mayor o igual a *t*, la instancia x_i será quitada del conjunto de entrenamiento.

- Según [34] el valor de t debe estar entre $1 \leq t \leq k$, entre menor sea t más agresivo es el sub-muestreo. La Figura 4 muestra como este algoritmo balancea los datos permitiendo una mejor identificación de las clases presentes y evitando que las clases minoritarias sean clasificadas como parte de la clase mayoritaria.

Figura 4. Sub-muestreo KNN-Undersampling. Donde (a) Conjunto de datos no balanceados, clase mayoritaria color azul, clase minoritaria color rojo; (b) Conjunto de datos balanceados luego de aplicar el algoritmo KNN-Und. **Fuente**: **[34]**.



Para el desarrollo multi-clase se toman distintos conjuntos negativos *N* que contienen las instancias de las clases mayoritarias, aplicando el desarrollo existente en la plataforma Weka [37].

3.7. CLASIFICADORES Y ESTIMADORES DE PROBABILIDAD

3.7.1. Máquinas de Soporte Vectorial (SVM)

En esta sección se describe el fundamento teórico de la máquina de soporte vectorial SVM (por sus siglas en inglés) como clasificador y estimador de probabilidad tal como lo expresa el autor en [38].
La propuesta de esta investigación para su inclusión dentro del sistema CBR se detallan en la Sección 4.2.

- Clasificador Bi-Clase y Multi-Clase

i) Clasificación binaria con SVMs

La máquina de soporte vectorial busca un hiperplano óptimo que separe las observaciones pertenecientes a dos clases, en un espacio de características de

mayor dimensión. Para ello resuelve el siguiente problema de optimización:

- Sea $\{(x_1, y_1),\ldots (x_n, y_n)\}$ un conjunto de datos, donde $\boldsymbol{x_i}$ es un vector de características de dimensión A y y_i el vector de respectivas etiquetas.

- Sea $\boldsymbol{w^T x} + b$ el hiperplano donde $w \in \mathbb{R}^A$ es un vector normal al hiperplano y b representa un término de sesgo. Juntos definen la dirección y traslación del hiperplano [39].

- Sea la función kernel $k(\boldsymbol{x}, \boldsymbol{x'}) := (\varphi(\boldsymbol{x}).\varphi(\boldsymbol{x'}))$ donde $\varphi: \boldsymbol{X} \rightarrow F$, es el mapeo definido para trasponer un vector de entrada en el Espacio de Característica [6].

De esta manera se busca:

$$\mathbf{min}\,\frac{\mathbf{1}}{\mathbf{2}}\sum_{\boldsymbol{i=1}}^{\boldsymbol{A}} \boldsymbol{w_i}^{\mathbf{2}} + \boldsymbol{C}\sum_{\boldsymbol{i=1}}^{\boldsymbol{n}} \boldsymbol{\xi_i} \tag{4}$$

$$\text{Sujeto a } \boldsymbol{y_i}(\boldsymbol{w}^T\boldsymbol{x_i} + b) \geq 1 - \boldsymbol{\xi_i}\ \forall i\ \in \{1, \ldots, n\}\ ,$$
$$\boldsymbol{\xi_i} \geq 0\ \forall i\ \in \{1 \ldots, n\}$$

donde ξ_i corresponde a los errores de entrenamiento mientras se obtiene el máximo margen de hiperplano ajustado por el parámetro C.

Para este fin existe la formulación dual conocida como Wolfe dual formulación [38]. Aquí, luego de determinar los parámetros óptimos para α se obtiene las salidas
continúas representadas por:

$$\boldsymbol{g(x_i)} = \sum_{\boldsymbol{i=1}}^{\boldsymbol{n}} \boldsymbol{\alpha_i\, y_i.\, k(x_i, x_j)} + \boldsymbol{b}\quad , \tag{5}$$

donde la clasificación del $j - esimo$ elemento es dada por $f(\boldsymbol{x_j}) = sign\left(g(\boldsymbol{x_j})\right)$.

ii) Clasificación multiclase con SVMs

Los desarrollos expuestos en [40] y [41] definen cinco aproximaciones de extensión de SVM bi-clase para la resolución de problemas de clases múltiples.

1) Uno-vs-Todos (OVA) [42] : Básicamente OVA tiene como objetivo entrenar K SVMs, uno por cada clase definidos por la etiqueta del $j - esimo$ objeto $y_i \in \{1 \ldots, K\}$.

2) Uno-vs-Uno (OVO) [43]: Por su parte, OVO debe ser entrenado para separar todas las clases una de la otra y por tanto es necesario determinar el etiquetado a la salida.
3) Mitad-vs-Mitad (HVH) [41]: La metodología HVH divide recursivamente el conjunto de K clases en dos sub conjuntos que pueden ser representados como un problema de agrupamiento de jerarquías.

4) Función objetivo multi-clase [43]

5) Codificación de corrección de error a la salida (ECOC) [44]: Finalmente ECOC define una estructura especial para resolver problemas de clasificación multi-clase a partir de una clasificación binaria dada.

Cabe resaltar que en esta investigación se profundiza el método OVA como se verá más adelante en la Sección 4.2

- Estimador de Probabilidad

Otro aspecto importante en el tratamiento de la teoría del SVM es la parte de estimación de probabilidad, para esta se desarrollan las partes de estimación binaria y multi-clase :

i) **Estimación de probabilidad para SVM binario**

Los desarrollos en [44] [45] y [46] extienden la salida (2) a una estimación de probabilidad bajo la expresión $\boldsymbol{p_{ij}} = P(y = i|\boldsymbol{x_i}), i = \{1,2, \dots, K\}, j = \{1,2, \dots, n\}$. Por su parte el autor en [45] propone una expresión analítica para la salida de probabilidad basada en una función sigmoidea, así:

$$(\widehat{\boldsymbol{P}}(\boldsymbol{y} = \boldsymbol{i}|\boldsymbol{g}(\boldsymbol{x_i})) \coloneqq \boldsymbol{p_{ij}} = \frac{\mathbf{1}}{\mathbf{1}+\boldsymbol{e}^{\boldsymbol{M}.\boldsymbol{g}(\boldsymbol{x_i})+\boldsymbol{B}}} \quad . \tag{6}$$

Consiste en mapear las distancias de cada objeto hacia el hiperplano que los divide (resultados $g(\boldsymbol{x})$ del SVM) en $\hat{P}(c|g(\boldsymbol{x}))$. Las constantes M y B pueden obtenerse aplicando el método de máxima verosimilitud [47].

i) **Estimación de probabilidad para SVMs multi-clase**

Para un número K de clases sea x_j el vector de atributos para la j^{th} observación y $y_j \in \{1 \dots, K\}$ su respectiva etiqueta de clase. La expresión $\boldsymbol{p_{ij}} = P(\boldsymbol{y_j} = i|\boldsymbol{x_j})$ para $i \in \{1,2, \dots, K\}$ denota la probabilidad de pertenencia que tiene esta observación de

pertenecer en la clase i y evidentemente satisface $\sum ip_{ij} = 1$.

Supóngase una observación x y su etiqueta de clase y. La probabilidad que tiene esta observación de pertenecer a la clase i se escribe $p_i = P(y = i|x)$. Se han propuesto varias aproximaciones para determinar esta probabilidad, destaca
el desarrollo expuesto en [48] donde se estima p_{ik} a partir de dos métodos cuya característica en común es asumir la estimación de probabilidad de clase "*pairwise*" r_{ik} de $P(y = i|y = i\ o\ k, x)$ que se obtiene del SVM binario [49].

La primera metodología basada en la teoría de la cadena de Markov (WLW1) determina la probabilidad $p = (p_1, \dots, p_k)$ acorde el siguiente problema lineal

$$\boldsymbol{p_i = \sum_{k:k\neq i}\left(\frac{p_i+p_k}{K-1}\right)p_{ik}, \forall i} \quad , \tag{7}$$

Sujeto a $\sum_{i=1}^{K} p_i = 1, p_i \geq 0, \forall i$.

La regla de decisión está dada por $\delta_1 = \arg\max_i [p_i]$

Existe una única solución que según [48] puede obtenerse fácilmente por eliminación Gaussiana o solución estacionaria de la cadena de Markov resolviendo:

$$\boldsymbol{Q_p = p, \sum_{i=1}^{K} p_i = 1\,, p_i \geq 0, \forall i, Q} = \begin{cases} \frac{r_{ik}}{K+1} & \boldsymbol{if\ i \neq k} \\ \sum_{s:s\neq i}\frac{r_{is}}{K+1} & \boldsymbol{en\ otro\ caso} \end{cases} . \tag{8}$$

La segunda metodología (WLW2) propone un problema de optimización como sigue:

$$\min_{p} \ \sum_{i=1}^{K}\sum_{k:k\neq i}(r_{ki}p_i - r_{ik}p_k)^2 \ , \tag{9}$$

Sujeto a $\sum_{i=1}^{K} p_i = 1, p_i \geq 0, \forall i$

Cuya regla de decisión es la misma que en (7). Por otra parte, cabe mencionar el desarrollo propuesto en [50] cuyo objetivo es minimizar la distancia Kullback-Leibler entre r_{ik} y v_{ik} donde $v_{ik} = p_i/(p_i + p_k)$, haciendo:

$$\boldsymbol{L(\mathbf{p}) = \sum_{i<k} q_{ik}(\log\frac{r_{ik}}{v_{ik}} + (1 - r_{ik})\log\frac{1-r_{ik}}{1-v_{ik}})} \ , \tag{10}$$

donde q_{ik} denota la cantidad de observaciones que pertenecen a la clase $i-esimo$ o $k-esimo$.

Un algoritmo iterativo se encarga de calcular los valores tales que la distancia Kullback-Leibler [51] es minimizada. Finalmente, cabe resaltar la aproximación propuesta en por su simplicidad, ya que los valores de probabilidad son aproximados usando los coeficientes r_{ik}, así:

$$\boldsymbol{p_i} = \frac{\mathbf{1}}{\sum_{\boldsymbol{k}:\boldsymbol{k}\neq\boldsymbol{i}} \frac{\mathbf{1}}{\boldsymbol{r_{ik}}} - (\boldsymbol{K}-\mathbf{2})} \,. \tag{11}$$

3.7.2. Clasificador y Estimador basado en densidades usando el método de Parzen

i) Clasificación

Tal y como se expone en [6] se da a conocer la clasificación de máxima esperanza dentro de un modelo iterativo genérico.

- Modelo iterativo genérico: Es una forma general e iterativa de clasificar densidades estudiando la proporción de pertenencia de una instancia a una clase y la influencia de cada instancia en el cálculo de las siguientes iteraciones.
  Esta proporción se calcula con una función de membresía $m(l|x_i)$ que se lee: grado de pertenencia x_i a la clase l. Esta proporción es definida positiva y la pertenencia absoluta es 1 por lo que m satiface

$$\boldsymbol{m(l|x_i)} \geq \mathbf{0} \quad \text{y} \quad \sum_{\boldsymbol{l}=\mathbf{1}}^{\boldsymbol{C}} \boldsymbol{m(l|x_i)} = \mathbf{1} \;, \tag{12}$$

  y de influencia en el cálculo de las actualizaciones y por tanto un factor de ponderación de los datos x_i. Las funciones m y w se relacionan directamente con la naturaleza de la función objetivo.

  Así la actualización de clases para la iteración r se escribe como:

. $$\boldsymbol{l^r} = \frac{\sum_{\boldsymbol{t}=\mathbf{1}}^{\boldsymbol{N}} \boldsymbol{m(l^{(r-1)}|x_i)w(x_i)x_i}}{\boldsymbol{m(l^{(r-1)}/x_i)w(x_i)}}, \boldsymbol{l} \in \{\mathbf{1}, \dots, \boldsymbol{C}\} \tag{13}$$

Inspirada en la ecuación para obtener el centro de masa de un cuerpo. Cabe resaltar que esta función m se puede adaptar a cualquier función objetivo y que la actualización de clases se hace iterativamente, entonces este es un modelo genérico que como se verá puede aplicarse en la clasificación de

máxima esperanza Gaussiana.

- Clasificador de máxima esperanza Gaussiana: Tiene como función objetivo la combinación lineal de distribuciones gaussianas centradas en los valores medios de cada clase según:

$$\boldsymbol{f_{GEMC}(X, l) = -\sum_{i=1}^{N} \log\left(\sum_{l=1}^{C} p(x_i|l)p(l)\right)}, \quad (14)$$

donde $p(x_i|l)$ es la probabilidad de x_i porque se genera de una distribución gaussiana centrada en μ_l y $p(l)$ es la probabilidad a priori de la clase l.

Las funciones correspondientes a membresía y peso de cada elemento son respectivamente:

$$\boldsymbol{m_{GEMC}(q_j|x_i) = \frac{p(x_i|q_j)p(q_j)}{p(x_i)}} \quad \text{y} \quad \boldsymbol{W_{GEMC}(x_i) = 1}. \quad (15)$$

Si se considera $p(x_i)$ como evidencia, la función de membresía es un valor de probabilidad para la cual la regla de Bayes puede usarse, así:

$$\boldsymbol{p(x_i) = \sum_{l=1}^{C} p(x_i|l)p(l)}, \quad (16)$$

donde el factor $p(\boldsymbol{x_i}|l)$ se obtiene a partir de:

$$\boldsymbol{p(x_i|l) = f(x_i, \mu_l, \sum_l) = \frac{1}{\det(\sum_l)^{\left(\frac{1}{2}\right)}} (2\pi)^{-d/2} e^{-\frac{1}{2}(x_i-\mu_l)\sum_l{}^{-1}(x_i-\mu_l)^T}}, \quad (17)$$

y μ_l es la medida de la clase l, d es la dimensión, y $\sum_l$ es la matriz de covarianza.

ii) Estimación de Probabilidad

Consiste en la superposición de distribuciones gaussianas de una tamaño fijo *h* centradas en cada x_i [52]. Esta clasificación de Parzen PC también es conocida como DBC no paramétrico, donde el valor óptimo de *h* se puede obtener con validación cruzada.

En términos matemáticos la distribución de probabilidad aplicando Parzen es:

$$\boldsymbol{p(x) = \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{x-x_i}{h}\right)}, \quad (18)$$

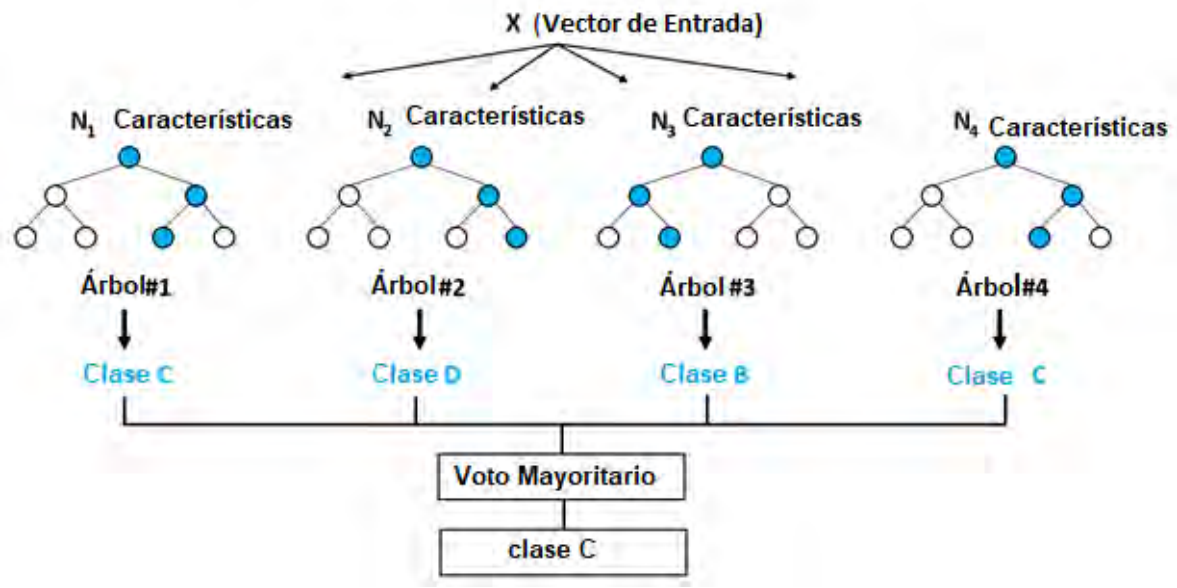
donde K es un kernel gaussiano que se define:

$$\boldsymbol{K(z) = \frac{1}{2\pi^{-d/2}} exp^{-\frac{1}{2}zz^T}}. \quad (19)$$

3.7.3. Random Forest

Una Random Forest consiste en una combinación de clasificadores donde cada clasificador contribuye con un voto único para la asignación de la clase más frecuente al vector de entrada. La idea central detrás de Random Forest es generar múltiples árboles pequeños de decisión a partir de subconjuntos aleatorios de los datos [53]. Cada uno de los árboles de decisión proporciona un clasificador que solo considera un subconjunto de los datos, este clasificador proporciona un voto de asignación de clase, el cual se suma al voto de los demás clasificadores, el voto de la mayoría clasifica una clase. Un ejemplo de esto se muestra en la Figura 5.

Figura 5. Esquema Random Forest. En donde hay un vector $\boldsymbol{X}$, con 4 subconjuntos de características, 4 árboles y 3 clases. El voto mayoritario lo obtiene la clase C.



Un aspecto importante es el hecho de poder combinar muchos clasificadores confiriendo a este clasificador algunas características especiales que la hacen sustancialmente diferente a los árboles de clasificación tradicionales (TC) y, por lo tanto, debe entenderse como un nuevo concepto de clasificadores. Entre algunas características se encuentran:

- Los datos pueden usarse más de una vez en el entrenamiento de clasificadores, mientras que otros pueden no ser utilizados. Por lo tanto, se logra una mayor estabilidad del clasificador, ya que lo hace más robusto cuando enfrenta ligeras variaciones en los datos de entrada y, al mismo tiempo, aumenta la precisión de la clasificación.

- Los árboles de un clasificador Random Forest crecen sin poda, lo que lo hace liviano, desde una perspectiva computacional. El diseño de árbol requiere elegir una medida de selección de atributo adecuada que maximice la diferencia entre clases. Hay muchas aproximaciones para seleccionar atributos que pueden usarse para inducción en árboles de decisión

- Finalmente, un Random Forest también puede producir una medida de proximidad entre cada par de casos. Para calcular la proximidad entre dos muestras de la misma clase, estecuenta el número de veces que dichas muestras aparecen en el mismo nodo terminal (es decir, la cantidad de etiquetas de árboles cada posible par de cajas de la misma clase con la misma regla de division). Una vez que se ha construido cada árbol y se calculan las proximidades son para cada par de casos, se normalizan dividiendo por número de árboles. Las proximidades son susceptibles de ser utilizadas en reemplazar datos faltantes y localizar valores atípicos (es decir, sitios mal etiquetados) en conjuntos de entrenamiento)

3.7.4. K-vecinos más cercanos.

i) Clasificación en KNN

Los métodos de vecinos más cercanos pertenecen a una clase de algoritmos no paramétricos conocidos como métodos prototipo [54]. Se distinguen de otros algoritmos de aprendizaje en el sentido de que están basados en memoria y no requieren ningún modelo para ajustarse. El principio detrás de los métodos vecinos más cercanos es encontrar un número de muestras de entrenamiento más cercanas en distancia a la nueva muestra, y luego deducir de ellas el valor de la variable de salida. Para la regresión, el k El algoritmo vecino más cercano [55] promedia los valores de salida de las k muestras de entrenamiento más cercanas, matemáticamente se expresa como [56]:

$$\boldsymbol{\delta}(\boldsymbol{x}) = \frac{\mathbf{1}}{\boldsymbol{K}} \sum_{(\boldsymbol{x_i},\boldsymbol{y_i})\epsilon \boldsymbol{NN}(\boldsymbol{x},\boldsymbol{\zeta},\boldsymbol{k})} \boldsymbol{y_i} \quad , \tag{20}$$

donde $NN(x,\zeta,k)$ es el k-vecino mas cercano de x en ζ. Para la clasificación, el

procedimiento es el mismo excepto que el valor de salida predicho se calcula como la clase mayoritaria entre los k vecinos más cercanos [56]:

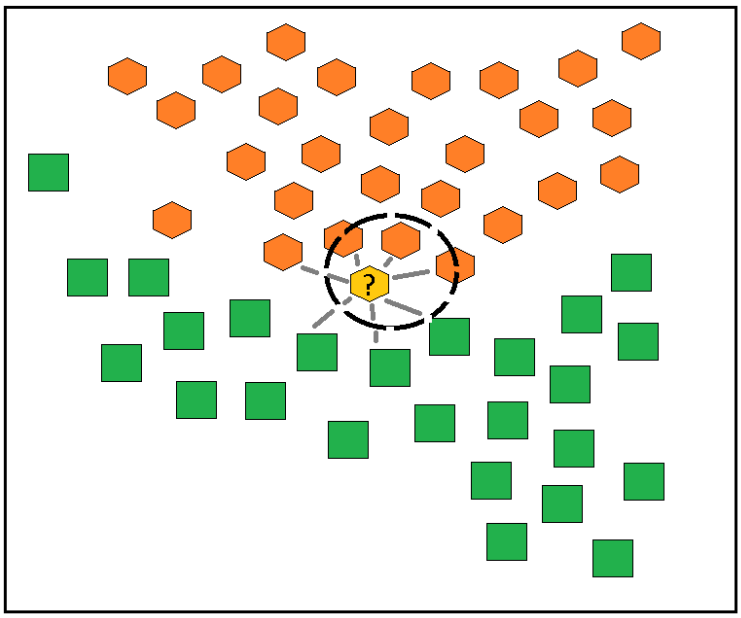
$$\boldsymbol{\delta(x) = \arg\max_{c \epsilon y} \sum_{(x_i, y_i) \epsilon NN(x,\zeta,k)} 1(y_i = C)} \ . \quad (21)$$

En general, la función de distancia utilizada para identificar los k vecinos más cercanos puede ser cualquier métrica como, la distancia de Manhattan, la distancia de Chebychev, la distancia del coseno, la distancia de Mahalanobis, pero la distancia Euclidiana estándar es la opción más común. La distancia Euclidiana sirve para calcular la distancia entre dos puntos en cualquier tipo de espacio (cualquier número de dimensiones), para hallar dicha longitud de segmento la distancia euclidiana entre dos puntos se define como:

$$\boldsymbol{d(X_1, X_2) = \sqrt{\sum_{i=1}^{d} (x_{1i} - x_{2i})^2}} \ . \quad (22)$$

Este proceso de utilizar el KNN con medidas de métrica Euclidiana se representa gráficamente en la Figura 6.

Figura 6. Método de los K-vecinos más cercanos. Se calcula los vecinos cercanos de la muestra por medio de la distancia euclidiana. El pentágono amarillo representa la nueva muestra y los conjuntos de cada clase están representados por los cuadros verdes y hexágonos naranjas. La circunferencia punteada encierra a los casos similares recuperados para la nueva muestra. **Fuente: [34]**



A pesar de su simplicidad, los métodos KNN generalmente dan buenos resultados en la práctica. Ellos a menudo son exitosos en situaciones de clasificación donde el límite de decisión es muy irregular.

ii) Estimación de probabilidad en KNN

La clasificación puede ser basada en la estimación de densidades para cada una de las clases. Por un grupo de vectores aleatorios analizados [56]:

$$\{x_1, x_2, \ldots., x_n\} \rightarrow p(x)$$

La probabilidad de que un vector x, escogido de $p(x)$, caiga en la región R en el espacio de muestras es:

$$\boldsymbol{P = \int_R p(x')\, dx'} \quad . \tag{23}$$

Cuando N vectores son observados en la distribución, la probabilidad de que exactamente k de ellos caigan en R es:

$$\boldsymbol{P(k) = \binom{N}{k} P^k (1-P)^{N-k}} \quad , \tag{24}$$

de acuerdo a las propiedades de distribución binomial:

$$\boldsymbol{E\left[\frac{k}{N}\right] = P} \qquad \boldsymbol{Var\left[\frac{k}{N}\right] = E\left[\left(\frac{k}{N} - P\right)^2\right] = \frac{P(1-P)}{N}} \quad , \tag{25}$$

- Cuando N se incrementa, la varianza disminuye. k/N se convierte en un buen estimador de P
- Cuando muestras lo suficientemente grandes están disponibles, un R pequeño puede ser usado tal que $p(x)$ varié muy poco dentro del mismo. Sea V el volumen

$$\boldsymbol{P = \int_R p(x')\, dx' \cong p(x) V} \ , \tag{26}$$

como también:

$$\boldsymbol{P \cong \frac{k}{N}} \quad , \tag{27}$$

entonces:

$$\boldsymbol{p(x) \cong \frac{k}{NV}} \ , \tag{28}$$

- Cuando N se incrementa y V decrece, la estimación se vuelve más acertada.

Consideraciones asíntotas:

- Se construye R_1, R_2, R_3 ... con un número grande de muestras
- Sea $V_n$los volúmenes, K_n el número de muestras incluidas y $p_n(x)$ el n estimado de $p(x)$

Tres condiciones son necesarias para que $p_n(x)$ converja a $p(x)$:

$$\lim_{n\to\infty} V_n = 0$$
$$\lim_{n\to\infty} K_n = \infty$$
$$\lim_{n\to\infty} k_n/n = 0$$

Para obtener una secuencia R_1, R_2, R_3 ... dos caminos o formas existen:

1. Especificar V_n com una función de n, por ejemplo $V_n = 1/\sqrt{n}$, Mostrar que K_n y K_n/n cumplen las tres condiciones (esta es la estimación de densidad de kernel)

2. Especificar K_n como una función de n, por ejemplo $K_n = \sqrt{n}$, Usar $V_n$tal que K_n muestras se encuentre en las vecindades, Mostrar que V_n cumple las condiciones (este es el método K_n vecinos más cercanos).

Para estimar $p(x)$, creamos una celda de x hasta que K_n muestras son capturadas. K_n es una función de n. La muestra son los K_n vecinos más cercanos de x.

La estimación de densidad es:

$$p_n(x) = \frac{K_n/n}{V_n} \text{ , Si } \quad K_n = \sqrt{n} \quad , \qquad \textbf{(29)}$$

entonces**:**

$$\boldsymbol{V_n \approx \frac{1}{\left(\sqrt{n}p(x)\right)}} \quad , \qquad (30)$$

$$\boldsymbol{V_n \approx V_1/\sqrt{n}} \quad , \qquad (31)$$

donde K_1 es determinado por la naturaleza de los datos.

Aunque KNN es muy similar a las ventanas de Parzen, n términos de clasificación, es usado en una forma más simple: estimando directamente la probabilidad posterior $P(\omega_i|x)$ de n muestras etiquetadas. Una celda con volumen V captura k

muestras, K_1en la clase 1; K_2en la clase 2 …

La probabilidad $p(x, \omega_i)$ es estimada por:

$$\boldsymbol{p_n(x, \omega_i) = \frac{K_i/n}{V}} \quad , \tag{32}$$

entonces

$$\boldsymbol{P_n(\omega_i|x) = \frac{p_n(x,\omega_i)}{\sum_{j=1}^{C} p_n(x,\omega_j)} = \frac{k_i}{k}} \quad . \tag{33}$$

3.7.5. Clasificador Bayesiano Ingenuo.

El algoritmo ingenuo de Bayes emplea una versión simplificada de la fórmula de Bayes para decidir a qué clase pertenece una instancia nueva. La probabilidad posterior de cada clase se calcula, dados los valores característicos presentes en la instancia; a la instancia se le asigna la clase con la probabilidad más alta. La ecuación 35 muestra la fórmula ingenua de Bayes, que supone que los valores de las características son estadísticamente independientes dentro de cada clase [56].

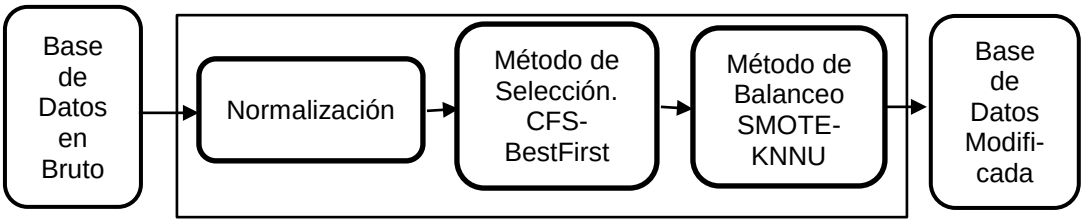
$$\boldsymbol{p(C_i|v_1, v_2, \ldots\ldots, v_n) = \frac{p(C_i)\prod_{j=1}^{n} p(v_i|C_i)}{p(v_1, v_2, \ldots\ldots, v_n)}} \tag{34}$$

El lado izquierdo de la Ecuación X es la probabilidad posterior de la clase C_i dada la característica valores, $\langle v_1, v_2, \ldots\ldots, v_n \rangle$ observado en la instancia que se va a clasificar. El denominador del lado derecho de la ecuación se omite a menudo porque es una constante que se calcula fácilmente si se requiere que las probabilidades posteriores de las clases sumen a uno. Aprender con el clasificador ingenuo de Bayes es sencillo e implica simplemente estimar las probabilidades en el lado derecho de la Ecuación 35 de las instancias de entrenamiento. El resultado es un resumen probabilístico para cada una de las posibles clases. Si hay características numéricas, es una práctica común asumir una distribución normal; una vez más, los parámetros necesarios se estiman a partir de los datos de entrenamiento.

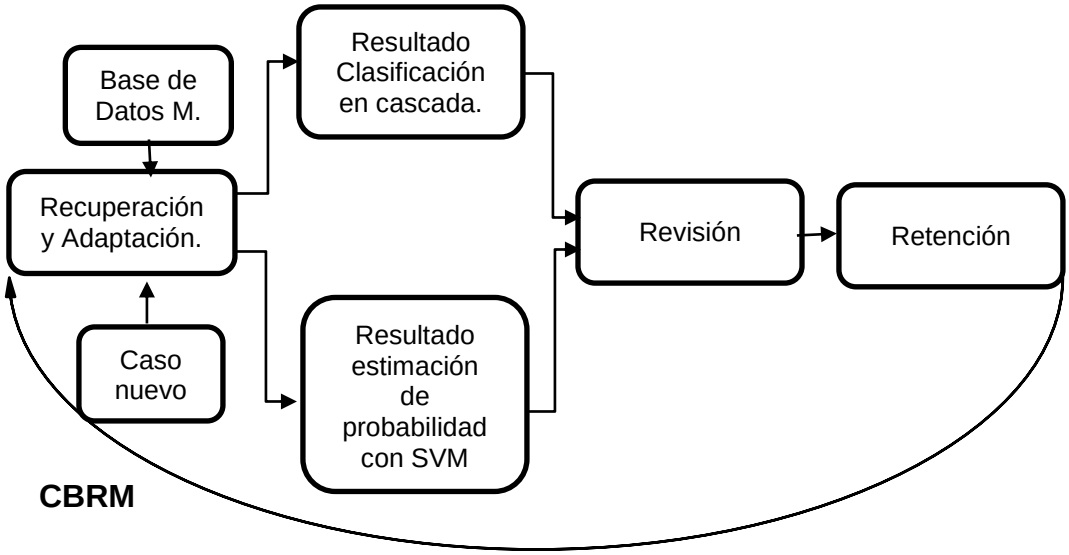
4. METODOLOGÍA

En esta sección se da a conocer la propuesta de metodología para sistemas CBR. La propuesta está divida en 3 partes importantes. La primera tiene que ver con un método de selección sugerido para realizar el proceso de disminución de atributos de las bases de datos, el segundo enfocado en un método de balanceo donde se unieron dos algoritmos comúnmente utilizados en este proceso, estas dos propuestas en conjunto forman parte de la etapa de pre-procesamiento. La tercera parte de esta propuesta es utilizar la clasificación en cascada o secuencial en conjunto con la probabilidad de estimadores SVM, para que en una etapa de recuperación y adaptación conjunta se provean resultados más óptimos en estos sistemas. La metodología propuesta se muestra en la Figura 7.

Figura 2. Metodología desarrollo CBR con etapa de adaptación y recuperación mejorada. Se muestra las etapas que lo conforman iniciando con el pre-procesamiento de los datos. La clasificación de cascada y la estimación de probabilidades son vinculadas a las etapas de adaptación y recuperación fusionadas



Pre-Procesamiento



4.1. PRE-PROCESAMIENTO

Este proceso se lleva en 3 etapas, descritas en las sub-secciones de Normalización, Método de Selección de características y Método de Balanceo.

➢ Método de selección de características

El método de selección propuesto se divide en dos partes, el algoritmo de selección de atributos CFS junto con el método de búsqueda Best First. Como se observa en la sección 3.6.2 de marco teórico los métodos de selección utilizados son:

- CFS-SubsetEval-Best First
- InfoGainAttribute - Ranker

Para evaluar su desempeño se evalúa el comportamiento de distintos clasificadores que operan a partir de la base de datos transformada como se observa en la Sección 6 (Resultados). De esta manera se determinaron ventajas y desventajas de cada metodología expuestas en la Tabla 1

Tabla 1. Métodos de selección de características

	Ventajas	*Desventajas*
CFS-Best First	Disminuye el tiempo de ejecución. Asegura una óptima clasificación. Entrega un numero de atributos destacados, para seleccionar.	En algunas bases de datos puede, que se ignoren algunos atributos relevantes, debido a la naturaleza heurística que define esta técnica.
InfoGainAtributte - Ranker	Disminuye el tiempo de ejecución. Asegura una óptima clasificación.	La información de pesos es muy general, se debe experimentar con diferentes cantidades de atributos para tener la mejor elección.
Sin ningún Método	-Se trabaja sin hacer ningún proceso adicional. -La base de datos provee la información de forma natural.	-Alto tiempo de ejecución -Problema de sobre-entrenamiento en clase mayoritaria - Propagación de errores

Según estas observaciones se aplica el método de selección CFS- Best First que determina los resultados más óptimos y muestra ser el que mayores ventajas

presenta frente a esta investigación.

- Método de Balanceo de datos

Tal como en el caso anterior se realizaron pruebas sobre los clasificadores a partir de los datos transformados con los métodos SMOTE para sobre-muestreo y KNN para sub-muestreo. La Sección 6 (Resultados) da a conocer que ambas metodologías ofrecen resultados favorables en clasificación, no obstante, cabe aclarar que como propuesta de esta investigación se estudia el comportamiento de las técnicas en cascada y buscando tomar las ventajas de uno y otro, se fusionan en un algoritmo propuesto denominado SMOTE-KNN-U, el cual opera según el siguiente proceso:

Algoritmo Hibrido KNNU-SMOTE

Entradas: Cantidad de muestras sintéticas a generar N, Datos, Clases, cantidad de vecinos cercanos a tomar M.
Salidas: base de datos modificada FBase.

```
1.  Xnew1[ ]: clase mayoritaria
2.  para l=1 hasta Y cantidad de clases
3.          Tam(l): cantidad de casos por clase dentro de Datos.
4.  Fin para
5.  Se excluye la clase con mayor cantidad de casos.
6.  para k=1 hasta Y-1
7.          synthetic[]: arreglo que almacenara las muestras sintéticas creadas.
8.          newindex: aumentara cada vez que se genere un nuevo atributo sintético
9.          para i=1 hasta Tam(k)
10.                 N1=N reinicio el contador para cada caso
11.                 cercain[ ]=M vecinos más cercanos al caso a analizar
12.                 Sample[ ]=[Datos(k) Case(k)]
13.                 mientras N1 ~=0
14.                         para attr=1 hasta numattr total de características
15.                         dif = Sample[cercain[nn]][attr] – Sample[i][attr]
16.                          gap = aleatorio entre 0 y 1
17.                         Synthetic[newindex][attr] = Sample[i][attr] + gap*dif
18.                         Fin para
19.                 newindex++
20.                 N1=N1-1
21.                 Fin mientras
22.         Fin para
23. Fin para
24. para k= hasta cantidad de casos en la mayoritaria
25.         para i=1 hasta cantidad de datos de la base
26.                 Xnew1:
27.                 Se obtienen las distancias de xnew1 con los demás casos
28.                         Dist=(norm(xnew1(i,:)-(Datos(k,:))))^2;
29.                 se guardan las distancias encontradas
30.         Fin para
```

31. Se encuentran las clases pertenecientes a la mayoritaria cercanas al caso y se
eliminan
32. **Fin para**
33. Fbase[]:Se unen la clase mayoritaria reducida y las clases minoritarias aumentadas

Este algoritmo hibrido presenta muy buenos resultados en torno a la disminución del error, puesto que la interacción de algoritmos de sobre-muestreo y sub-muestreo permite obtener las ventajas de estos dos tipos de balanceo. Una desventaja es el tiempo de duración de procesamiento del algoritmo, pero que en nuestra metodología no afecta de ninguna forma al sistema CBR, ya que el pre-procesamiento es independiente al ciclo y por lo tanto solo se ve favorecido por la buena representación de los datos entregada por este algoritmo de balanceo.

4.2. ALGORITMO DE CBR CON ETAPAS DE ADAPTACIÓN Y RECUPERACION MEJORADAS

En esta etapa se realiza unas modificaciones al algoritmo de CBR multi-clase mostrado en el Anexo 6. Con el objetivo de separar los integrantes de una clase de las bases de datos, se dividen dos grupos: el primero integrado por dicha clase y el otro por el conjunto de las sobrantes, a los que se les da el valor de 0 y 1 respectivamente, con el propósito de realizar una clasificación Bi-clase por medio del clasificador Random Forest (sección 3.7.3). Con el resultado de la clasificación Bi-clase, el grupo clasificado como 0 pasa (previo a devolver a su clase original a los integrantes del grupo) a una combinación de clasificadores multi-clase en cascada integrado por los clasificadores Random Forest, SVM y Parzen (secciones 3.7.3, 3.7.1, 3.7.2), dando como resultado el diagnostico ya sea por el resultado de la clasificación bi-Clase o multi-clase. Este resultado puede ser tomado como final por el usuario que puede añadir un nuevo caso con esta clase. O puede apoyarse tomando el resultado de una etapa estimación de probabilidad utilizando SVM multiclase (sección 3.7.1), el cual provee un porcentaje de pertenencia a cada clase. Estos resultados dan una mayor confiabilidad en la revisión para la toma de decisiones por parte del especialista, que puede decidir retener o no el caso evaluado.

CBRM

1. Car [ ][ ]: características de la base de datos
2. Cla [ ]: clases de la base de datos
3. Clab [ ]:Clases para entrenamiento clasificador Bi-clase
4. División de porcentaje para entrenamiento y prueba de los clasificadores
5. **si** k==clase de interés a remover **entonces**

6. Cla[k]=1
7. **sino**
8. Cla[k]=0
9. **Fin si**
10. Unión de características con clases
11. Total=[Car[ ][ ] Clab[ ]]

Adaptación y Recuperación.

12. Entrenamiento del clasificador ***Random Forest*** bi-clase
13. Remoción de la clase de interés de la base de datos junto con sus características
14. Total=[Car [ ][ ] Cla [ ]]
15. Entrenamiento del clasificador ***Random Forest – SVM- Parzen***
16. Estimación del error

Adición de un nuevo caso

17. Búsqueda de los vecinos cercanos al caso
18. **mientras** ban==0
19. clasib: etiqueta dada por el clasificador Bi-clase
20. clasim: etiqueta dada por el clasificador Multi-clase
21. clasfinal: etiqueta final asignada al nuevo caso
22. **si** clasfi==0 **entonces**
23. clasfinal=clase removida
24. **sino**
25. clasfinal=clasim
26. **fin si**

Estimación Probabilidad

27. **Para** i=1 **hasta** cantidad de clases
28. p_i: probabilidad de pertenencia a diferentes clases
29. $p_i = \frac{1}{\sum_{k:k\neq i}\frac{1}{r_{ik}}-(K-2)}$
30. **Fin para**

Fin de Adaptación y Recuperación.

Interfaz Usuario (Revisión y Retención)

31. aux: entrada dada por el usuario
32. ¿Que desea hacer?
    i. Correr el código otra vez
    ii. Analizar otro caso
    iii. Guardar
    iv. Cerrar
33. **si** aux==1 **entonces**
34. ban=0
35. **Fin si**

```
36.     si aux==1 entonces
37.             desea guardar el caso anterior
                        1. Si
                        2. No
38.             ban=0
39.     Fin si
40.     si aux==1 entonces
41.             guarda el nuevo caso
42.             ban=0
43.     Fin si
44.     si aux==1 entonces
45.             ban=0
46.     Fin si
47. Fin mientras
```

Los métodos de clasificación y estimación propuestos son el resultado de diversos experimentos expuestos en la sección siguiente, los cuales avalan el algoritmo propuesto. Dichos métodos demostraron ser la mejor forma de mejorar la precisión y exactitud del sistema CBR para las bases de datos objeto de estudio.

5. MARCO EXPERIMENTAL

En el desarrollo de los experimentos de esta sección, se utiliza las bases de datos disponibles públicamente de la *UCI Machine Learning Repository* de la Universidad de California [49]. Se evalúa el desempeño de los diferentes métodos de selección, balanceo y las combinaciones de clasificadores multi-clase en secuencia, con métricas de desempeño como el promedio, matriz de confusión, curvas ROC y gráficos de cajas, con el objetivo de identificar los mejores métodos e integrarlo a la etapa de adaptación y recuperación del CBR.

5.1. BASES DE DATOS

La primera base de datos es llamada Hipotiroidismo, en la Tabla 2 se pueden observar los atributos con sus respectivos valores, mientras que la cantidad de registros por cada clase se exponen en la Tabla 3.

Tabla 2. Información de atributos de la base de datos de Hipotiroidismo

Características:	*Valor de la variable.*	
1.age:	Valor continuo	
2.sex:	M	F
3.on thyroxine:	F	T
4.query on thyroxine:	F	T
5.on antithyroid medication:	F	T
6.sick:	F	T
7.pregnant:	F	T
8.thyroid surgery:	F	T
9.I131 treatment:	F	T
10.query hypothyroid:	F	T
11.query hyperthyroid:	F	T
12.lithium:	F	T
13.goitre:	F	T
14.tumor:	F	T
15.hypopituitary:	F	T
16.psych:	F	T
17.TSH measured:	F	T
18.TSH:	Valor Continuo	
19.T3 measured:	F	T
20.T3:	Valor Continuo	
21.TT4 measured:	F	T
22.TT4:	Valor Continuo	
23.T4U measured:	F	T

24.T4U:	Valor Continuo	
25.FTI measured:	F	T
26.FTI:	Valor Continuo	
27.TBG measured:	F	T
28.TBG:	Valor Continuo	
29.referral source:	WEST, STMW, SVHC, SVI, SVHD, other.	

Tabla 3. Cantidad de datos por clase para base de datos de Hipotiroidismo

Clase	*Diagnostico*	*Cantidad*
1	Hypothyroid	1790
2	Primary Hypothyroid	102
3	Negative	50

La segunda base de datos es de Dermatología, las características son expuestas en la Tabla 4 y las clases en la Tabla 5, cabe destacar que los atributos del 1 al 12 y 34 son conocidos como atributos clínicos, toman valores enteros de 0, 1, 2 y 3 siendo 0 el de sintomatología nula y 3 el de sintomatología total, excepto en 34 en donde se ve solo la edad, otra parte está constituida por los valores del 12 al 33, son llamados atributos histopatológicos, también se miden de 0 a 3 y poseen el mismo significado que los anteriores.

Tabla 4. Información de atributos de base de datos de Dermatología

Característica:
1: Erythema
2: Scaling
3: Definite Borders
4: Itching
5: Koebner Phenomenon
6: Polygonal Papules
7: Follicular Papules
8: Oral Mucosal Involvement
9: Knee And Elbow Involvement
10: Scalp Involvement
11: Family History, (0 Or 1)
12: Melanin Incontinence
13: Eosinophils In The Infiltrate
14: Pnl Infiltrate
15: Fibrosis Of The Papillary Dermis
16: Exocytosis
17: Acantosis
18: Hyperkeratosis

19: Parakeratosis
20: Clubbing Of The Rete Ridges
21: Elongation Of The Rete Ridges
22: Thinning Of The Suprapapillary Epidermis
23: Spongiform Pustule
24: Munro Microabcess
25: Focal Hypergranulosis
26: Disappearance Of The Granular Layer
27: Vacuolisation And Damage Of Basal Layer
28: Spongiosis
29: Saw-Tooth Appearance Of Retes
30: Follicular Horn Plug
31: Perifollicular Parakeratosis
32: Inflammatory Monoluclear Inflitrate
33: Band-Like Infiltrate
34: Age (Linear)

Tabla 5. Cantidad de datos por clase para base de datos de Dermatología

Clase	*Diagnostico*	*Cantidad*
1	Psoriasis	112
2	Seborrheic Dermatitis	61
3	Lichen Planus	72
4	Pityriasis Rosea	49
5	Chronic Dermatitis	52
6	Pityriasis Rubra Pilaris	20

5.2. ERROR DE LOS CLASIFICADORES

El MSE es usado para determinar la medida cuando el clasificador no se ajusta a la información, o si eliminando ciertos términos es posible mejorar el rendimiento del mismo, dando la información necesaria para elegir el mejor clasificador. Para esto se utiliza un conjunto de datos de prueba que debe contener muestras de todas las clases de la base de casos, si un MSE es mínimo, indica una variación mínima, y por lo tanto indica una buena estimación en el proceso de clasificación, si por el contrario el MSE es elevado indica una mala estimación de la asignación de las muestras a sus clases verdadera.

5.3. MEDIDAS DE DESEMPEÑO

Se utilizan las siguientes medidas de desempeño: sensibilidad (Se), especificidad (Sp) y porcentaje de clasificación (CP). Las ecuaciones que de estas medidas de desempeño se expresan en la Tabla 6, para estas se tienen los siguientes cuatro casos:

- VP Son los verdaderos positivos o casos de la clase de interés clasificados correctamente.
- VN Son los verdaderos negativos o casos diferentes de la clase de interés clasificados correctamente.
- FP Son los falsos positivos o casos diferentes de la clase de interés clasificados como casos de la clase de interés.
- FN Son los falsos negativos o casos de la clase de interés clasificados como casos diferentes de la clase de interés.

Tabla 6. Ecuaciones de medidas de desempeño

Nombre de la medida	***Definición***	***Descripción***
Sensibilidad	$\frac{Vp}{Vp + Fn}$	Mide la proporción de muestras positivas correctamente clasificadas
Especificidad	$\frac{Vn}{Vn + Fp}$	Mide la proporción de muestras negativas correctamente clasificadas.
Exactitud	$\frac{Vn + Vp}{Vn + Fp + Vp + Fn}$	Entrega una relación de los datos correctamente clasificados con respecto al número total de datos del conjunto de prueba

Estas medidas se usan para medir el desempeño del sistema, pero no tienen implicación en la sintonización de los parámetros del proceso de clasificación.

5.4. MATRÍZ DE CONFUSIÓN

Una matriz de confusión es una herramienta que permite la visualización del desempeño (exactitud de una clasificación) de un algoritmo que se emplea en aprendizaje supervisado. Esta es una matriz cuadrada de $n\ x\ n$, donde n representa el número de clases, las columnas de la matriz representan el número de predicciones de cada clase, mientras que cada fila representa a las instancias en la clase real. En el caso bi-clase la matriz se representa como la Figura 8.

Figura 8. Matriz de confusión para el caso Bi-clase.

		Clasificación	
		Verdadero	**Negativo**
Clase Real	**Verdadero**	Verdaderos positivos	Falsos positivos
	Negativo	Verdaderos negativos	Falsos negativos

Cada columna de la matriz representa el número de predicciones de cada clase, mientras que cada fila representa a las instancias en la clase real. Entre más elementos estén en la diagonal (color azul) es mejor la clasificación.

Para el caso multi-clase la matriz de confusión se explica en la Figura 9:

Figura 3. Matriz de confusión para el caso multi-clase.

		Clasificación			
		Clase 1	**Clase 2**	**Clase 3**	**Clase 4**
Clase Real	**Clase 1**	$\boldsymbol{VP_1}$	E_{12}	E_{13}	E_{14}
	Clase 2	E_{21}	$\boldsymbol{VP_2}$	E_{23}	E_{24}
	Clase 3	E_{31}	E_{32}	$\boldsymbol{VP_3}$	E_{34}
	Clase 4	E_{41}	E_{42}	E_{43}	$\boldsymbol{VP_4}$

Cada columna de la matriz representa el número de predicciones de las clases reales, mientras que cada fila representa a las instancias de dicha clase. Entre más elementos estén en la diagonal (color azul) es mejor la clasificación, donde:

$\boldsymbol{Vp}$: Pertenece a los casos de la clase de interés clasificados correctamente. (ubicados en la diagonal).

$\boldsymbol{Fn}$: Corresponde a la suma de valores de la fila de la clase de interés (Excluyendo VP).

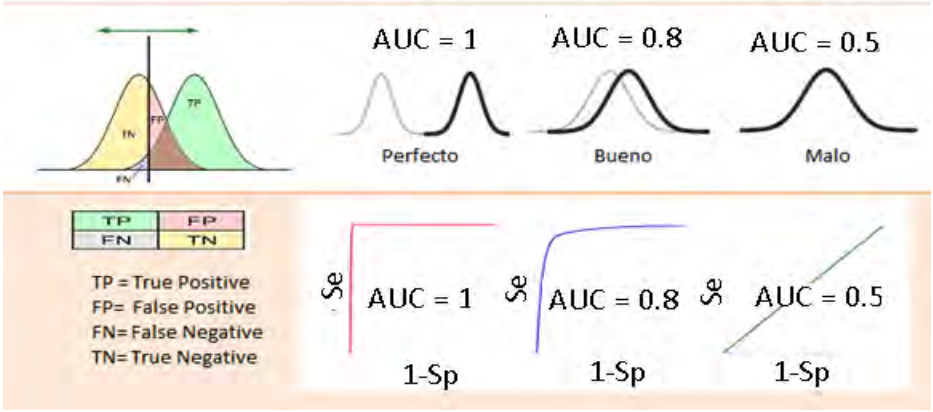
$\boldsymbol{Fp}$:corresponde a la suma de valores de la columna de la clase de interés (Excluyendo VP).

$\boldsymbol{Vn}$:corresponden a la suma de valores de todas las filas y columnas (Excluyendo la fila y columna de la clase de interés).

5.5. CURVAS ROC (Receiver-Operating Characteristic)

La curva ROC es una técnica de visualización que permite observar el desempeño de los clasificadores. En este se presenta la sensibilidad de una prueba diagnóstica que produce resultados continuos en función de los falsos positivos para distintos puntos de corte. También se refiere a la representación de la razón o ratio de verdaderos positivos (VPR) frente a la razón o ratio de falsos positivos (FPR). En la Figura 10 se encuentra la relación entre las curvas ROC y las curvas de dos poblaciones, se puede observar que cuando el valor del AUC es igual a 1 hay una separación perfecta de las poblaciones, mientras que si el valor de AUC es de 0.5 no existe ninguna separación.

Figura 10. Tipos de curvas ROC. Representa una herramienta para seleccionar los modelos posiblemente óptimos de acuerdo a los valores de sensibilidad y especificidad. Se muestra el comportamiento de clasificación en dos poblaciones, en tres situaciones diferentes: perfecta, buena y mala clasificación y su relación con las curvas ROC.


TP
FP
TN
FN
AUC = 1
AUC = 0.8
AUC = 0.5
Perfecto
Bueno
Malo
TP
FP
FN
TN
TP = True Positive
FP= False Positive
FN= False Negative
TN= True Negative
Se
AUC = 1
1-Sp
Se
AUC = 0.8
1-Sp
Se
AUC = 0.5
1-Sp

5.6. EXPERIMENTOS REALIZADOS

5.6.1. Pre-procesamiento

- Método de selección de características.

En la búsqueda de determinar la necesidad de un método de reducción de características se opta por realizar 3 experimentos para analizar el comportamiento del error de los clasificadores (SVM (kernel lineal), Parzen, Random Forest, KNN y Naive Bayes) frente a las bases de datos estudiadas. En este proceso se hace uso de la herramienta de minería de datos WEKA, esta nos permite utilizar el algoritmo de selección de atributos CFS junto con el método de búsqueda Best First y el algoritmo de selección infogainAttribute con el método Ranker. Estas dos combinaciones serán evaluadas en torno a los diferentes porcentajes de clasificación favorable, no favorable y el tiempo de ejecución que tomo cada clasificador. Cabe destacar que los experimento se realizaron en dos computadores con características diferentes, con el fin de analizar el comportamiento de los mismos experimentos ante diferentes condiciones. El primer computador tiene un procesador Intel Core i5 de 2.3Ghz y memoria ram de 8GB (denominado C1), el segundo tiene un procesador Intel Core i7 de 2.8Ghz, y memoria ram de 12GB (denominado C2). Por último, el grupo de entrenamiento y test fue de 70% y 30% respectivamente para todos los experimentos.

- Experimento 1. Con el objetivo de mirar el comportamiento de las dos bases de datos sin modificaciones, los clasificadores fueron entrenados con las bases de datos en bruto. Esta información se toma como base para los otros dos experimentos y así concluir si es necesario la aplicación de dichos métodos.

- Experimento 2. Se aplicó el método de selección CFS-SubsetEval -Best First, este redujo el número de atributos de 33 a 19 en la base de datos de Dermatología, estos atributos son los números: 2, 3, 4, 5, 7, 9, 13, 14, 15, 16, 20, 21, 22, 25, 26, 28, 29, 31 y 33, estos atributos se nombran en la Tabla 4

- Experimento 3. Es aplicado el método de selección InfogainAtribute-Ranker, este a diferencia del algoritmo anterior, no entrega un número menor de atributos de forma directa, sino que da un peso de importancia a cada atributo, por lo tanto, se toman los 19 atributos con mayores pesos, los cuales son: 21, 20, 22, 33, 29, 27, 12, 25, 6, 16, 8, 28, 9, 15, 10, 24, 14, 5 y 26. Este es el orden de importancia que da el método de selección, el nombre de los atributos está en la Tabla 4.

- Métodos de Balanceo de Clases

Una vez determinada la necesidad de reducir la base de datos se procedió a hacer un análisis para determinar si la cantidad de registros por clase es relevante para el óptimo entrenamiento de los clasificadores, por lo que se procede a tomar diferentes técnicas de balanceo de datos para mejorar la clasificación evitando problemas como el sobre-entrenamiento, falta de información y el exceso de costo computacional. El clasificador para realizar esta serie de experimentos fue el KNN, debido a que no se necesitó evaluar el desempeño del clasificador frente a otros sino el comportamiento de los datos frente a este, además de ser uno de los más utilizados. Los grupos de entrenamiento y test fueron del 70% y 30% respectivamente para todas las pruebas. A diferencia de los métodos de selección estas pruebas fueron hechas con la herramienta de programación Matlab.

- Experimento 1. Las dos bases de datos se prueban utilizando solo el método de selección para determinar el error de asignación de cada clase y el tiempo que tarda en realizar el proceso de clasificación. Estos valores son tomados como referencia para próximas pruebas en las cuales la disminución o aumento del error y tiempo definirá si ha existido un cambio significativo en el rendimiento del proceso de clasificación.

- Experimento 2. Buscando que cada clase en las bases de datos tenga una misma cantidad de registros, todas las clases son reducidas al valor de registros de la clase minoritaria. Con esto la base de datos de Dermatología pasó a tener 20 registros en sus 6 clases e Hipotiroidismo 50 registros en sus 3 clases, estos registros son seleccionados aleatoriamente para evitar tener los mismos casos en cada clasificación. Este experimento se realiza con el propósito de determinar si un pequeño conjunto de registros por clase representa significativamente las bases de datos (datos homogéneos), de tal forma que sea innecesario utilizar un gran número de casos.

- Experimento 3. Se utilizó el método de balanceo de sobre-muestreo SMOTE, se aumentaron los registros de la clase minoritaria de la siguiente forma: para la base de datos de Hipotiroidismo la clase 3 paso de tener 50 casos a 102 casos (igual a la clase 2), después se tomaron grupos al azar de 100 casos en la clase 1. De forma similar en la base de datos de Dermatología las clases de la 2 a la 6 fueron aumentados hasta los 100 registros, para tener un valor casi igual al de la clase 1(clase mayoritaria). Este proceso fue realizado con el fin de tener similar un grupo de entrenamiento homogéneo en cuanto al número de casos de cada clase.

- Experimento 4. En la búsqueda de un enfoque diferente al anteriormente expuesto se procede a un método de sub-muestreo llamado KNN undersampling, las clases mayoritarias fueron reducidas de la siguiente

manera: en la base de datos de Hipotiroidismo la clase 1 paso de 1790 a 60 casos, en el caso de Dermatología la clase 1 paso a tener 20 casos. Las demás clases conservaron la misma cantidad de registros que las bases de datos originales.

- Experimento 5. Teniendo en cuenta los buenos resultados de los experimentos 3 y 4, se realizó un algoritmo hibrido compuesto por la técnica SMOTE y KNN-U. El parámetro general para las dos bases de datos fue el dejar todas las clases con un número igual a 100 registros para evitar un costo computacional muy grande, debido a que una gran cantidad de registros requiere un mayor tiempo para clasificar y si por el contrario se tienen muy pocos casos la información no puede ser suficiente para entregar un clasificador bien entrenado

.

5.6.2. Experimentación para clasificación en cascada.

En la implementación de nuevas formas de clasificación dentro en un sistema CBR, se busca por medio de un proceso secuencial encontrar una mejora en la eficiencia de dicho sistema. Para los experimentos siguientes se tomaron porcentajes del grupo de entrenamiento y test del 70% y 30% respectivamente. El número de veces que se realizó la clasificación fue 100 veces, para cumplir con el criterio de repetitividad en las pruebas. Los clasificadores usados son:

1. SVM: Este método utiliza el kernel para calcular el hiper- plano no lineal discriminante entre clases. Para este experimento, se selecciona el kernel lineal.

2. Parzen: Es un método de clasificación basado que se basa en probabilidades requiriendo un parámetro de suavizado, el cual se optimiza durante el entrenamiento para el cálculo de la distribución Gaussiana.

3. Random Forest: Este método combina árboles predictores tal que cada árbol depende de los valores de un vector aleatorio probado independientemente y con la misma distribución para cada uno de estos, el número de árboles en todos los experimentos es 100.

4. KNN: Esta técnica de clasificación basada en muestras, necesita el valor del número de vecinos (*K*), para estos experimentos este parámetro es igual a 4.

5. Naive Bayes: Este método calcula las probabilidades condicionales para las combinaciones de atributos y el atributo de objetivo. A partir de los datos de entrenamiento se establece una probabilidad independiente.

   - *Experimentación sin separación de clase para clasificación multi-clase.*

Colocando clasificadores en secuencia en combinaciones de 2 y 3

clasificadores y teniendo en cuenta que el resultado en el proceso de clasificación se ve afectado por el lugar en el cual el clasificador se encuentre en la serie, se implementan secuencias sin repetir ningún clasificador en cada serie, así se evitar parecerse a un método como el Adaboost que tiene un mismo clasificador (weak clasifier) para entrenar en toda la secuencia. Se realizaron 20 diferentes combinaciones para una composición de 2 clasificadores y 55 para 3 clasificadores, los resultados obtenidos se expresan en términos de error por combinación de clasificador y tiempo de ejecución.

- *Experimentación con separación de una clase.*

Se procedió a realizar la separación de una clase, con el fin de tener un clasificador con una menor cantidad de clases para procesar, con esto se buscó un mejor rendimiento en la clasificación.

- Experimento para encontrar mejor clasificador bi-clase y la mejor clase para separar.

Con el objetivo de tomar la clase con mejor error de clasificación, se procedió hace una clasificación bi-clase en donde se toman cada uno de las clases frente a un grupo de las otra, para posteriormente clasificarlas y así escoger la combinación de clasificador y clase que tenga un error MSE Igual a 0 o muy cercano. Para este proceso se le asignó el numero 1 a la clase evaluada y el numero 0 al conjunto de las demás clases, con el fin de hacer la clasificación bi-clase.

- Experimentación con separación de una clase y utilizando las mismas combinaciones de experimentación sin separación para clasificación multi-clase.

Tomando las 20 diferentes combinaciones para una composición de 2 clasificadores y 55 para 3 clasificadores (Experimentación sin separación de clase para clasificación multi-clase), se procede a clasificar el grupo de las clases que fueron seleccionados como 0, previo a una reconversión a su clase original.

5.6.3. Experimentación estimación de la probabilidad.

En este proceso se tienen en cuenta los estimadores de probabilidad, Windows Parzen, KNN y SVM, en la sección 3.4.2 se habla sobre sus características. Para la experimentación se hizo un algoritmo simulando un CBR automático que guardaba solo los resultados pronosticados bien, Para este proceso se tomó el 20% de los datos totales para pronosticar, el restante 80% fue utilizado para entrenar el estimador. Después de entrenado el 20% de los datos eran enviados para que se les asigne una clase, con el conocimiento previo de la clase real, se corroboraba, si este pronóstico era acertado, si este fuera el caso

se guardaba sino era desechaban. Este proceso fue realizado en ambas bases de datos fue realizado 100 veces, para tener una media de porcentaje con una cantidad aceptable de repeticiones, los resultados son expuestos en la sección 6.4.

6. RESULTADOS

Por medio de tablas y figuras se presentan los resultados obtenidos de los experimentos descritos en la sección anterior, se omiten algunos que, aunque ayudan a soportar la metodología propuesta no son tan relevantes como los expuestos en esta sección.

6.1. MÉTODOS DE SELECCIÓN

- Experimento 1.

Base de datos de Hipotiroidismo clasificada por 6 diferentes clasificadores y sin ningún método de selección.

Tabla 7. Base de datos de Hipotiroidismo sin ningún método de selección.

Clasificador	Porcentaje de Clasificación %				Tiempo (S)	
	Favorable		No Favorable			
	C1	C2	C1	C2	C1	C2
NaiveBayes	**97,26**	91,97	2,73	8,03	0,00	0,00
Multilayer Perceptron	96,17	92,90	3,82	7,10	**24,72**	**8,48**
KNN(1)	94,53	90,22	5,46	9,78	0,00	0,00
SVM	95,35	92,49	4,64	7,51	0,08	0,33
Random Forest	95,90	93,77	4,09	6,23	0,05	0,63

Tabla 8. Base de datos de Dermatología sin ningún método de selección.

Clasificador	Porcentaje de Clasificación %				Tiempo (S)	
	Favorable		No Favorable			
	C1	C2	C1	C2	C1	C2
NaiveBayes	95,28	97,54	4,71	2,46	0,05	0,00
Multilayer Perceptron	94,16	98,36	5,83	1,64	**22,68**	**2,18**
KNN(1)	91,51	95,36	8,48	4,64	0,00	0,00

SVM	93,61	97,27	6,38	2,73	2,34	0,05
Random Forest	**99,31**	97,27	0,68	2,73	0,97	0,17

-

- Experimentos 2 y 3.

Base de datos de Hipotiroidismo clasificada por 6 diferentes clasificadores y con los métodos de selección CFS-SubsetEval-BestFirst y InfoGainAttribute-Ranker.

Tabla 9. Hipotiroidismo con métodos BestFirst y Ranker.

Clasifi-cadores	**Porcentaje Clasificación Favorable %**				**Porcentaje Clasificación No Favorable %**				**Tiempo (S)**			
	Best First		Ranker		Best First		Ranker		Best First		Ranker	
	C1	C2	C1	C2	C1	C2	C1	C2	C1	C2	C1	C2
NaiveBayes	**97,81**	94,66	**97,26**	93,10	2,18	5,34	2,73	6,90	0,00	0,00	0.00	0,00
Multilayer Perceptron	96,44	96,25	95,90	93,15	3,55	3,75	4,09	6,85	**9,38**	**1,28**	**9,66**	**1,50**
KNN(1)	96,44	95,07	95,35	89,19	3,55	4,93	4,64	10,81	0,00	0,00	0.00	0,00
SVM	**97,26**	93,11	**97,26**	92,07	2,73	6,89	2,73	7,93	0,06	0,04	0,06	0,03
Random Forest	96,44	98,30	95,62	92,69	3,55	1,70	4,37	7,31	0,05	0,03	0,03	0,03

Tabla 10. Dermatología con métodos Best First y Ranker.

Clasifi-cadores	**Porcentaje Clasificación Favorable %**				**Porcentaje Clasificación No Favorable %**				**Tiempo (S)**			
	Best First		Ranker		Best First		Ranker		Best First		Ranker	
	C1	C2	C1	C2	C1	C2	C1	C2	C1	C2	C1	C2
NaiveBayes	94,64	94,66	94,72	92,08	5,35	5,34	5,27	7,92	0.00	0,00	0.00	0,00
Multilayer Perceptron	**96,10**	96,25	96,26	95,98	3,89	3,75	3,73	4,02	**2,94**	**1,28**	**3,02**	**0,99**
KNN(1)	93,16	95,07	94,22	95,44	6,83	4,93	5,77	4,56	0.00	0,00	0.00	0,00
SVM	93,13	93,11	93,50	90,44	6,86	6,89	6,49	9,56	0,20	0,14	0,13	0,04
Random Forest	95,78	98,30	**97,50**	89,62	4,21	1,70	2,49	10,3	0,45	0,30	0,45	0,05

En los resultados expuestos en la Tabla 8 y Tabla 10 pertenecientes a la base de datos de Dermatología se puede observar que para todos los clasificadores hubo

un aumento en el porcentaje de clasificación favorable usando el método características Best First, mientras que en términos de selector Ranker el clasificador SVM fue el único en mostrar un aumento en el porcentaje de clasificación en comparación a los demás, exceptuando al clasificador Naivebayes que mantuvo su valor en comparación a las bases de datos sin ningún método de selección añadido. En términos de tiempo de ejecución todos los clasificadores con método de selección tardan menos en comparación con la tabla 8 (independientemente del proceso de selección), exceptuando el clasificador RandomForest con selector Best first.

Para la base de datos de Hipotiroidismo (Tablas 7 y 9), sin importar el tipo de selector usado los clasificadores Multilayer Perceptron y KNN presentaron un aumento en el porcentaje de clasificación favorable, al contrario, NaiveBayes, SVM y Random Forest disminuyeron en su porcentaje. Respecto al tiempo empleado al igual que con la base de datos anterior todos los tiempos disminuyeron de forma considerable un ejemplo es el tiempo dado por el clasificador Multilayer Perceptron, en los dos métodos este baja considerablemente, es así como se pasó de un tiempo de 24.72s y 8.48s a unos de alrededor de 9s y 1.3s para el C1 y C2 respectivamente.

6.2. MÉTODOS DE BALANCEO

El resumen de los 5 experimentos se colocó en las Tablas 11 y 12. La primera se refiere al porcentaje de buena clasificación en cada experimento para cada base de datos, en la segunda el tiempo dado en segundos que duro cada proceso de clasificación utilizando el método de balanceo respectivo.

Tabla 11. Promedio del porcentaje de error, experimentos del 1 al 5, para las bases de datos de Hipotiroidismo y Dermatología

Base de Datos	**Experimentos**				
	1	2	3	4	5
Hipotiroidismo	5,39±5,39	**25,26±8,78**	9,20±9,36	14,13±5,96	**2,37±5,53**
Dermatología	**2,90±8,68**	4,81±8,82	4,57±8,77	3,28±9,87	**2,69±9,87**

Tabla 12. Promedio de tiempos para los experimentos del 1 al 5 para las bases de datos de Hipotiroidismo y Dermatología

Base de Datos	Experimentos				
	1	2	3	4	5
Hipotiroidismo	0,60±0,05	**0,20±0,61**	0,35±0,01	9,13±0,03	**38,26±0,002**
Dermatología	**0,23±0,01**	0,23±0,03	0,35±0,01	0,36±0,01	**8,41±0,008**

Como resultados destacados para la Tabla 11 de porcentajes de error, se tiene que el experimento 5 fue el de mejor clasificación con 2.37±5.53% y 2.69±9.87% para Hipotiroidismo y Dermatología respectivamente. Es de destacar que los errores en la base de datos de Dermatología, en todos los experimentos tienen una diferencia que no pasa del 3%, mientras que en la otra la diferencia llega hasta el 23% de diferencia (experimentos 5 y 2).

Para la Tabla 12 de tiempos, es de resaltar que el experimento 5 (algoritmo hibrido SMOTE-KNNU) presenta el mayor costo computacional para las dos bases de datos. La razón de este comportamiento es la programación para unir los dos algoritmos, puesto que el algoritmo KNNU no empieza a proceder hasta que el SMOTE produce todas sus muestras sintéticas, para después iniciar su proceso de sub-muestreo para todas las clases, esto proceso requiere de un gran gasto de tiempo computacional.

Las tablas en donde se muestran los datos de error y tiempo por cada clasificación son expuestas en el Anexo 1.

- Matriz de confusión y Curvas ROC

Los resultados dados en las Tablas 11 y 12, aunque brindan una buena información tomando el promedio del error y tiempo para todas las pruebas, no proporcionan información suficiente sobre problemas de sobre-entrenamiento en una clase, para esto utilizamos los datos que proporciona la matriz de confusión. En esta solo se tienen en cuenta los experimentos 1 y 5, debido a que el comportamiento de estos experimentos (clasificación sin ningún método y con el menor error) entregan la información necesaria para utilizar método o no de balanceo.

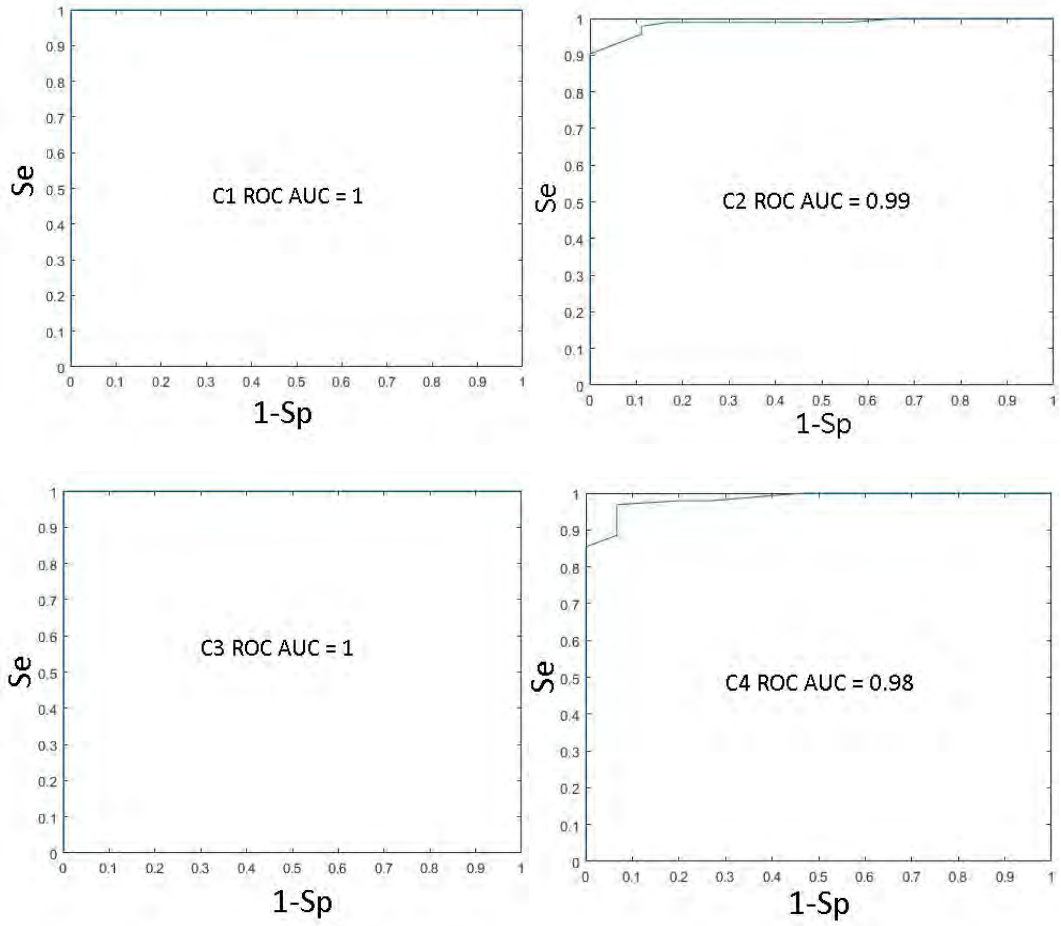
- *Experimento 1*

Tabla 13. Se y Sp de Dermatología para experimento 1.

Medidas	Clase 1	Clase 2	Clase 3	Clase 4	Clase 5	Clase 6	Pro.
Se	1.00	0.94	1.00	0.86	1.00	0.98	**0.96**

Sp	1.00	0.98	1.00	0.99	1.00	1.00	**0.99**

Figura 11. Curvas ROC de las 6 clases pertenecientes a Dermatología para el experimento 1, nombradas C1, C2, C3, C4 y C5, donde se muestra el valor de su respectivo AUC (Área bajo la curva).



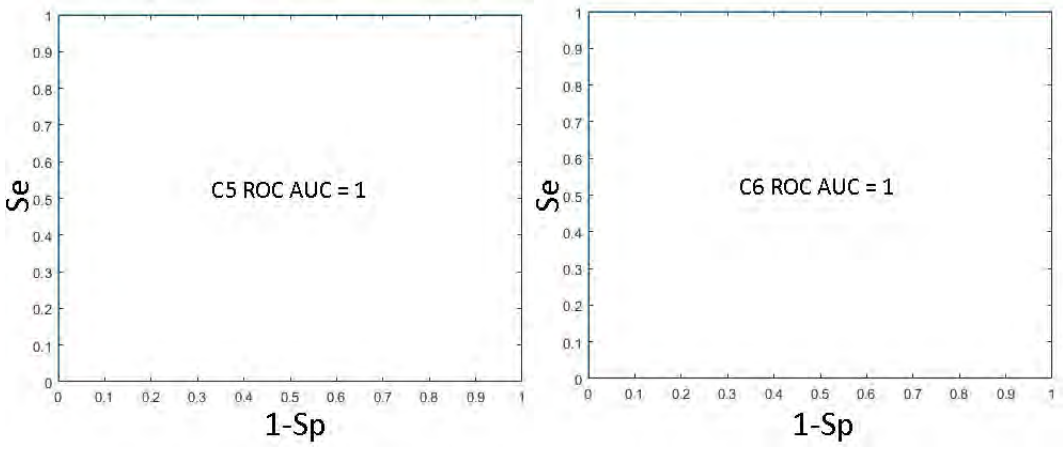


Tabla 14. MC para la base de datos de Dermatología experimento 1.

	1	2	3	4	5	6
1	6794	6	0	0	0	0
2	0	3394	0	206	0	0
3	0	0	4400	0	0	0
4	0	424	0	2576	0	0
5	0	1	0	2	3197	0
6	6	9	0	0	0	1185

Para la base de datos de Dermatología, según sus curvas ROC (Figura 11) se tiene una buena clasificación en todas sus clases. En razón que las AUC en todas las clases están alrededor de 0.99 (siendo lo ideal 1), siendo esta una muy buena clasificación. En tanto a la sensibilidad e especificidad, observadas en la Tabla 13, se obtuvieron valores de promedio altos con Se =0.96 y Sp =0.99. En la matriz de confusión (Tabla 14), no se observan sobre-entrenamiento en ninguna clase puesto que los datos no presentan ninguna acumulación en ninguna columna de clase estimada.

Tabla 15. Se y Sp base de datos de Hipotiroidismo para experimento 1.

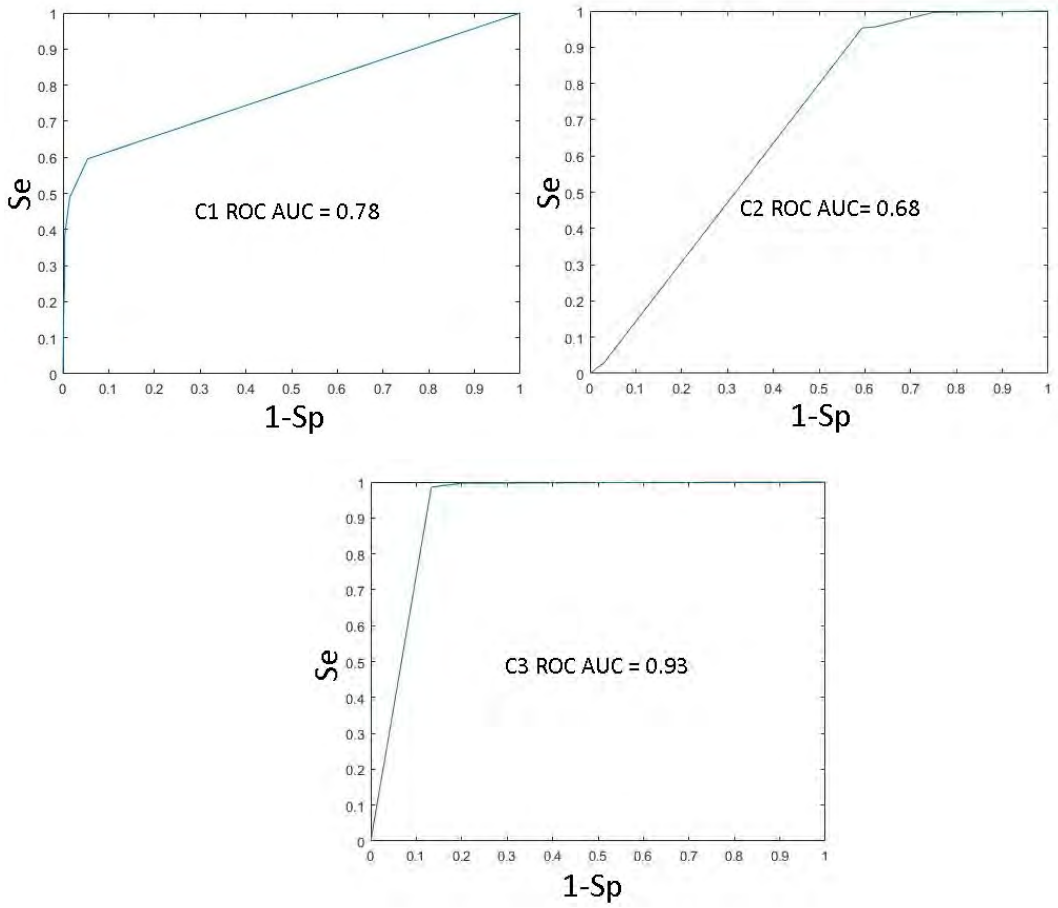
Medidas	**Clase 1**	**Clase 2**	**Clase 3**	**Promedio**
Se	0.99	0.28	0.83	0.70

Sp	0.46	0.99	1.00	0.82

Tabla 16. MC para la base de datos de Hipotiroidismo experimento 1.

	1	2	3
1	106246	767	387
2	4536	1765	99
3	452	50	2498

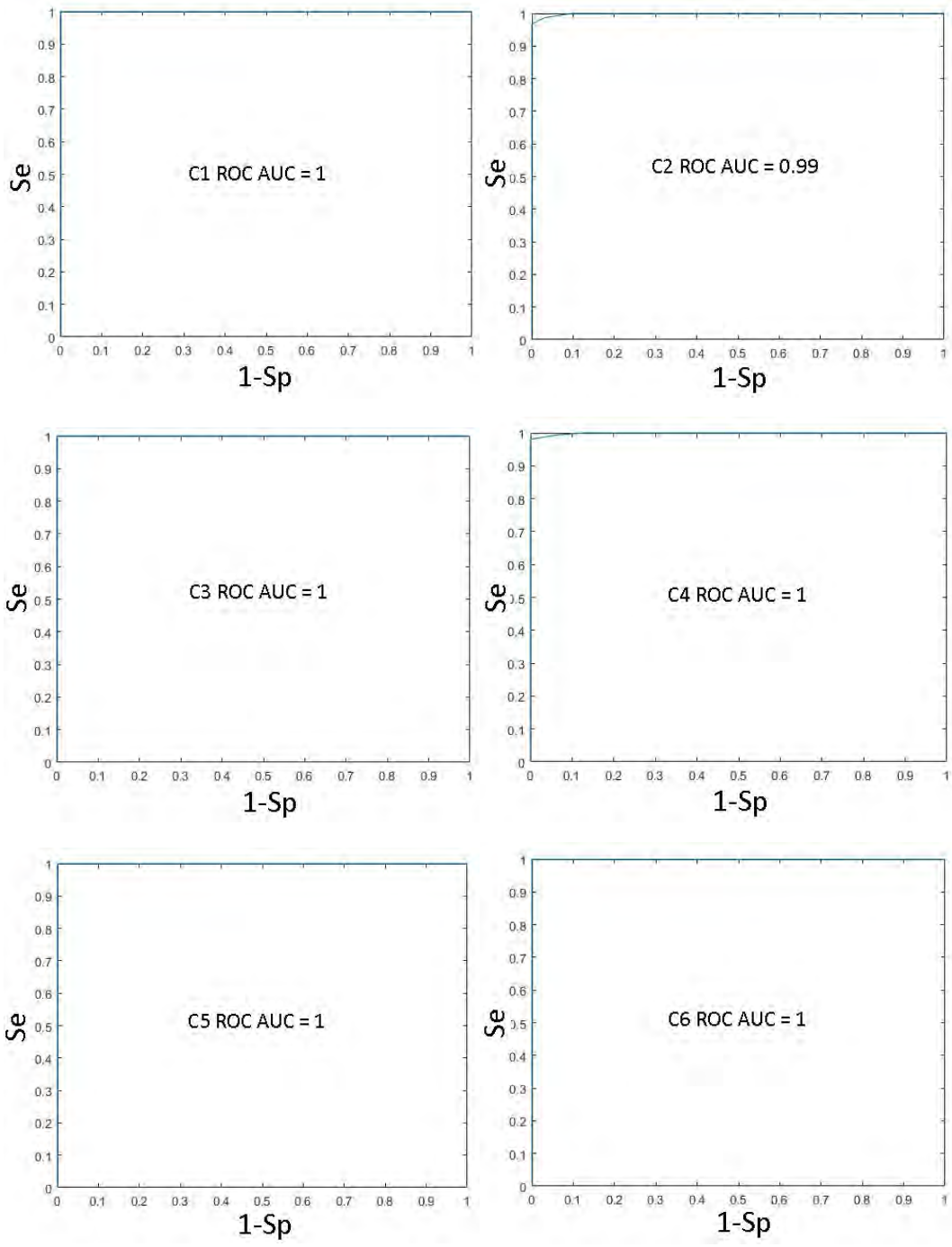
Figura 4. Curvas ROC de las 3 clases de Hipotiroidismo para el experimento 1, nombradas C1, C2 y C3, donde se muestra en valor de su respectivo AUC (Área bajo la curva).



En las curvas de la base de datos de Hipotiroidismo (Figura 12), al contrario de lo observado en la otra base de datos, los valores de la AUC para las dos primeras clases son muy bajos con 0.78 y 0.67 para la clase 1 y la clase 2 respectivamente. Para las medidas de Se y Sp (Tabla 15), los promedios son relativamente bajos con 0.7 en los dos casos y para la matriz de confusión (Tabla 16), se tienen muchos valores de falsos negativos y falsos positivos. También en esta tabla se puede ver un sobre-entrenamiento en la clase 1, en donde existen muchos valores mal estimados en esta clase.

- *Experimento 5*

Figura 5. Curvas ROC de las 6 clases pertenecientes a Dermatología para el experimento 5, nombradas C1, C2, C3, C4 y C5, donde se muestra el valor de su respectivo AUC (Área bajo la curva).



.

Tabla 17. Se y Sp base de datos de Dermatología para experimento 5.

Medidas	**Clase 1**	**Clase 2**	**Clase 3**	**Clase 4**	**Clase 5**	**Clase 6**	**Promedio**
Se	1.00	0.97	1.00	0.93	1.00	1.00	0.98
Sp	1.00	0.99	1.00	0.99	1.00	1.00	1.00

Tabla 18. MC para la base de datos de Dermatología experimento 5.

	1	2	3	4	5	6
1	6188	10	0	0	0	2
2	0	6031	0	169	0	0
3	0	0	6200	0	0	0
4	0	445	0	5755	0	0
5	0	0	0	0	6200	0
6	0	0	0	0	0	6200

En las curvas ROC (Figura 13), se ve una leve mejoría en las de las clases 2 y 4 en donde la AUC pasó de 0.98 a 0.99 en la clase en los dos casos. En la Se y Sp (Tabla 17), también hubo mejoras en tanto en la primera se pasó de 0.96 a 0.98 y de 0.99 a 1 en la segunda. En la matriz de confusión (Tabla 18) se obtienen muy buenos resultados ya que la gran mayoría de resultados se ubican en la diagonal, lo que hace que los verdaderos positivos para cada clase tengan un buen porcentaje de acierto.

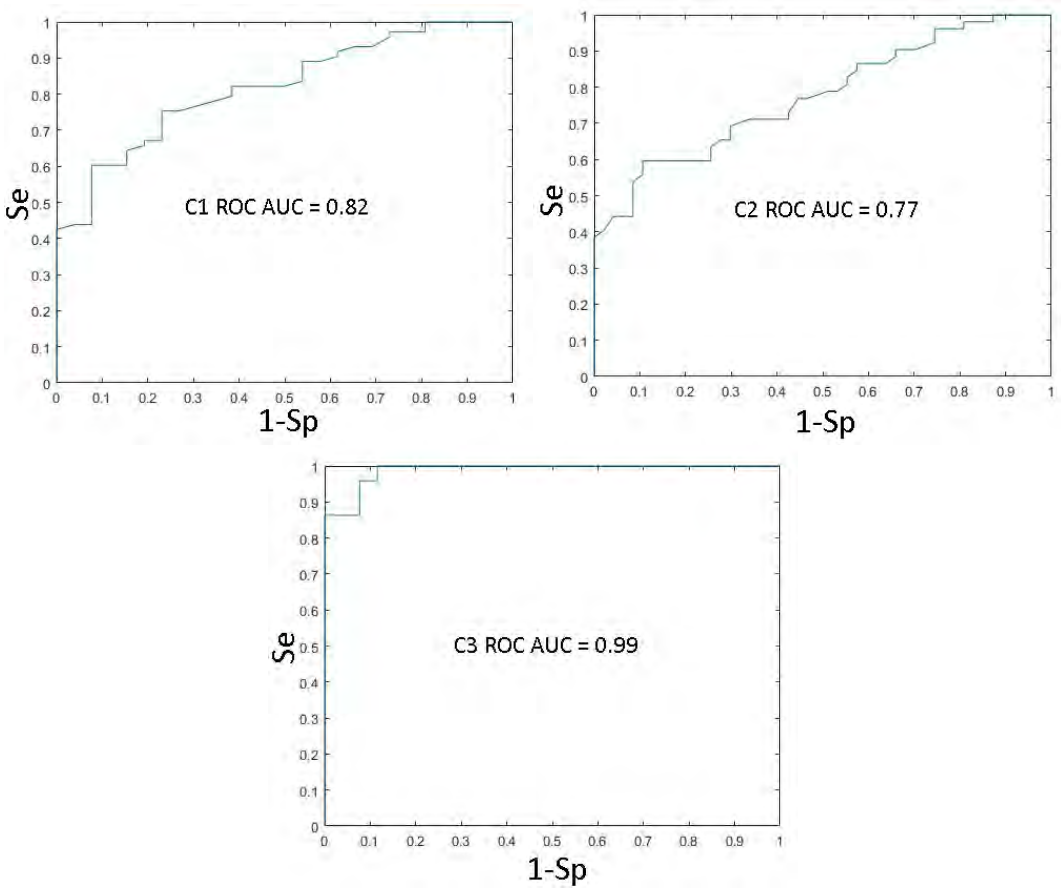
Tabla 19. Se y Sp base de datos de Hipotiroidismo para experimento 5.

Medidas	**Clase 1**	**Clase 2**	**Clase 3**	**Promedio**
$\boldsymbol{Se}$	0.98	0.97	0.98	0.98
$\boldsymbol{Sp}$	0.99	0.98	0.99	0.99

Tabla 20. MC para la base de datos de Hipotiroidismo experimento 5.

	1	2	3
1	3500	1560	140
2	2599	6641	160
3	66	311	4823

Figura 6. Curvas ROC de las 3 clases de Hipotiroidismo para el experimento 5, nombradas C1, C2 y C3, donde se muestra el valor de su respectivo AUC (Área bajo la curva).



Al igual que la anterior base de datos se presenta una mejora en las curvas AUC (Figura 14), en todas las clases el valor de mejora esta alrededor del 0.1. En cuanto a valores de Se y Sp (Tabla 19) hay una mejora sustancial de alrededor de 0.2 en las dos bases de datos. En la matriz de confusión (Tabla 20), aunque hay varios datos mal clasificados, no hay una concentración mayoritaria de datos estimados en una clase como si lo estaba en la Tabla 19.

6.3 CLASIFICACIÓN EN CASCADA

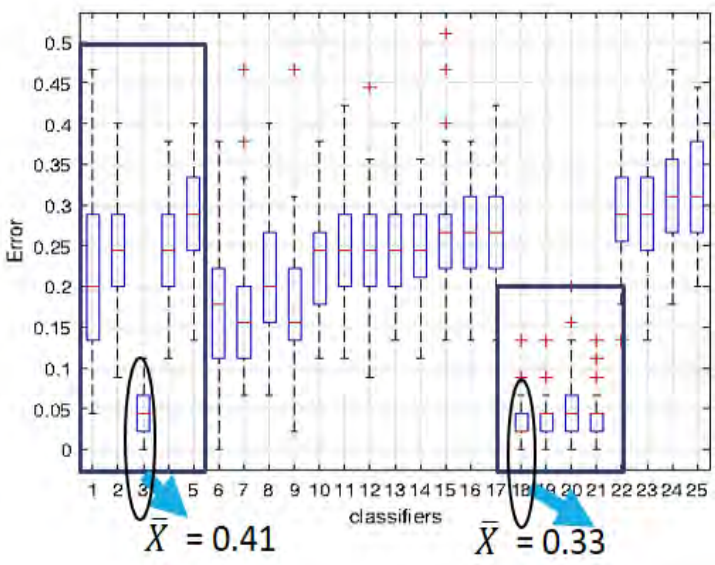
En esta sección se presentan los resultados más relevantes en combinación secuencial de clasificadores. De las diversas pruebas se escogieron aquellas que tenían una mejoría notable con respecto a las pruebas de clasificadores individuales. Las demás pruebas se muestran en el Anexo 1, debido que no presentan una mejoría a las pruebas con un solo clasificador.

- Resultados sin separación de clase para clasificación multi-clase.

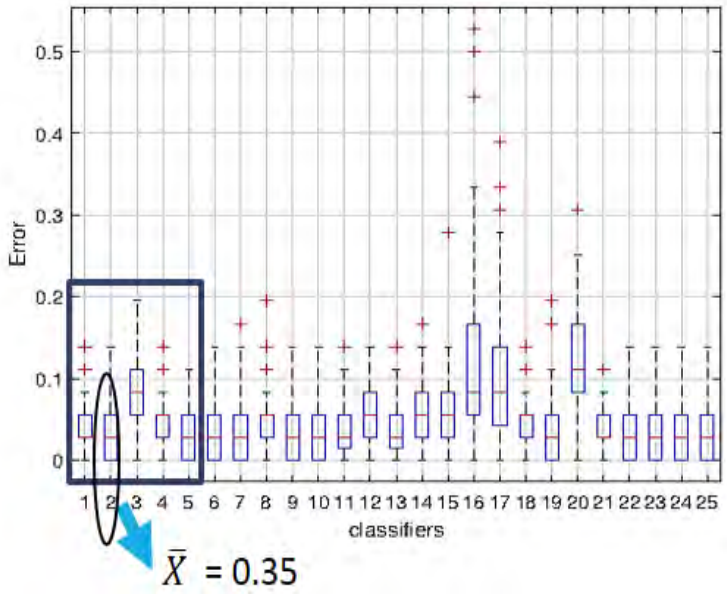
En las Figura 15 se puede ver el comportamiento de los 5 primeros clasificadores los cuales son: en su orden SVM, Parzen, Random Forest, KNN con 1 vecino y Naive Bayes, las combinaciones de la 6 a las 25 son todas las combinaciones de 2 clasificadores sin repetir.

Los resultados más importantes se dieron en la base de datos de Hipotiroidismo donde el mejor clasificador individual se encuentra en la posición 3 (Random Forest), con una media de 0.4 con desviación estándar de 0.2. En la combinación de cascada de clasificadores dobles, se encontró una mejoría en la en la combinación 18 (Random Forest - SVM), lo que demuestra una mejora del error utilizando clasificación en cascada. Por el contrario, en la base de datos de Dermatología no hubo mejora alguna ya que la mejor clasificación está en la posición 2 (Parzen) con 0.3.

Figura 7. Diagrama de cajas de errores de clasificadores individuales (1-5) y en combinación de dos clasificadores (6-25). La primera pertenece a la base de datos de Hipotiroidismo y la segunda a Dermatología.



(a)



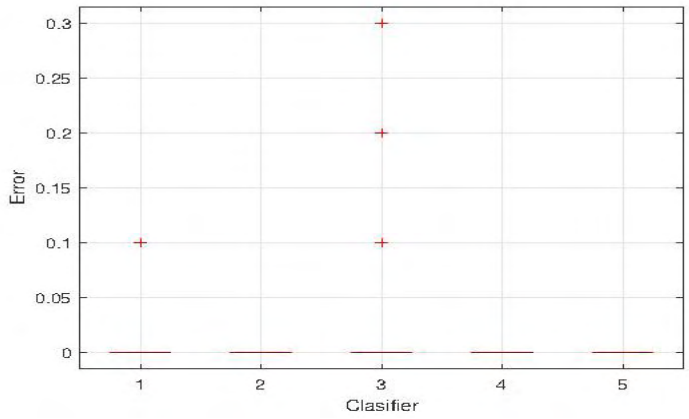
(b)

- Resultados con separación de una clase.

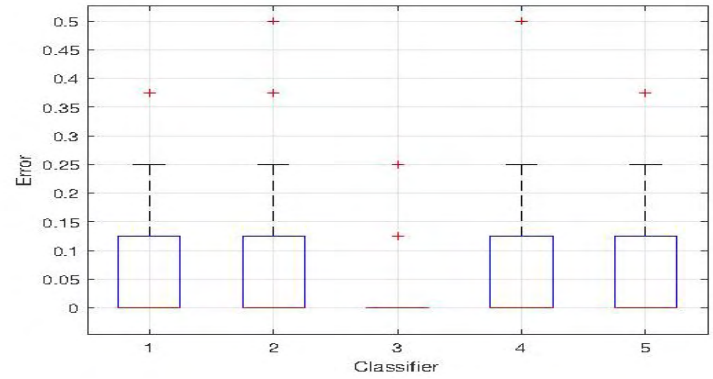
- Experimento para encontrar mejor clasificador bi-clase y la mejor clase para separar.

En la propuesta de separar una clase para mejorar la clasificación se utilizaron clasificadores individuales. La Figura 16 presenta a continuación fueron el resultado de separar la clase 3 en las dos bases de datos, en las cuales fue el mejor resultado. Los demás resultados de clasificación Bi-Clase separando una clase se pueden observar en el Anexo 1.

Figura 8. Diagrama de cajas de errores de clasificadores individuales de separación de clase 3 vs las otras.



(a)



(b)

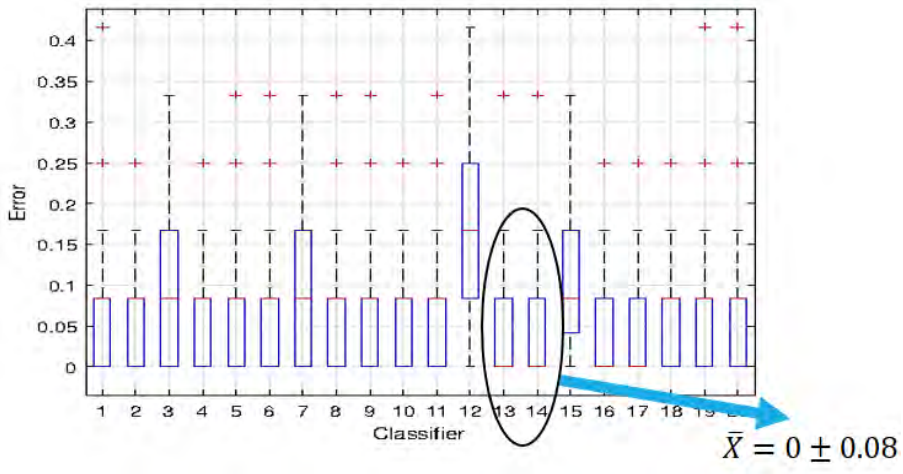
Los clasificadores son: 1. SVM, 2. Parzen, 3. Random Forest, 4.KNN y 5. Naive Bayes., la primera pertenece a la base de datos de Dermatología y la segunda a Hipotiroidismo.

Teniendo en cuenta los resultados en Dermatología se observa que en los todos los clasificadores la media del error es cero, con algunos datos alejados de la media en el clasificador Random Forest, mientras que en la base de datos de Hipotiroidismo el único clasificador que tiene una media igual a cero es el mismo. Por lo tanto, se decidió utilizar el clasificador bi-Clase Random Forest para separar la clase 3.

- Resultados con separación de una clase y utilizando las mismas combinaciones de experimentación sin separación para clasificación multi-clase.

Después de la separación de la clase 3 y al utilizar las combinaciones dobles de cascadas, la mejora en la clasificación se produjo únicamente en la base de datos de Hipotiroidismo (Figura 17), en donde la combinación 13 (Random Forest – SVM) y 14. (Random Forest – Parzen). Los dos obtuvieron una media de error igual a 0 con una desviación estándar aceptable de 0.08. Los resultados de la otra base de datos se encuentran en el Anexo 1

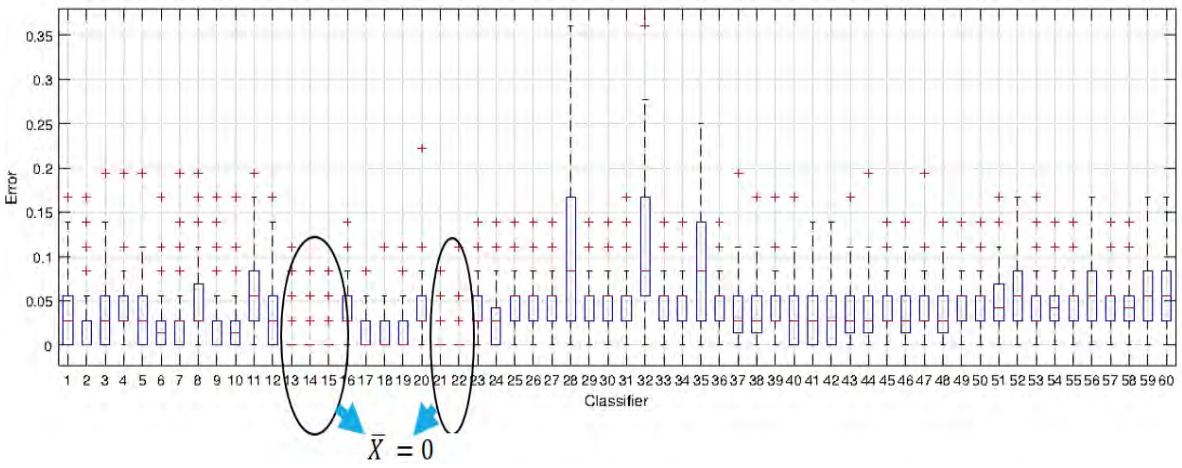
Figura 9. Diagrama de cajas, resultados cascados dobles Hipotiroidismo.



Los resultados al utilizar las cascadas triples se pueden observar en la Figura 18. Los mejores resultados se obtuvieron en la base de datos de Dermatología, en donde las combinaciones 13, 14, 15, 21 y 22 llegaron a tener una media en el promedio de 0 además de tener una desviación estándar 0 con 3 valores que están fuera de este rango, por lo tanto, se decide utilizar la combinación 14 (Random Forest- SVM- Parzen). Mientras que en la base de datos de Hipotiroidismo no hubo mejora adicional de la presentada por las combinaciones

dobles.

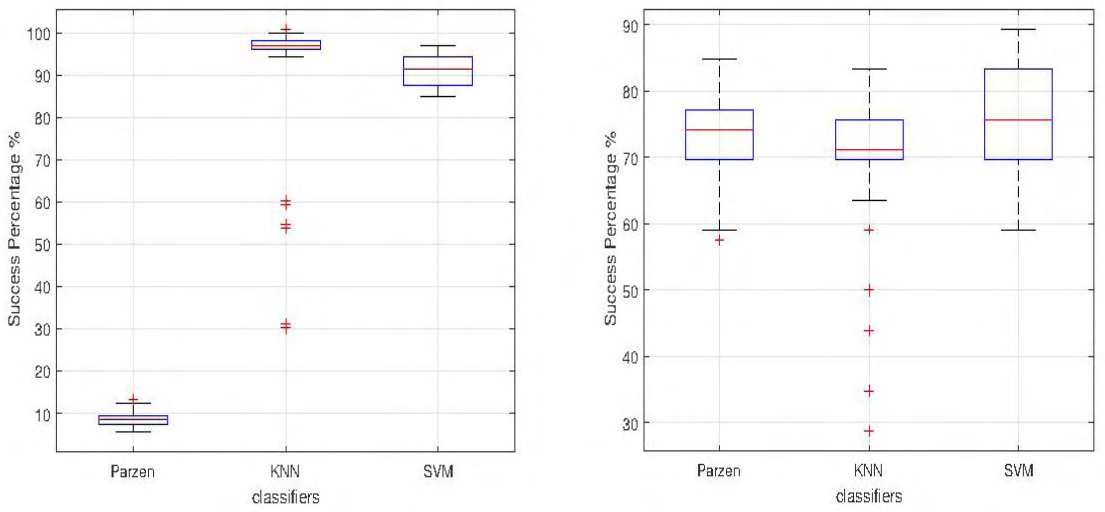
Figura 10. Diagrama de cajas**,** resultados cascados triples Dermatología.



6.4 PROBABILIDADES

Teniendo en cuenta la experimentación de la sección 5.5.3, se obtuvieron los siguientes resultados comparando los tres estimadores.

Figura 11. Estimación probabilidad, en la primera se presenta el porcentaje de aciertos de los estimadores para la base de datos de Dermatología, como segunda los resultados de la base de datos de Hipotiroidismo.



Para la base de datos de Dermatología (Figura 19) el estimador de ventanas

de Parzen tuvo el peor desempeño con solo el 9% de clases pronosticadas correctamente. Los clasificadores KNN y SVM tuvieron muy buenos resultados con el 96% y 92% respectivamente. Aunque en el estimador KNN existen algunos valores irregulares que se encuentran en el 30%, 55% y 60%, mientras que en el SVM los valores son mucho más compactos a la media. En la base de datos de Hipotiroidismo los tres estimadores entregan valores de media muy similares, en Parzen la media está alrededor del 75%, en KNN está en 72% y en SVM está alrededor del 77%. Otra cosa a tener en cuenta son los datos atípicos de cada estimador, para esto se observa que en las ventanas de Parzen hay un solo dato atípico cercano a la media, mientras que en KNN hay 6 incluso uno llegando a un valor por debajo del 30%. En el caso del SVM se presenta un comportamiento muy similar a la anterior base de casos, con todos los datos compactos dentro de la misma desviación estándar. Con esto se deduce que la mejor opción como estimador de probabilidad en el SVM, por lo tanto, se incluye dentro de la metodología propuesta sustentada en estos resultados.

6.5 INTERFAZ

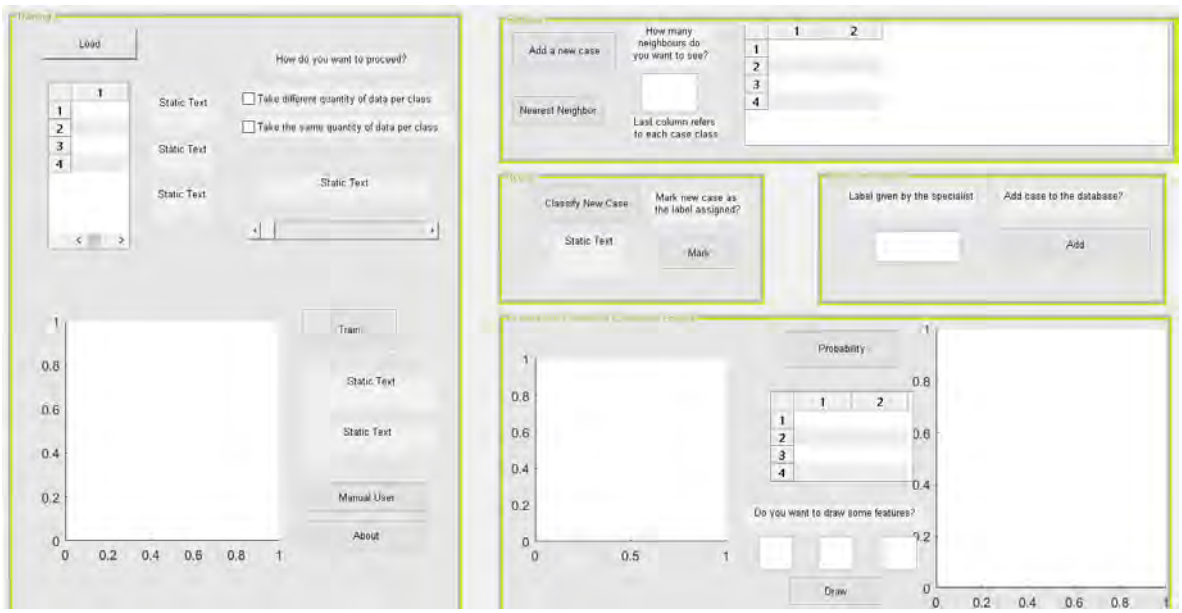
Este apartado contiene la descripción detallada de la interfaz creada para la interacción entre el Sistema CBR y el usuario, el cual, por este medio, obtendrá un mejor entendimiento de la interfaz y dará un mejor uso y rendimiento de la misma.

:

Las características principales de esta interfaz son:

- Entrenamiento de los clasificadores Random Forest, SVM y Parzen en secuencia por medio de una base de datos escogida por el usuario
- Visualización de la información con la que se trabaja, es decir, cantidad de datos a usar, porcentajes para entrenamiento y prueba de los clasificadores y porcentaje de probabilidad de pertenencia a cada caso.
- Guardado de nuevos casos añadidos en la interfaz para su próximo uso en entrenamiento o su descarte de acuerdo a la necesidad y situación del nuevo caso.

Figura 12. Interfaz CBR propuesta. Se muestran sus diferentes secciones, la cuales son: Training, Retrieve, Reuse, Revise and Retain y Probability and Feature Drawing.



La Interfaz de CBR propuesta (Figura 20) tiene: botones para la carga, visualizacion y propocision de porcentajes son usados para el entrenamiento, un boton para la adicion de un nuevo caso a analizar y un boton para obtener la etiqueta dada por el clasificador entrenado junto con su respectivo guardado o visualizacion de la probabilidad de pertenencia a cada clase de ese nuevo caso.

- Como resultados se obtienen de la interfaz
    - El numero de clases, atributos y cantidad de datos total de la base de datos cargada.
    - Error de los clasificadores una vez entrenados.
    - Vecinos cercanos al nuevo caso añadido de acuerdo al usuario.
    - Etiqueta del clasificador.
    - Un archivo en formato excel conteniendo la base de datos original junto con los nuevos casos añadidos.

- La interfaz permite:

    - Visualizacion de la cantidad de datos a usar en el proceso de entrenamiento y prueba.
    - Visualizacion de la cantidad de porcentaje destinado al entrenamiento y prueba de los clasificadores bi-clase y mul-ticlase.

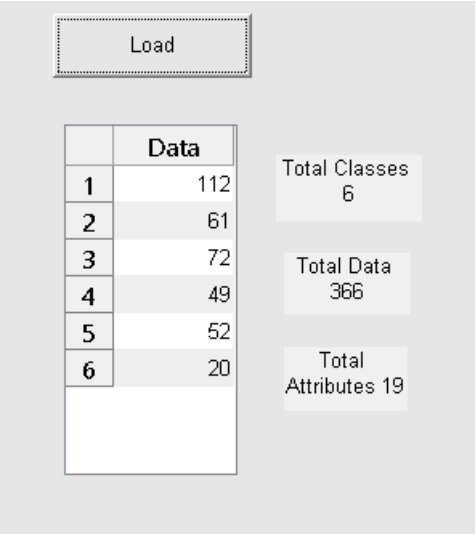
- Decidir el numero de vecinos cercanos al nuevo caso.
- Decidir si la clase asignada por el clasificador pertenece o no a la descripcion del nuevo caso.
- Visualizacion de la probabilidad perteneciente a cada clase dentro de la base de datos.
- Decidir añadir el nuevo caso a la base original

Uso de la interfaz:

- Inicialmente se añaden la base de datos (Botón “Load”, Figura 21) a usar en el entrenamiento de los clasificadores, la base de datos debe introducirse inicialmente un Excel conteniendo las características de la base de datos seguido de otro Excel con las clases pertenecientes.

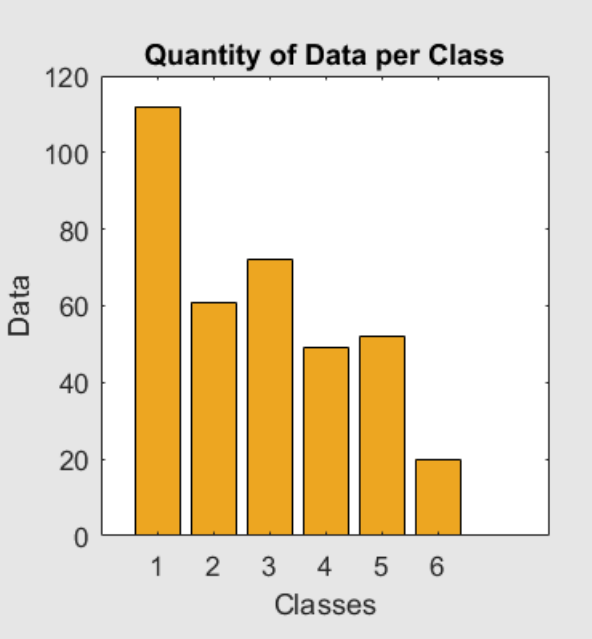
- En una tabla aparte (Data, Figura 21) se mostrarán la cantidad de atributos por clase la cantidad de datas en la base y las clases existentes.

Figura 13. Carga de datos. Está compuesta por un botón “Load”, una tabla de visualización que permite ver las clases y cantidades de datos y tres cuadros de texto donde se visualizan las principales características de las bases de datos cargadas.



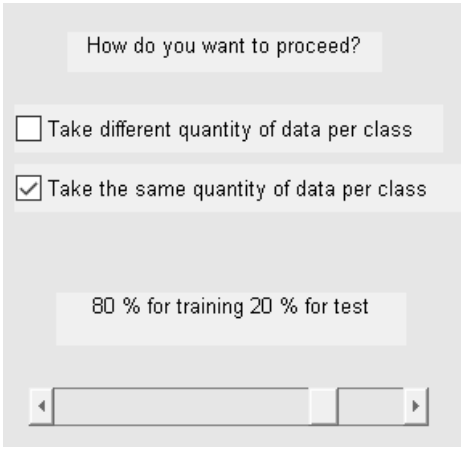
- Al igual que con la tabla la cantidad de datos se puede visualizar por medio de un gráfico (Figura 22).

Figura 14. Visualización base de datos original. Se muestran un gráfico de barras con la cantidad de registros por cada clase.



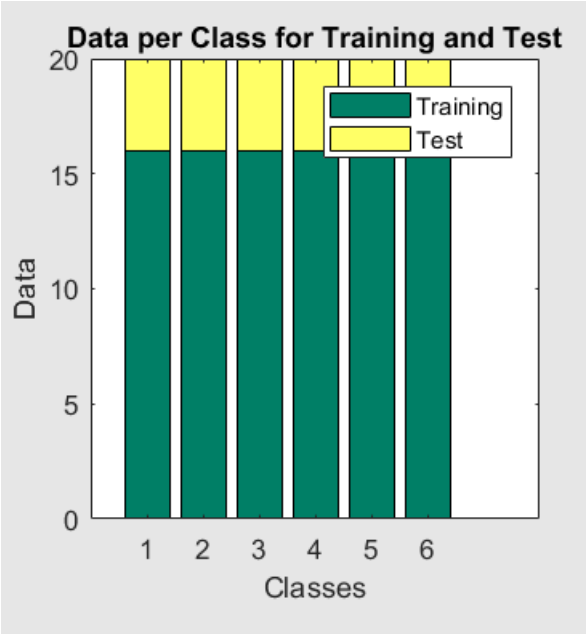
- Una vez añadido la cantidad de datos se decide por medio de las barras de movimiento o sliders(Figura 23) la cantidad de datos para prueba y entrenamiento.

Figura 15. Cantidad de información a usar para entrenamiento y prueba del clasificador. Tiene dos opciones, se puede tomar la misma cantidad de registros o diferente en el caso que la base de datos no este balanceada (Tomando una cantidad de datos proporcional).



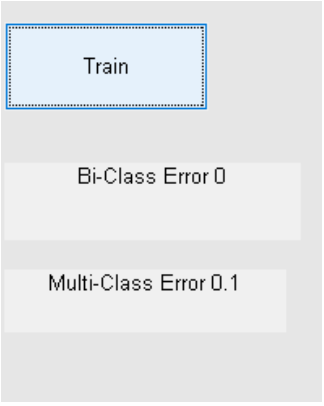
- Igualmente, para comodidad del usuario se decide mostrar visualmente la cantidad de porcentaje a usar para cada tarea.

Figura 16. Visualización información de entrenamiento y prueba para clasificador para cada cantidad de casos por clase en la base de datos introducida.



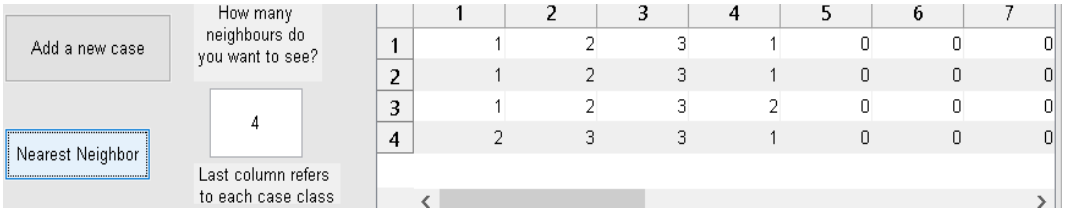
- Una vez estos parámetros han sido elegidos se decide por entrenar los clasificadores (Botón “Train, Figura 25”) y se visualiza el error de los dos clasificadores.

Figura 17. Entrenamiento clasificador, el cual entrega dos resultados pertenecientes al error por entrenamiento de los dos clasificadores bi-clase como multi-clase.



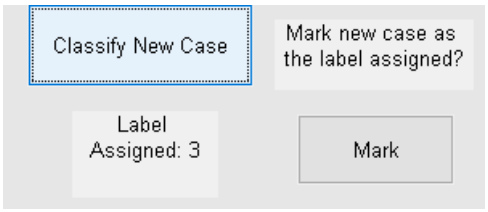
- El botón “Add case”(Figura 26) permite añadir un nuevo caso a analizar junto con la caja en blanco la cual permite escribir el número de vecinos cercanos a encontrar junto con la visualización de estos en una tabla al lado.

Figura 18. Adición de nuevo caso y observación de sus vecinos más cercanos de acuerdo a la cantidad deseada por el usuario la última columna corresponde a la clase perteneciente a cada caso.



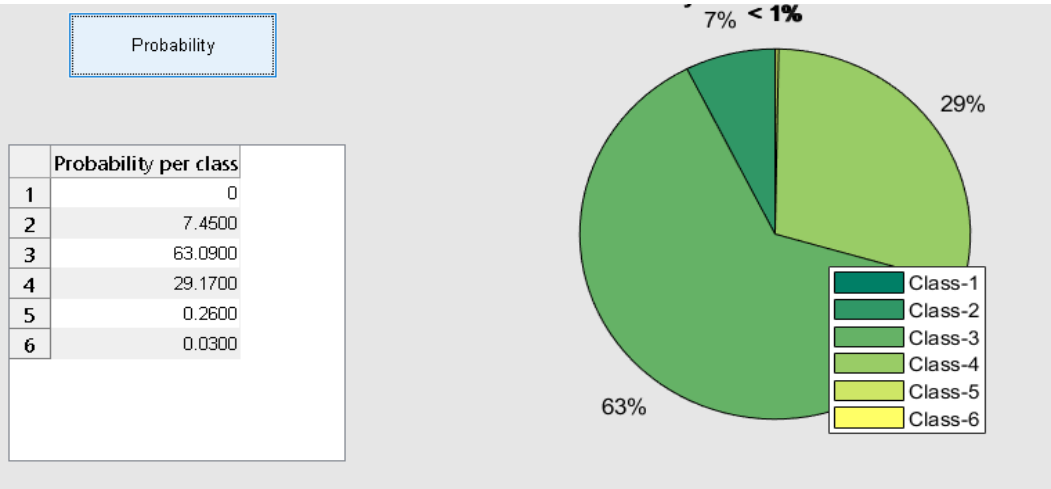
- El botón “Classify” new case (Figura 27) permite determinar la etiqueta asignada por los clasificadores al igual que el botón “Mark” si se desea marcar el caso con esa clase sin necesidad de visualizar la probabilidad de pertenencia.

Figura 19. Etiqueta entregada por el sistema entrenado, si se desea, se puede etiquetar de una vez el caso con el botón Mark sin necesidad de escribir la clase la cual se piensa pertenece el caso.



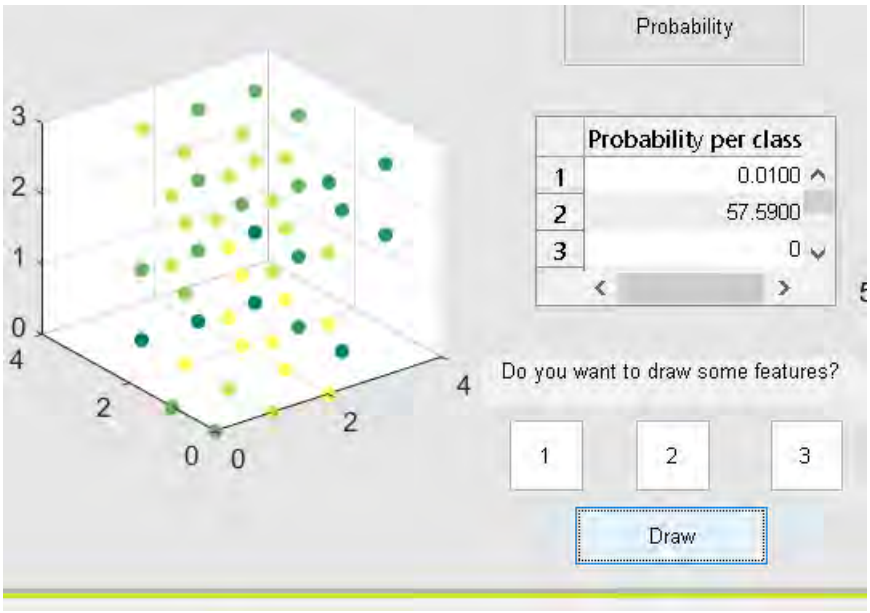
- Con el botón “Probability” (Figura 28) se puede visualizar la probabilidad de pertenencia de cada caso a cada clase.

Figura 20. Probabilidad de pertenencia del nuevo caso a todas las clases, están se muestran en una tabla y en un gráfico circular con sus respectivos porcentajes.



- Si se desea se puede observar el comportamiento de diferentes características dentro de la base de datos en un plot en 3D de 3 características que el usuario desee, las características se digitan en los cuadros en blanco y al finalizar se presiona el botón “Draw” (Figura 29).

Figura 21. Grafica de 3 características que el usuario desee observar, estas deben estar contenidas dentro de la cantidad de características presentes dentro de la base de datos.



- El ultimo botón con su caja respectiva permite al usuario colocar una clase con la cual se cree es asignado el nuevo caso añadido al igual que si desea guardar el nuevo caso en la base de datos original.

Figura 22. Guardado del nuevo caso y etiqueta dada por el especialista, si se presiona el botón Mark anteriormente esta casilla quedara inhabilitada.

7. CONCLUSIONES Y TRABAJO FUTURO

- Nuevamente en esta investigación se comprueba que los sistemas CBR son una herramienta eficaz para el apoyo al ámbito médico. Los resultados generales demuestran que la exactitud y precisión dados por los CBR, brindan la información adecuada para que en conjunto con los conocimientos de un especialista se tome la decisión acertada en un escenario como el diagnóstico médico.

- Se logró evidenciar que la implementación de la clasificación en cascada en las etapas de adaptación y recuperación de un CBR trae una disminución del error con respecto a un clasificador individual. Cabe resaltar que dichos valores alcanzaron una media de error igual a 0, esto comprueba que los sistemas multi-clasificadores, como la cascada, son una buena opción en torno a mejorar la clasificación para bases de datos con un grado de complejidad alto. Sin dejar a un lado la buena representación de los datos, resultado de involucrar métodos de selección y balanceo para facilitar la clasificación. Este resultado en el campo del Aprendizaje Automático puede servir para involucrar más temas de pre-procesamiento y combinación de clasificadores en estudios que buscan una exigencia más alta debido al objetivo o campo para el cual es realizado dicho análisis.

- Tal como esta investigación lo ha demostrado la estimación de probabilidad con SVM tiene un alto porcentaje de acierto, que en conjunto con un buen resultado de clasificación en sistemas en cascada proporciona un resultado coherente a la hora de asignar clases o etiquetas a nuevos casos encontrados. Es de destacar que este estimador produjo mejores resultados de pronostico que los basados en ventanas de Parzen y KNN, que tienen una gran información y son frecuentemente citados por la comunidad científica. Este resultado sugiere investigar otros tipos de estimación de probabilidad que los que convencionalmente se utilizan, esto proporcionaría mejores herramientas en campos como la Probabilidad y el Aprendizaje Automático, generando una mayor diversidad para trabajar con un estimador.

- Finalmente se evidencia que la integración de la Clasificación en cascada y Estimación probabilidad basada en SVM dentro de un ciclo CBR entrega unos resultados favorables, debido a que el buen resultado de pronóstico de una clase afecta de manera favorable la etapa de revisión y retención de estos sistemas. Esto en el contexto de diagnóstico médico trae ventajas, ya que el especialista tendrá una información más concluyente que puede servir de apoyo para dictaminar de forma más adecuada un diagnóstico. En el campo del Aprendizaje Automático invita a la investigación de formas

diversas de implementar métodos más sofisticados, como metodologías diferentes de implementación de estos sistemas con la fusión de algunas de sus etapas, o la utilización de métodos de clasificación más complejos.

Teniendo en cuenta las anteriores conclusiones se puede proponer varias ideas de trabajo futuro en esta área de investigación. Un abordaje de los sistemas CBR es a través de sistemas multi-clasificación, en este trabajo fue utilizada la arquitectura en cascada, pero existen otras arquitecturas que pueden ser evaluadas y pueden producir resultados óptimos, los cuales pueden incluirse en la etapa de recuperación del CBR. Otro aspecto que puede ser objeto de estudio es enfocarse en procesos de probabilidad para estimar con la mayor exactitud la clase real, para esto se recomienda incursionar en estimadores basados en árboles de decisión como el Random Forest. Por último, otra aproximación puede hacerse con los métodos de selección y balanceo, puesto que existen diferentes estrategias con que se pueden abordar estos problemas. Cabe mencionar que la ventaja de dichos métodos es que pueden incluirse en cualquier metodología de aprendizaje automático.

BIBLIOGRAFÍA

[1] HERRERO, Juárez, MANUEL, José. Una aproximación Multimodal al Diagnostico Temporal Mediante Razonamiento Basado en Casos y Razonamiento Basado en Modelos. Aplicaciones en Medicina e Inteligencia Artificial. En: Revista Iberoamericana de Inteligencia Artificial. 2007. vol. 11, no. 36, p. 77- 80.

[2] LOZANO, Laura, FERNÁNDEZ, Javier. Razonamiento Basado en Casos: Una Visión General. Universidad de Vlladolid, España. 2008.

[3] AADMOT, Agnar, PLAZA, Enric. Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches. AI Communications. IOS Press.1994. vol. 7, no. 1, p. 39-59.

[4] RODRÍGUEZ, CUADRADO, Santiago, RODRÍGUEZ, GONZÁLEZ, Emilio, HERNÁNDEZ, CURBELO, Haydee, CARVAJAL, Yaquelin, CASAS, Gladys, GUTIÉRREZ, Iliana. Sistema experto basado en casos para el diagnóstico de la hipertensión arterial. Revista Facultad de Ingeniería Universidad de Antioquia. Septiembre 2011. No.60, p. 202-213.

[5] LOPÉZ, Edwin. Inteligencia Artificial Aplicada al Diagnóstico Médico. Ingeniería de Sistemas y Computación, Universidad Nacional de Colombia, Sede Bogotá. p. 1-2.

[6] VALENCIA, Xiomara, Patricia. Sistema genérico de razonamiento basado en casos (CBR) multi-clase como soporte al diagnostico médico mediante técnicas de reconocimiento de patrones.Tesis Doctoral.Salamanca, España. Universidad de Salamanca. Facultad de Ciencias. Departamento de informática y automática.2017. p. 1-3.

[7] BREGÓN, Anibal, RODRÍGUEZ, Jose. Un sistema de razonamiento basado en casos para la clasificación de fallos en sistemas dinámicos. Taller Nacional de Minería de Datos y Aprendizaje.2005. vol. 3, p. 1-2.

[8] HURTADO, Teobaldo. Diagnóstico médico. Biociencias. 2016. vol. 11, no.1, p. 3-4.

[9] SCHANK, Roger, ABELSON, Robert. Scripts, plans, goals and understanding: an inquiry into human knowledge structures. Yale University. New Haven, CT. 1977. no. 38, p. 248.

[10] KOLODNER, Janet. Reconstructive Memory: A Computer Model. Cognitive Science. 1983. vol. 7, p. 287-328.

[11] ARJONA, Miguel, Manuel. Estudio para la implementación de un sistema de razonamiento basado en casos.2006. Universidad de Rovira. p. 7.

[12] BERGMANN, Ralph, BREEN, Sean, GÖKER, Mehmet. Methodology for Building and Maintaining CBR Applications. 1998. INRECA, vol. 2.

[13] BENÍTEZ, Ignacio, DIEZ, Jose, Luis.Técnicas de Agrupamiento para el Análisis de Datos Cuantitativos y Cualitativos. Valencia, España, 2005.

[14] QUISPE, José, YARI, Yessenia. Modelo Estocástico a partir de Razonamiento Basado en Casos para la Generación de Series Temporales. XII Congreso de la Sociedad Peruana de Computación. Arequipa, Perú.

[15] GETIAL, Jesús, ERAZO, David, PANTOJA, Andres. A quantitative comparison of path planning methods in mobile robotics. Automatic Control (CCAC), IEEE 3rd Colombian Conference.2017. p. 10.

[16] SHWARTZ, Saleb, BEN, David. UNDERSTANDING MACHINE LEARNING From Theory to Algorithms. Cambridge University Press. 2014.

[17] WAGSTAFF, Kiri, ROGERS, Seth. Constrained K-means Clustering with Background Knowledge. Proceedings of the Eighteenth International Conference on Machine Learning. 2001. p. 577–584.

[18] HERRERA, Ignacio, FIGUEROA, Alejandro. Aprendizaje Semi-Supervisado de Multiples Vistas para Detectar Tmporalidad de Preguntas. Santiago, Chile. 2016.

[19] KOLODNER, Janet. Maintaining organization in a dynamic long-term memory. *1983.* Cognitive Science. p. 243–280.

[20] TING, S. L. KWOK, S. K. TSANG, Albert, LEE W.B. A hybrid knowledge-based approach to supporting the medical prescription for general practitioners: Real case ina hong kong medical center. Knowledge-Based

Systems**.** Department of Industrial and Systems Engineering, The Hong Kong Polytechnic University.2011.vol. 24, no. 3, p. 444 – 456.

[21] FAN, Chin-Yuan,CHANG, Pei-chann, LIN, Jyun-Jie.A hybrid model combining case-based reasoning and fuzzy decision tree for medical data classification. 2011.vol. 11, no. 1, p. 632 – 644.

[22] DUVAL-POO, Miguel,VEGA-POONS, Sandro, RUIZ-SHULCLOPER, José.Combinación de clasificadores supervisados: estado del arte. CENATAV. 2012. Series Azules.

[23] LI, Yu.Knowledge integration in a multiple classifier system. 1996. Applied Intelligence, vol. 6, no. 2, p. 75-86.

[24] MAUDES, Jesús.Combinación de clasificadores: construcción de características e incremento de la diversidad. Universidad de Burgos, Burgos, España, 2010.

[25] HERRERA, Francisco, CANO, J.R.Técnicas de reducción de datos en KDD. El uso de Algoritmos Evolutivos para la Selección de Instancias. *2006.*Actas del I Seminario Sobre Sistemas Inteligentes. p. 165-181.

[26] LANGLEY, Pat. Selection of relevant features in machine learning. 1994. Proceedings of the AAAI Fall symposium on relevance. p. 245-271.

[27] RIQUELME, José, Cristóbal, RUIZ, Roberto, GILBERT, Karina. Minería de datos: Conceptos y tendencias.2006. Inteligencia artificial: Revista Iberoamericana de Inteligencia Artificial, vol. 10, p. 11-18.

[28] FERNÁNDEZ, Carlos. Estadistica Avanzada. Quimiometría. Valencia: Universidad de Valencia, 2011.

[29] HALL, Mark. Correlation-based feature selection for machine learning. Universidad de Waikato. 1999.

[30] FRANK, Eibe, HOLMES, Geoffrey, HALL, Mark. The WEKA data mining software: An update. Exploration.2009. vol. 11, no. 1, p. 10–18.

[31] DINAKARAN, S. THANGAIAH, Ranjit. Role of Attribute Selection in Classification. International Journal of Scientific & Engineering Research. 2013. vol. 4, p. 67-71.

[32] PUENTE-, Liliana,LÓPEZ, Asdrubal,CRUZ, William.Método rápido de preprocesamiento para clasificación en conjuntos de datos no balanceados. *2014.*Research in Computing Science, vol. 73, p. 129-142.

[33] VIVEROS, Diana, ORTEGA, Mabel.SISTEMA DE RAZONAMIENTO BASADO EN CASOS COMO SOPORTE ALDIAGNÓSTICO MÉDICO MEDIANTE CLASIFICACIÓN DE DATOS MULTI-CLASE.Universidad de Nariño. Pasto-Nariño Colombia. 2017.

[34] BECKMANN, Marcelo, EBECKEN, Nelson, DE LIMA, Beatriz. A KNN Undersampling Approach for Data Balancing. 2015. Journal of Intelligent Learning Systems and Applications. No. 7, p. 104-116.

[35] CHAWLA, N. V. BOWYER, K. W. HALL, L. O. KEGELMEYER, W. P. Synthetic Minority Over-sampling Technique. *2002.* Journal of Artificial Intelligence Research, vol. 16, p. 321-257, 2002.

[36] BATISTA, Gustavo, PRATI, Ronaldo, MONARD, Maria. A Study of the Behavior of Several Methods for Balancing Machine Learning Training Data. 2004. ACM SIGKDD Explorations Newsletter, vol. 6, p. 20-29.

[37] WITTEN, Ian, FRANK, Eiben. Data Mining: Practical Machine Learning Tools and Techniques. Second Edition. San Francisco: Morgan Kaufmann, 2005.

[38] BRAVO, Cristian, L'Huillier, LOBATO, Luis, WEBER, Richard. Probability estimation for multiclass problems combining SVMs and neural networks.2009. Neural Network World, pp. 475-489.

[39] HENAO, Ricardo.Selección de hiperparámetros en máquinas de soporte vectorial. Universidad Nacional de Colombia. 2004.

[40] HSU, Wei, LIN, Jen. A comparison of methods for multiclass support vector machines. 2002. IEEE Transactions on Neural Networks, no. 13, p. 415-425.

[41] LEI, Hansheng, GOVINDARAJU, Venu. Half-against-half multi-class support vector machines. 2005. Lecture Notes in Computer Science, Springer. p. 156-164.

[42] RIFKIN, Ryan, KLATAU, Aldebaro. In Defense of One-Vs-All Classification.2004. Journal of Machine Learning Research, p. 101-141.

[43] WESTON, J. WATKINS, C. Multi-class support vector machines. Technical Report CSD-TR9800-04, 9800-04, Department of Computer Science, Royal Holloway, University of London, London, 1998.

[44] KONG, Bae, DIETTERICH, Thomas. Probability estimation via error-correcting output coding. *IASTED International Conference*.1997. Canada.

[45] PLATT, John.Probabilistic outputs for support vector machines and comparison to regularize likelihood method. Advances in Large Margin Classifler.2000., p. 61-74.

[46] ZADROZNY, Bianca, ELKAN, Charles. Transforming classifler scores into accurate multiclass probability estimates.2002. Proceedings of the eighth ACM SIGKDD international conference, pp. 694-699.

[47] AUBONE, Anibal, WÖHLER, Otto.Aplicacion del metódo de maxima verosimilitud a la estimación de parametros y comparacion de curvas de crecimiento de Von Bertalanffy. Printed Argentine, Mar de Plata, Argentina, 2000.

[48] WU, Fang, LIN,Jen, WENG, Ruby. Probability estimates for multi-class classiflcation by pairwise coupling. Journal of Machine Learning Research. 2004 no. 5, p. 975-1005.

[49] VAPNIK, Vladimir.The Nature of Statistical Learning Theory (Information Science and Statistics. Springer, 1999.

[50] HASTIE, Trevor, TIBSHIRANI, Robert.Classiflcation by pairwise coupling. Proceedings of the 1997 conference on Advances in neural information processing systems 10,1998. p. 507-513.

[51] PRICE, David, KNERR, Stefan, PERSONNAZ, Léon, DREYFUS, Gérard.Pairwise neural network classiflers with probabilistic outputs. 1994.MIT press. p. 1109-1116.

[52] BREUNIG, Robert. Nonparametric density estimation for stratified samples. The Australian National University working papers in economics and econometrics, 2005.

[53] HUANG, Chen, DING, Xiaoqing, FANG, Chi. Head Pose Estimation Based on Random Forests for Multiclass Classification. International Conference on Pattern Recognition. 2010. p. 934-937.

[54] ZOU, Hui, HASTIE, Trevor. Regularization and variable selection via the elastic net.Stanford University.USA. 2005.

[55] HODGES, Joseph, Fix, E.An Important Contribution to Nonparametric Discriminant Analysis and Density Estimation. International Statistical Review. Vol. 57, no. 3, p. 233-238.

[56] MARTÍNEZ, Wendy, MARTÍNEZ, Angel.Computational Statistics Handbook with MATLAB. New York: CHAPMAN & HALL/CRC, 2002.

ANEXOS

Anexo A. Graficas de Experimentación Adicionales

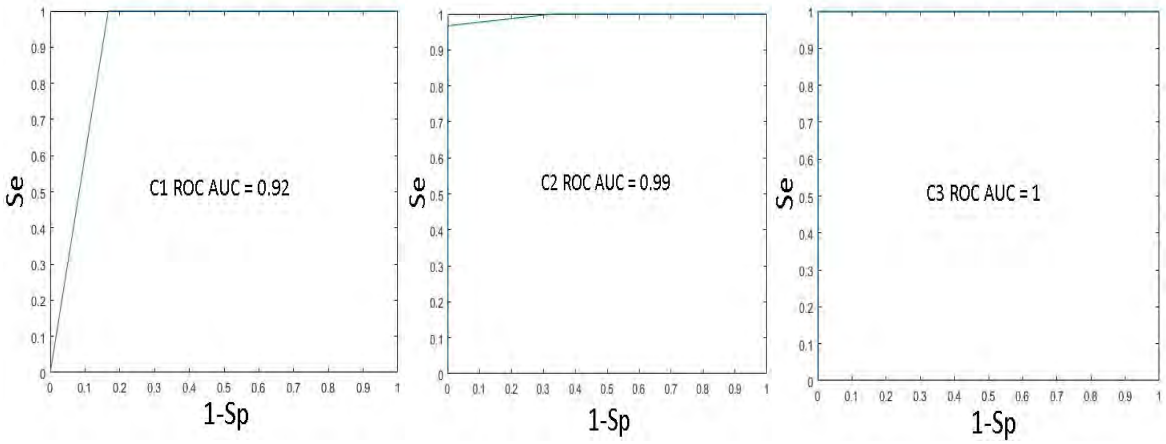
1) METODOS DE BALANCEO

En esta sección se presentan los diferentes resultados obtenidos por distintos métodos de balanceo descritos en las pruebas 2 a 4 en resultados sección 6

- **Experimento 2**

- *Curvas ROC*

Figura 1. Curvas ROC de las 6 clases de Dermatología para el experimento 2. Curvas ROC de las 6 clases de la base de datos de Dermatología, nombradas C1, C2, C3, C4 y C5, donde se muestra el valor de su respectivo AUC (Área bajo la curva), para el experimento 2.



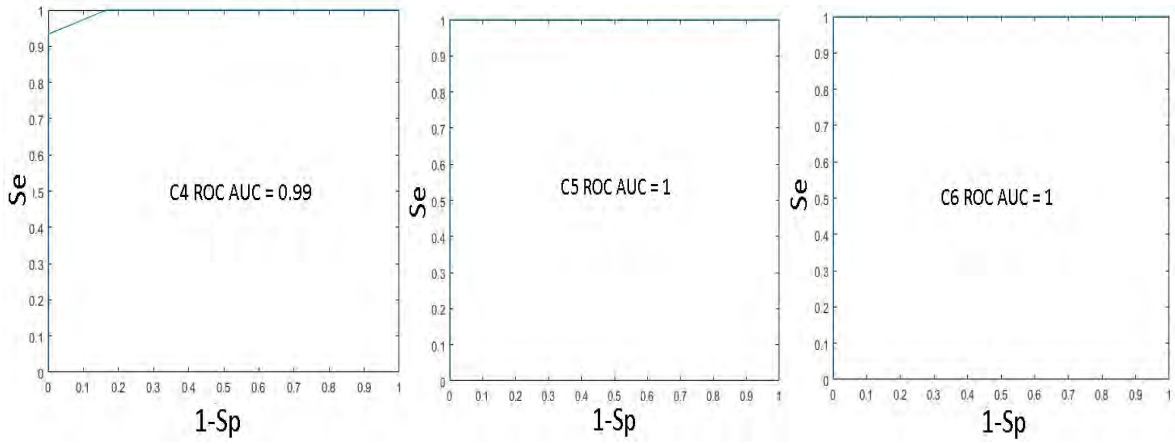
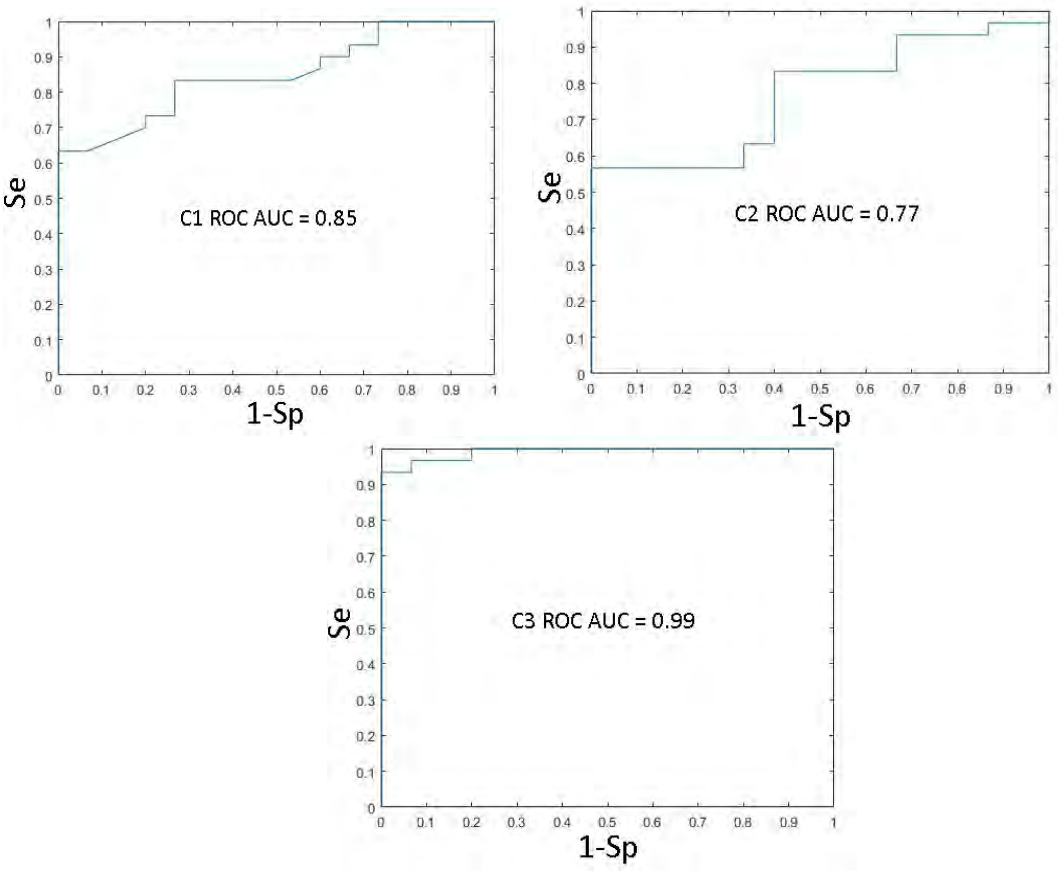


Figura 2. Curvas ROC de las 3 clases de Hipotiroidismo para el experimento 2. Curvas ROC de las 3 clases de la base de datos de Hipotiroidismo, nombradas C1, C2, y C3, donde se muestra el valor de su respectivo AUC (Área bajo la curva), para el experimento 2.



- *Tablas Se y Sp*

Tabla 21. Tablas Se y Sp para Dermatología, experimento 2

Medida	**Clase 1**	**Clase 2**	**Clase 3**	**Clase 4**	**Clase 5**	**Clase 6**	**Promedio**
Se	0,98	0,89	0,99	0,86	0,99	1,00	0,98
Sp	1,00	0,97	1,00	0,97	1,00	1,00	1,00

Tabla 22. Tablas Se y Sp para Hipotiroidismo, experimento 2

Medida	**Clase 1**	**Clase 2**	**Clase 3**	**Promedio**
Se	0,71	0,61	0,92	0,75
Sp	0,81	0,82	0,97	0,87

- *Matriz de confusión*

Tabla 23. MC de confusión, Dermatología, experimento 2

	1	2	3	4	5	6
1	1178	8	0	6	4	4
2	0	1063	0	136	1	0
3	0	2	1192	6	0	0
4	0	166	0	1034	0	0
5	0	3	0	5	1192	0
6	6	0	0	0	0	1194

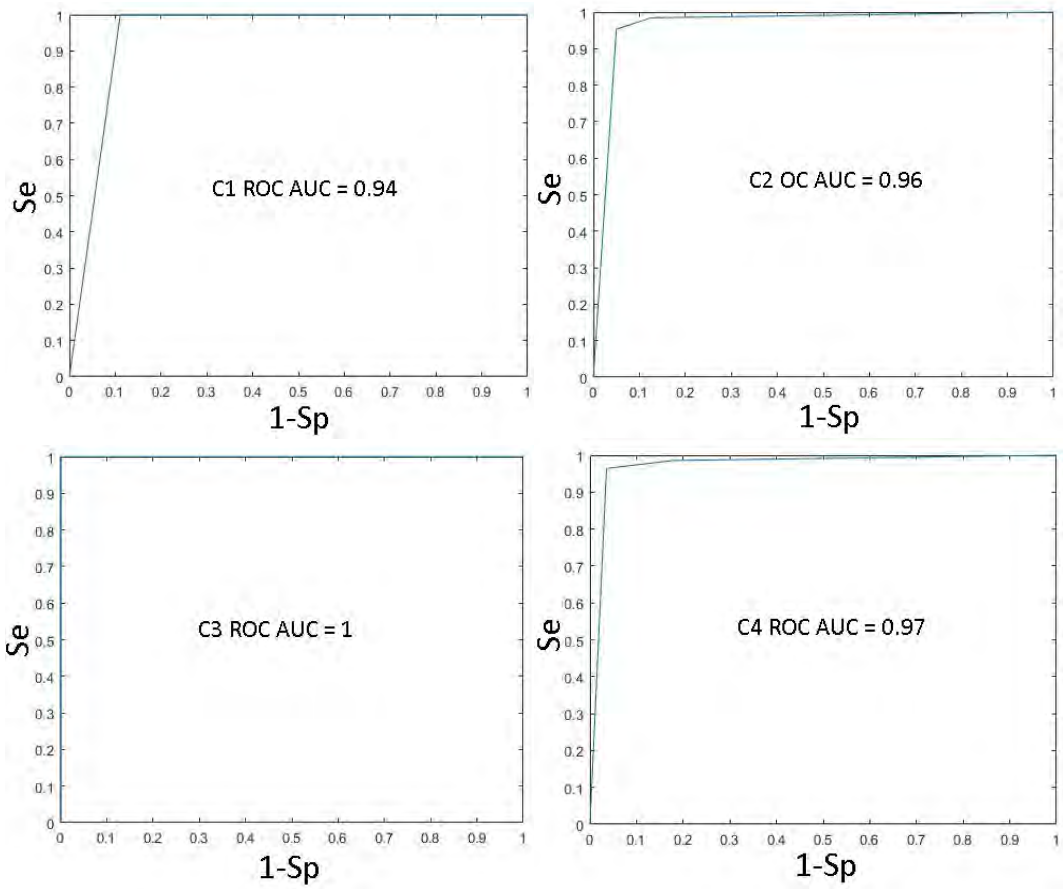
Tabla 24. Matriz de confusión, Hipotiroidismo, experimento 2

	1	2	3
1	2133	825	42
2	1067	1837	96
3	18	225	2757

➢ Experimento 3

- *Curvas ROC*

Figura 3. Curvas ROC de las 6 clases de Dermatología para el experimento 3. Curvas ROC de las 6 clases de la base de datos de Dermatología, nombradas C1, C2, C3, C4 y C5, donde se muestra el valor de su respectivo AUC (Área bajo la curva), para el experimento 3



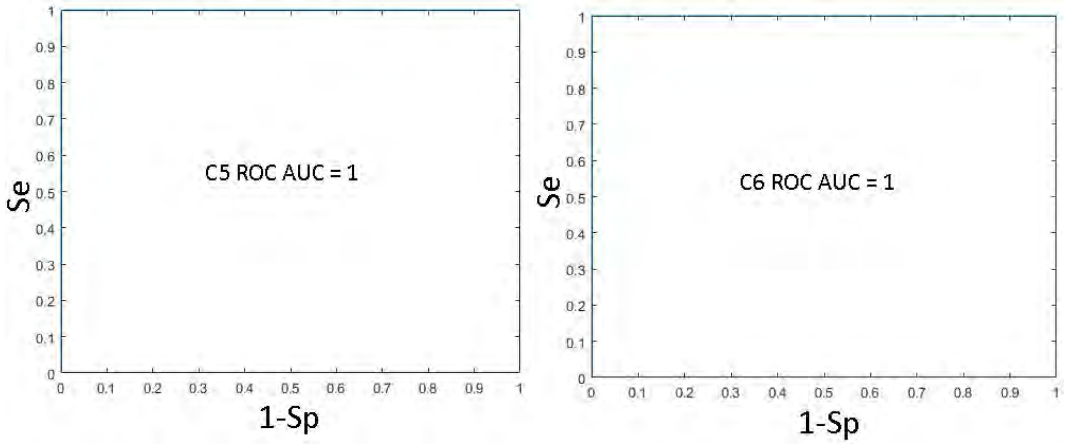
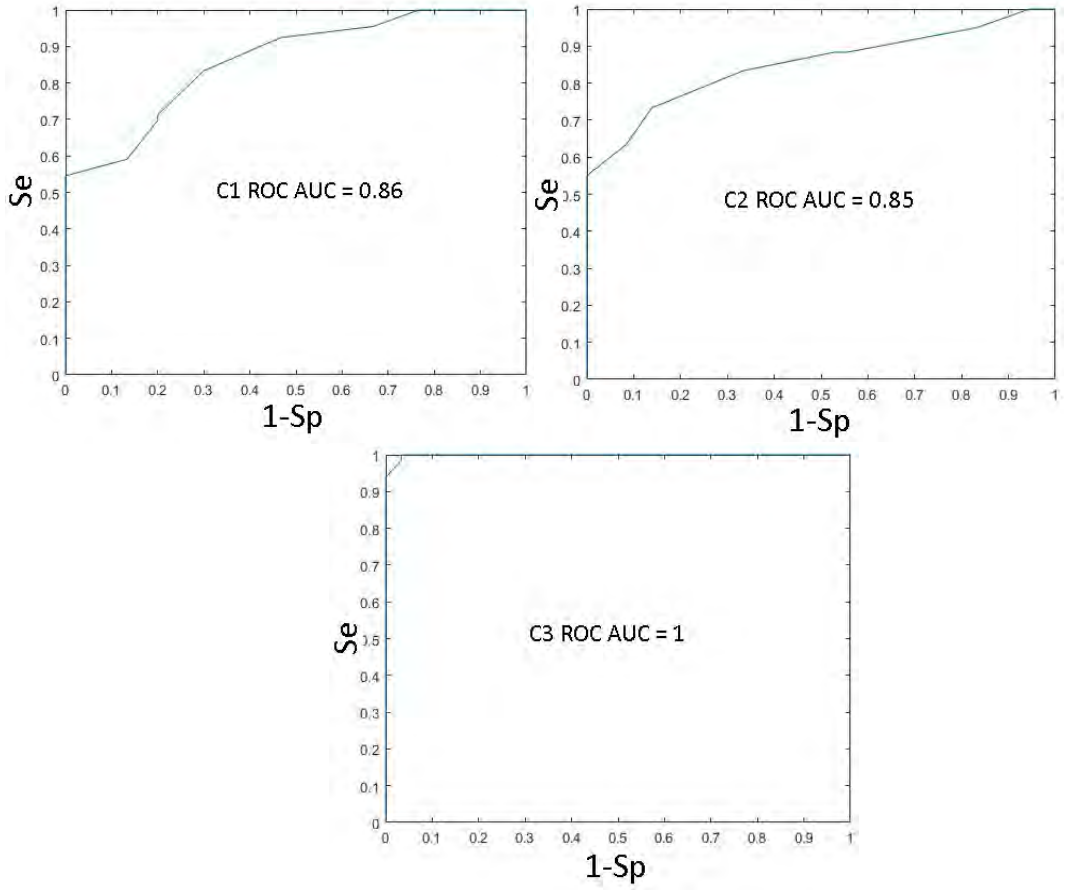


Figura 4. Curvas ROC de las 3 clases de Hipotiroidismo para el experimento 3. Curvas ROC de las 3 clases de la base de datos de Hipotiroidismo, nombradas C1, C2, y C3, donde se muestra el valor de su respectivo AUC (Área bajo la curva), para el experimento 3



- *Tablas Se y Sp*

Tabla 25. Tablas Se y Sp para Dermatología, experimento 3

Medida	**Clase 1**	**Clase 2**	**Clase 3**	**Clase 4**	**Clase 5**	**Clase 6**	**Promedio**
Se	0,99	0,90	1,00	0,88	1,00	1,00	0,99
Sp	1,00	0,97	1,00	0,97	1,00	1,00	1,00

Tabla 26. Tablas Se y Sp para Hipotiroidismo, experimento 3

Medida	**Clase 1**	**Clase 2**	**Clase 3**	**Promedio**
Se	0,72	0,67	0,95	0,78
Sp	0,82	0,84	0,98	0,88

- *Matriz de confusión*

Tabla 27. MC para Dermatología, experimento 3

	1	2	3	4	5	6
1	5970	18	0	1	0	11
2	0	5757	0	243	0	0
3	0	0	5999	1	0	0
4	0	347	0	5653	0	0
5	0	0	0	0	6000	0
6	0	0	0	0	0	6000

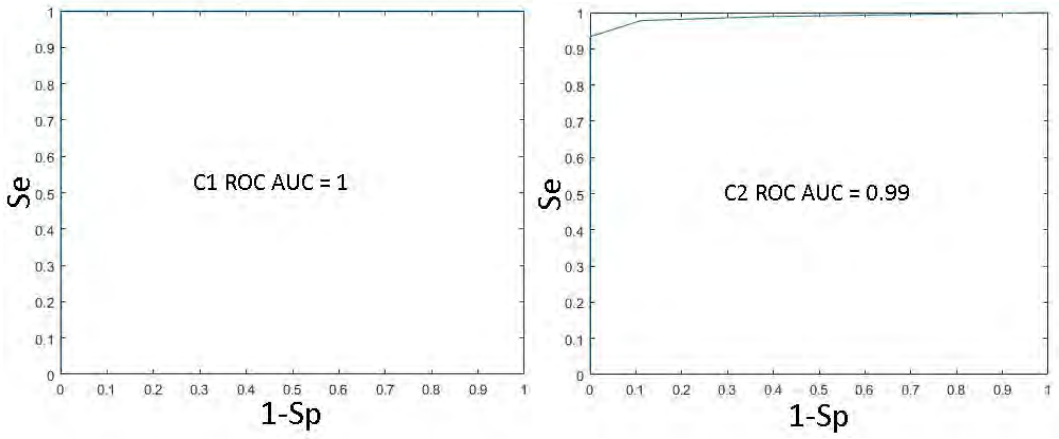
Tabla 28. Matriz de confusión, Hipotiroidismo, experimento 3

	1	2	3
1	4308	1654	38
2	2231	4848	121
3	38	287	5675

- Experimento 4

- *Curvas ROC*

Figura 5. Curvas ROC de las 6 clases de Dermatología para el experimento 4. Curvas ROC de las 6 clases de la base de datos de Dermatología, nombradas C1, C2, C3, C4 y C5, donde se muestra el valor de su respectivo AUC (Área bajo la curva), para el experimento 4



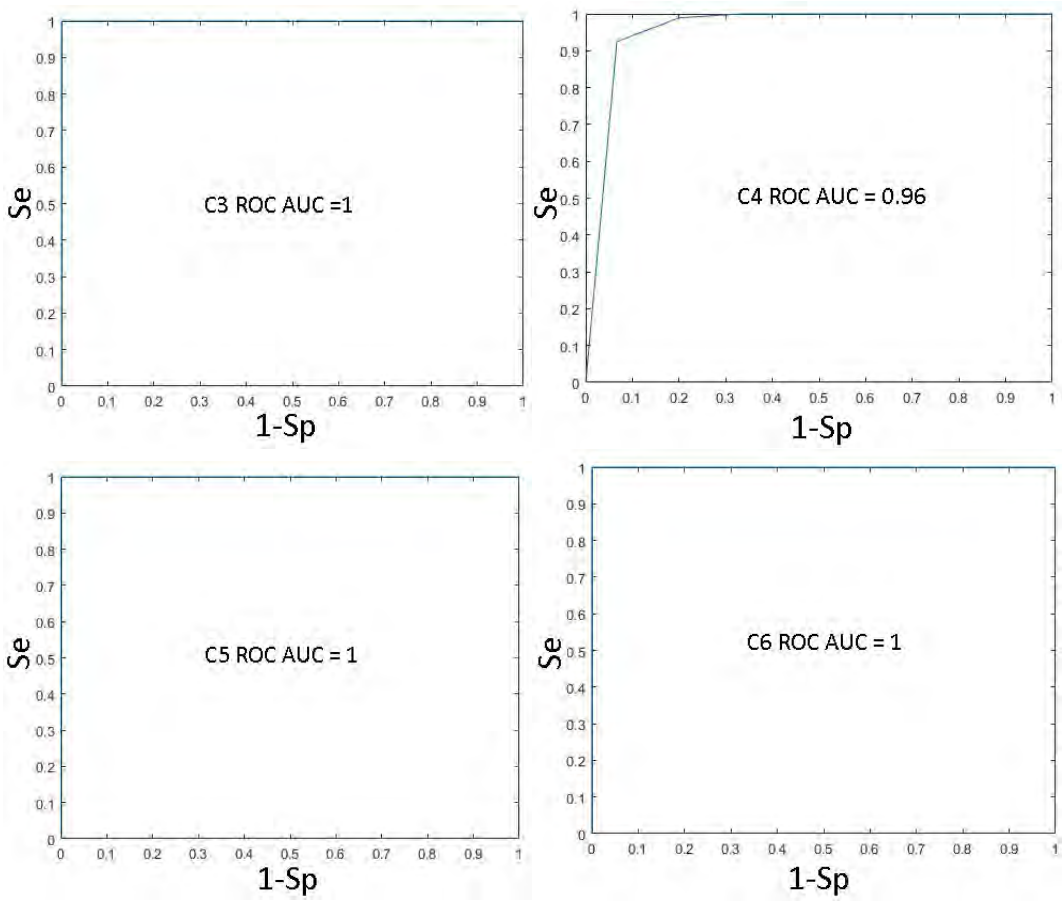
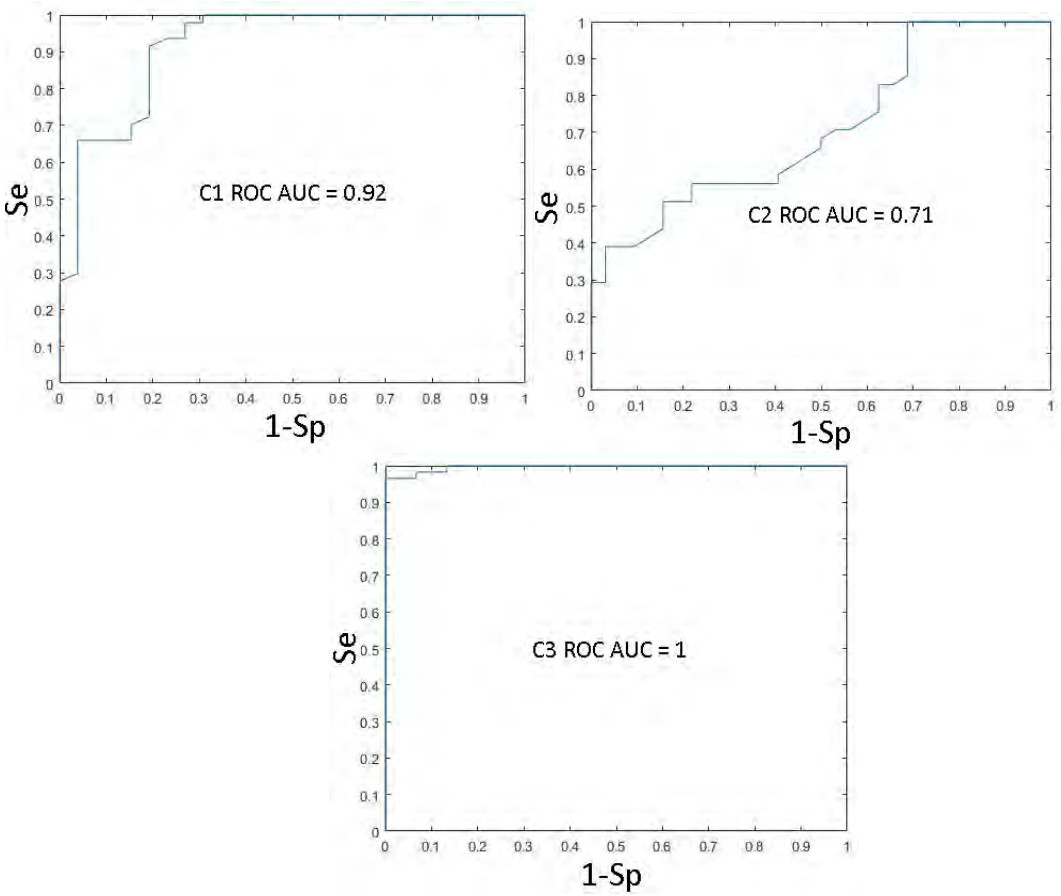


Figura 6. Curvas ROC de las 3 clases de Hipotiroidismo para el experimento. Curvas ROC de las 3 clases de la base de datos de Hipotiroidismo, nombradas C1, C2, y C3, donde se muestra el valor de su respectivo AUC (Área bajo la curva), para el experimento 4



- *Tablas Se y Sp*

Tabla 29. Tablas Se y Sp para Dermatología, experimento 4

Medida	**Clase 1**	**Clase 2**	**Clase 3**	**Clase 4**	**Clase 5**	**Clase 6**	**Promedio**
Se	1,00	0,94	1,00	0,85	1,00	0,99	1,00
Sp	1,00	0,97	1,00	0,99	1,00	1,00	1,00

Tabla 30. Tablas Se y Sp para Hipotiroidismo, experimento 4

Medida	**Clase 1**	**Clase 2**	**Clase 3**	**Promedio**
Se				

	0,84	0,86	0,90	0,86
Sp	0,91	0,87	0,98	0,92

- *Matriz de confusión*

Tabla 31. Matriz de confusión, Dermatología, experimento 4

	1	2	3	4	5	6
1	6200	0	0	0	0	0
2	0	3367	0	233	0	0
3	0	0	4400	0	0	0
4	0	459	0	2541	0	0
5	0	0	0	2	3198	0
6	0	15	0	0	0	1185

Tabla 12. Matriz de confusión, Hipotiroidismo, experimento 4

	1	2	3
1	4342	788	70
2	763	5498	139
3	84	219	2697

2) Experimentación Cascadas de clasificadores

En la siguiente sección se muestran las figuras de los experimentos 3 al 5 respecto a la implementación de clasificadores en cascada:

- Experimento 1: Cascada simple

Figura 7. Cascada doble, error por clasificador y tiempo, Dermatología.

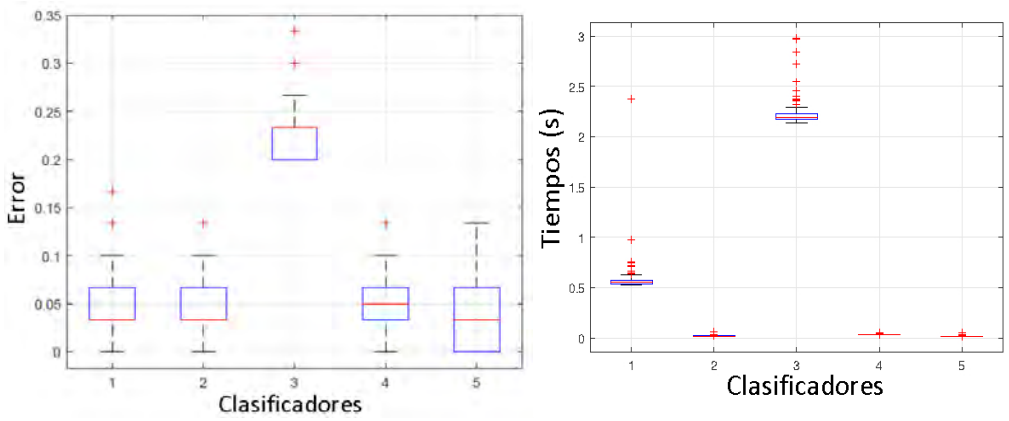
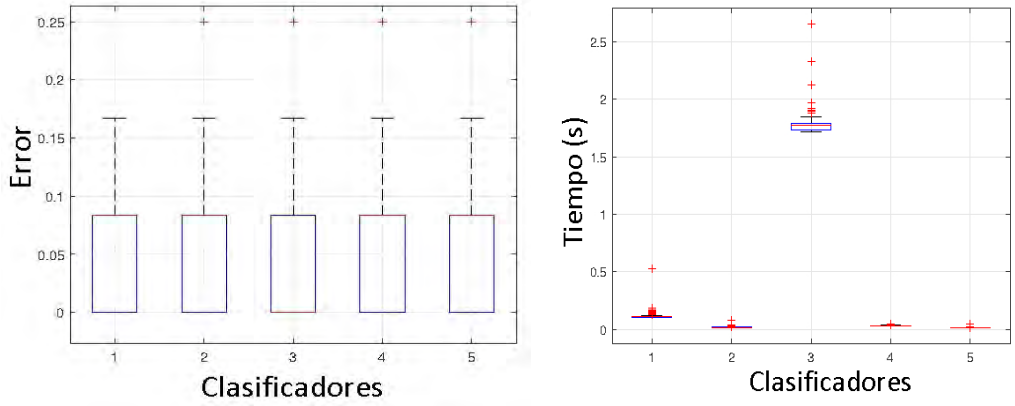
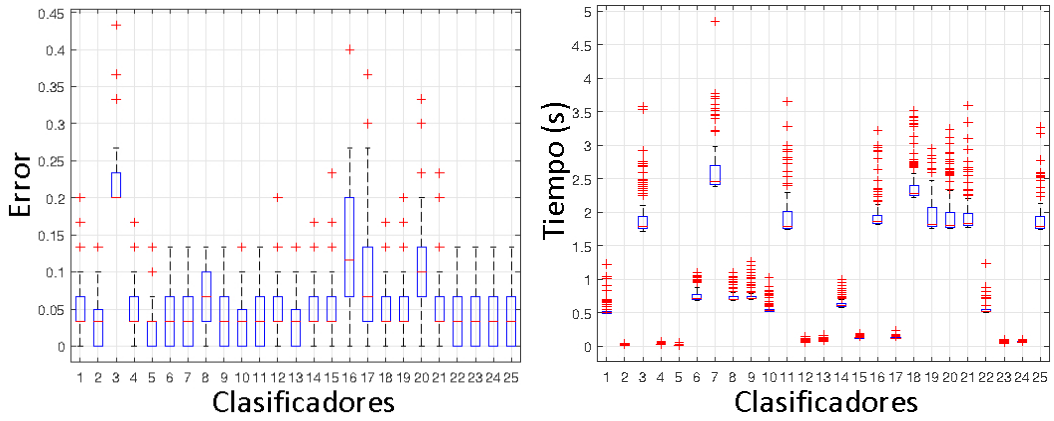


Figura 8. Cascada doble, error por clasificador y tiempo, Hipotiroidismo.



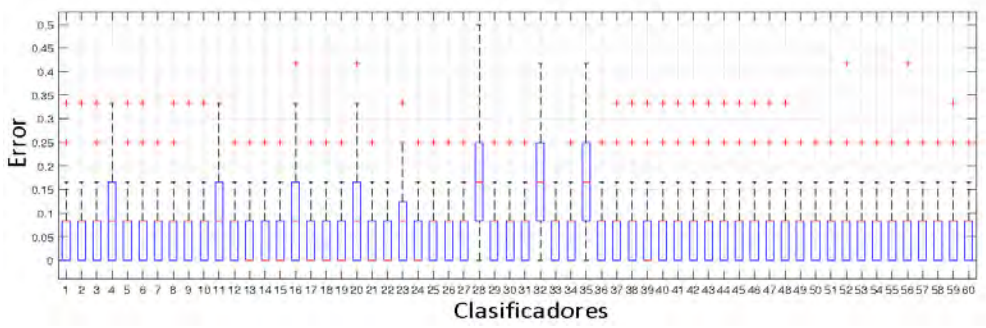
- Experimento 2: Cascada doble

Figura 9. Cascada doble, error por clasificador y tiempo, Dermatología.



- Experimento 3: Cascada triple

Figura 10. Cascada triple, error por clasificador, Hipotiroidismo.



Anexo B. Combinaciones de clasificadores en cascada. Los clasificadores usados son: SVM (Kernel lineal), Random Forest (100 Arboles), KNN (1 Vecino más

cercano), Parzen y Naive Bayes. En la combinación de 2 clasificadores los 5 primeros son los clasificadores individuales, que sirven para tomar de referencia para las combinaciones.

Tabla 13. Combinaciones para 2 y 3 clasificadores.

Combinaciones de 2 clasificadores	**Combinaciones de 3 clasificadores**
1. SVM 2. Parzen 3. Random Forest 4. KNN 5. Naïve Bayes 6. SVM - Parzen 7. SVM - Random Forest 8. SVM - Naïve Bayes 9. SVM - KNN 10. Parzen - SVM 11. Parzen - Random Forest 12. Parzen - Naïve Bayes 13. Parzen - KNN 14. KNN - SVM 15. KNN - Parzen 16. KNN - Random Forest 17. KNN - Naïve Bayes 18. Random Forest - SVM 19. Random Forest - Parzen 20. Random Forest - Naïve Bayes 21. Random Forest - KNN 22. Naïve Bayes - SVM 23. Naïve Bayes - Parzen 24. Naïve Bayes - KNN 25. Naïve Bayes – Random Forest **Observación:** En las cascadas dobles con separación de clase se eliminan los clasificadores del 1 al 5, y el conteo comienza desde el 6.	1. SVM - Parzen - Random Forest 2. SVM - Random Forest - Parzen 3. SVM - Parzen - Naive Bayes 4. SVM - Naive Bayes - Parzen 5. SVM - Parzen - KNN 6. SVM - KNN - Parzen 7. SVM - Random Forest - Naive Bayes 8. SVM - Naive Bayes - Random Forest 9. SVM - Random Forest - KNN 10. SVM - KNN - Random Forest 11. SVM - Naive Bayes - KNN 12. SVM - KNN - Naive Bayes 13. Random Forest - Parzen - SVM 14. Random Forest - SVM - Parzen 15. Random Forest - Parzen - Naive Bayes 16. Random Forest - Naive Bayes - Parzen 17. Random Forest - Parzen - KNN 18. Random Forest - KNN - Parzen 19. Random Forest - SVM - Naive Bayes 20. Random Forest - Naive Bayes - SVM 21. Random Forest - SVM - KNN 22. Random Forest - KNN - SVM 23. Random Forest - Naive Bayes - KNN 24. Random Forest - KNN - Naive Bayes 25. KNN - Parzen - Random Forest 26. KNN - Random Forest - Parzen 27. KNN - Parzen - Naive Bayes 28. KNN - Naive Bayes - Parzen 29. KNN - Parzen - SVM 30. KNN - SVM - Parzen 31. KNN - Random Forest - Naive Bayes 32. KNN - Naive Bayes - Random Forest 33. KNN - Random Forest - SVM

	34.KNN - SVM - Random Forest 35.KNN - Naive Bayes - SVM 36.KNN - SVM - Naive Bayes 37.Naive Bayes - Parzen - Random Forest 38.Naive Bayes - Random Forest - Parzen 39.Naive Bayes - Parzen - SVM 40.Naive Bayes - SVM - Parzen 41.Naive Bayes - Parzen - KNN 42.Naive Bayes - KNN - Parzen 43.Naive Bayes - Random Forest - SVM 44.Naive Bayes - SVM - Random Forest 45.Naive Bayes - Random Forest - KNN 46.Naive Bayes - KNN - Random Forest 47.Naive Bayes - SVM - KNN 48.Naive Bayes - KNN - SVM 49.Parzen - SVM - Random Forest 50.Parzen - Random Forest - SVM 51.Parzen - SVM - Naive Bayes 52.Parzen - Naive Bayes - SVM 53.Parzen - SVM - KNN 54.Parzen - KNN - SVM 55.Parzen - Random Forest - Naive Bayes 56.Parzen - Naive Bayes - Random Forest 57.Parzen - Random Forest - KNN 58.Parzen - KNN - Random Forest 59.Parzen - Naive Bayes - KNN 60.Parzen - KNN - Naive Bayes

Anexo C. Pseudocódigo del algoritmo SMOTE

Algoritmo SMOTE (*T, N, k*)

Entradas: Número de muestras de la clase minoritaria *T*; Cantidad de SMOTE *N%*; Número de vecinos más cercanos *k*
Salida: (*N / 100*)**T* Muestras sintéticas de la clase
minoritaria 1. **si** *N* < 100
2. **entonces** Aleatorizar las muestras de clase minoritaria T
3. *T = (N / 100) * T*
4. *N = 100*
5. **termina si**
6. *N = (int)(N / 100)(* La cantidad de SMOTE es asumida que está en múltiplos enteros de 100 *)*
7. *k* = Numero de vecinos cercanos
8. *numattrs* = Número de atributos
9. *Sample* [ ] [ ]: Arreglo para muestras minoritarias originales
10. *newindex:* Guarda un recuento del número de muestras sintéticas generadas, inicializado a 0
11. *Synthetic* [ ] [ ]: Arreglo para las muestras sintéticas *(* Calcula k vecinos más cercanos para cada muestra de la clase minoritaria solamente *)*
12. **desde** *i* ← *1* **hasta** *T* **hacer**
13. Calcula los *k* vecinos mas cercanos para cada muestra de la clase minoritaria solamente
14. Populate(*N, i, nnarray*)
15. **termina desde**

Populate(*N, i, nnarray) (* Función para generar las muestras sintéticas *)*

16. **mientras** *N ≠ 0*
17. Escoge un número aleatorio entre *1* y *k*, llamado *nn.* Este paso escoge uno de los *k* vecinos más cercanos de *i*
18. **desde** attr ← *1* **hasta** *numattrs* **hacer**

Calcular: *dif* = *Sample[nnarray[nn]][attr]* –
Sample[i][attr] Calcular: *gap* = número aleatorio entre
0 y 1 *Synthetic[newindex][attr]* = *Sample[i][attr]* +
*gap*dif*
termina desde
newindex+
termina mientras
devuelve (* *Final de Populate*
*) Final de Pseudocódigo

Anexo D. Pseudocódigo del algoritmo KNN-U

Algoritmo KNN-U

Entradas: Cantidad de muestras sintéticas a eliminar N, Datos, Clases, cantidad de vecinos cercanos a tomar M.
Salidas: base de datos modificada FBase.

1. **para** k= **hasta** cantidad de casos en la mayoritaria
2. **para** i=1 h**asta** cantidad de datos de la base
3. Xnew1:
4. Se obtienen las distancias de xnew1 con los demás casos
5. `Dist=(norm(xnew1(i,:)-(Datos(k,:))))^2;`
6. se guardan las distancias encontradas
7. **Fin para**
8. Se encuentran M casos cercanos a la mayoritaria y se eliminan N casos
9. **Fin para**
10. Fbase[]:se guardan la base de datos con N datos eliminados

Anexo E. Pseudocódigo del algoritmo KNN

K- Vecinos más cercanos, Clasificar (***X, Y***, *x, k*)

Entradas: Conjunto de entrenamiento *X*; Etiquetas de *X;* Muestra a evaluarle los vecinos cercanos *x*; Número de vecinos cercanos *k*

Salida: Etiquetas de *k* Vecinos más cercanos de *x*

1. **desde** *i = 1* **hasta** *m* **hacer**
2. Calcula la distancia $d_i = d(\boldsymbol{X}_i, x)$
3. **termina desde**
4. Calcula el conjunto I conteniendo los índices para las *k* distancias más cortas $d(\boldsymbol{X}_i, x)$
5. Ordenar $d_i (i = 1,\ldots,N)$ en orden ascendente
6. Quedarnos con los □ casos más cercanos a □
7. **devuelve** etiqueta de$\{\boldsymbol{Y}_i\}$ $donde\ i \in I$

Final de Pseudocódigo

Anexo F. Pseudocódigo del sistema CBR

Case Base Reasoning CBR

Entradas: Conjunto de entranamiento (Base de Casos) X_BC; Etiquetas de X_BC, Y_BC; Nuevo Caso NC

Salidas: Casos Similares X_cer ; Probabilidades de pertenencia a cada clase $\Pr\_NC$, Clase del nuevo caso $Classf$; Error del clasificador $error$.

1. Crear conjunto de datos
   A = dataset(X_BC,Y_BC)
2. Seleccionar conjunto de entrenamiento y validación
   C = Entrenamiento
   D = Validación
3. Entrenamiento del clasificador
4. Estimación de

error Nuevo Caso

5. **mientras** flag ==1
6. Recuperar casos similares y ordenarlos en forma ascendente $X_{ce} = sort(dist(NC, X_{BC}))$
7. Identificar la clase de NC que fue asignada por el clasificador
8. Estimación de

probabilidades 9. $X_P = [X_{BC}; NC]$

10. $Y_P = [Y_{BC}; (n)]$
11. $X_{ds} = d(X_P, Y_P)$
12. $Prob = Parzen\ (X_{ds})$

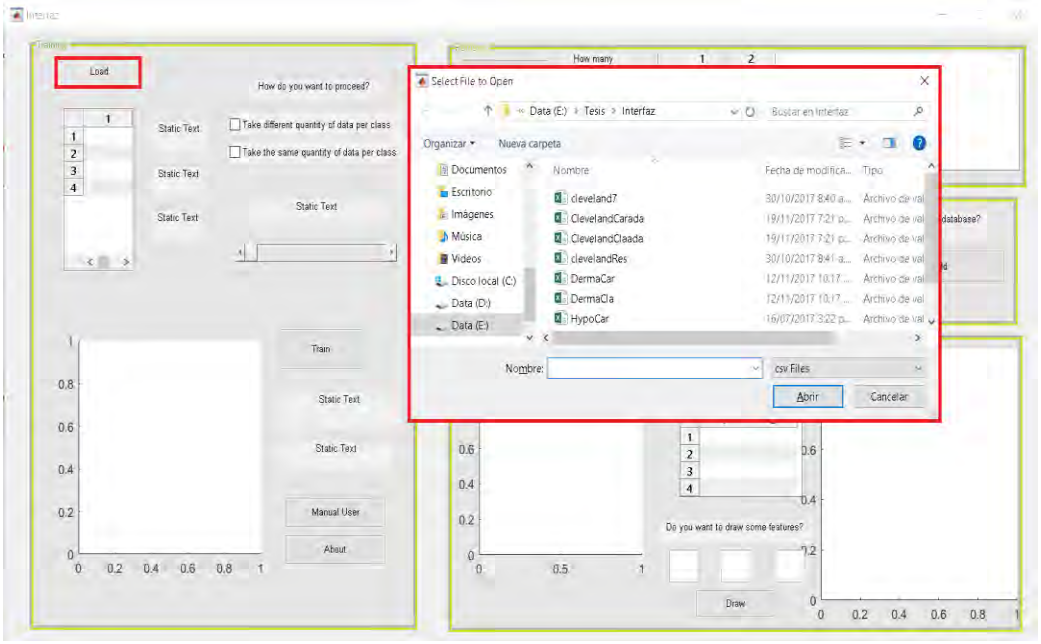
13. Clase asignada
14. Mostrar $Classf$
15. Retener información?
    0 Añadir a base de casos
    1 No añadir
    2 Enviar a cuarentena
16. Agregar nuevo caso?
    0 Salir → flag = 0
    1 Continuar → flag = 1 → Limpiar variables
17. **termina mientras**
    Final de Pseudocódigo
4.

Anexo G. Manual de usuario interfaz CBR

- Subir la base de datos

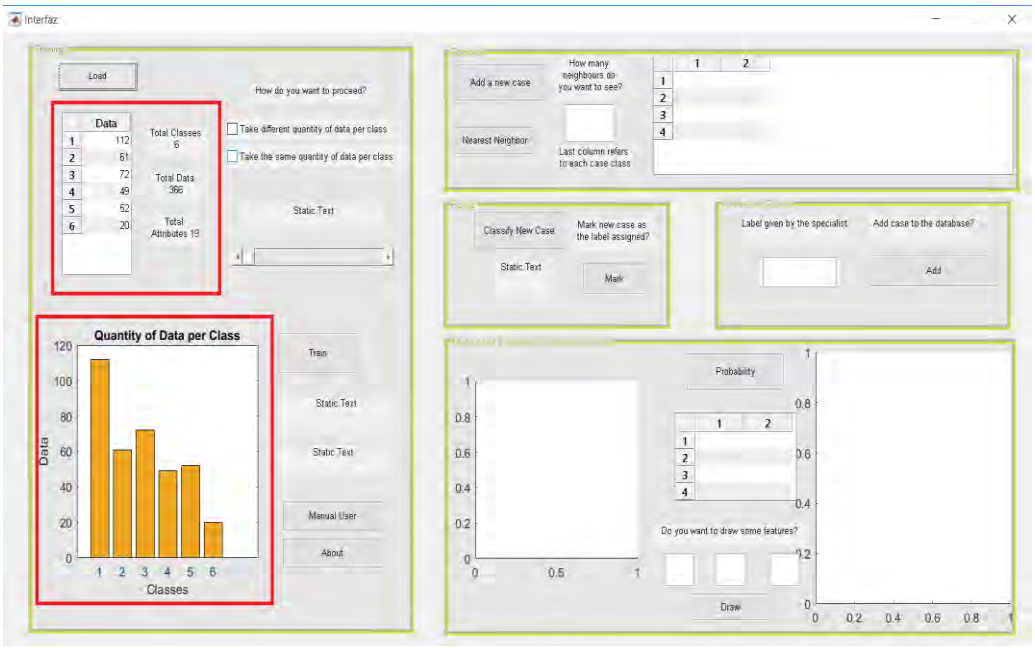
Se carga la base de datos encargada de entrenar el sistema CBR por medio del botón Load, el cual se encarga de preguntar en una ventana adjunta, por los archivos que contengan las características y clases respectivas a la base de datos separadas en formato Excel.

Figura 11. Cargado de la base de datos a la interfaz.



Una vez cargada la información por medio de la interfaz se podrá observar la cantidad de características, clases y datos en total listos para ser usados.

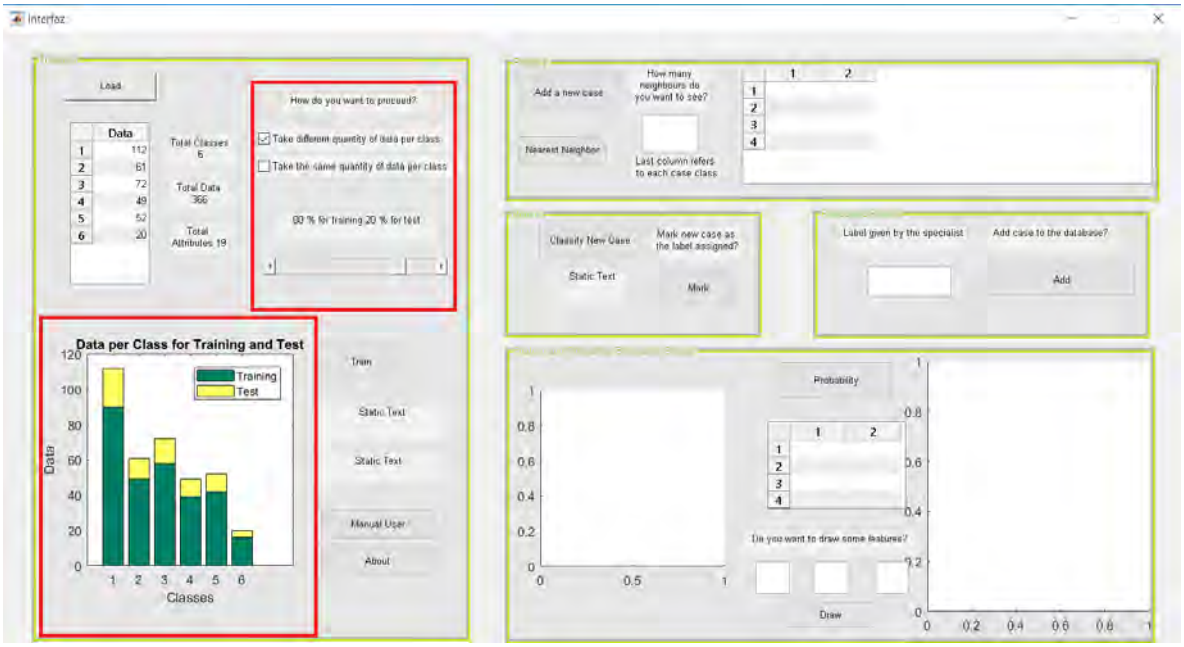
Figura 12. Visualización de la base de datos cargada.



- Selección de datos

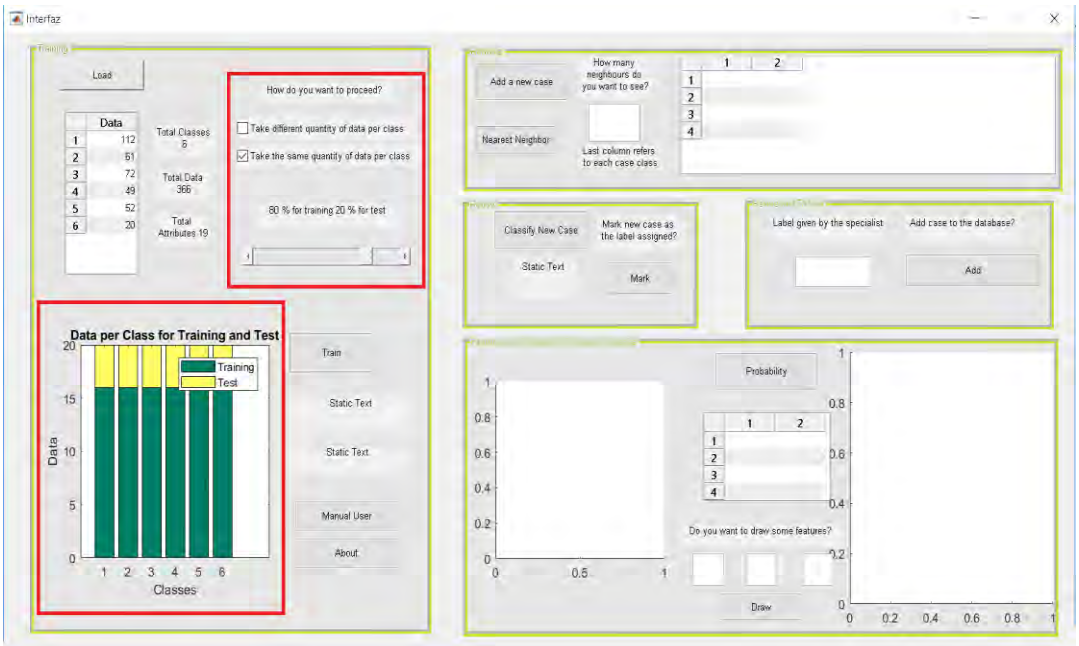
    1. la cantidad de datos con la cual se entrena y prueba los clasificadores Bi-clase y Multi-clase tomando la mínima cantidad posible, o la cantidad original para cada clase en la base de datos, para que todas las clases presenten una misma cantidad de datos, estará presentes en dos iconos para marcar cualquiera de estas dos opciones

Figura 13. Decisión de porcentaje para entrenamiento y prueba del sistema.



2. El porcentaje en el cual se dividirá la cantidad de datos tomada para entrenamiento y para prueba de os clasificadores.

Figura 14. Visualización de porcentaje para entrenamiento y prueba del sistema.



- Entrenamiento

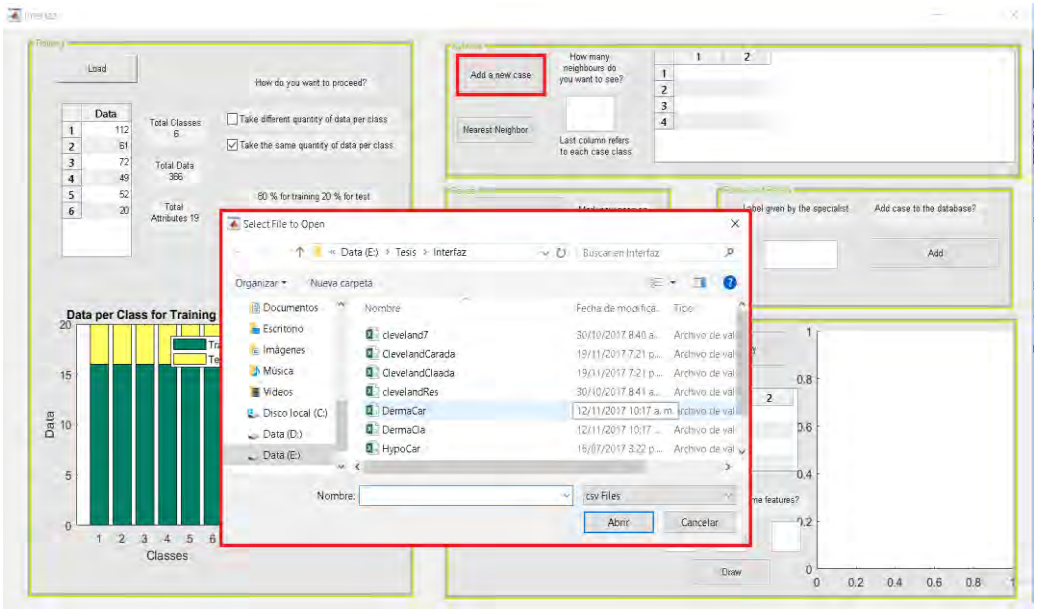
Haciendo uso del botón Train se procede a entrenar el clasificador Bi-clase para luego probar el clasificador Multi-clase eliminando una clase de la base de datos en total, los resultados se observan en las dos cajas de texto debajo de botón.

Figura 15. Entrenamiento del sistema.

- Añadir nuevo caso

Por medio del botón “Add a new case”, al desplegar una nueva ventana, se obtiene, en formato Excel, a obtener el nuevo caso.

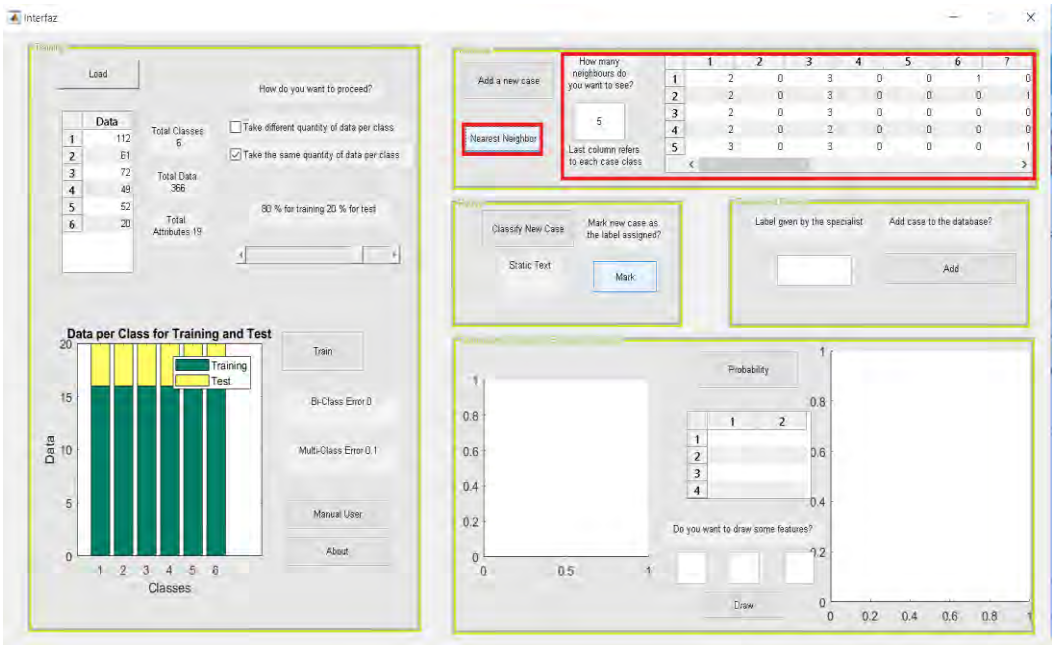
Figura 16. Adición de un nuevo caso.



- Recuperar

Al presionar el botón “Nearest Neighbor” se obtienen los casos pertenecientes a la base de casos más cercanos al nuevo caso el número de casos debe ser introducido por el usuario en el cuadro en blanco.

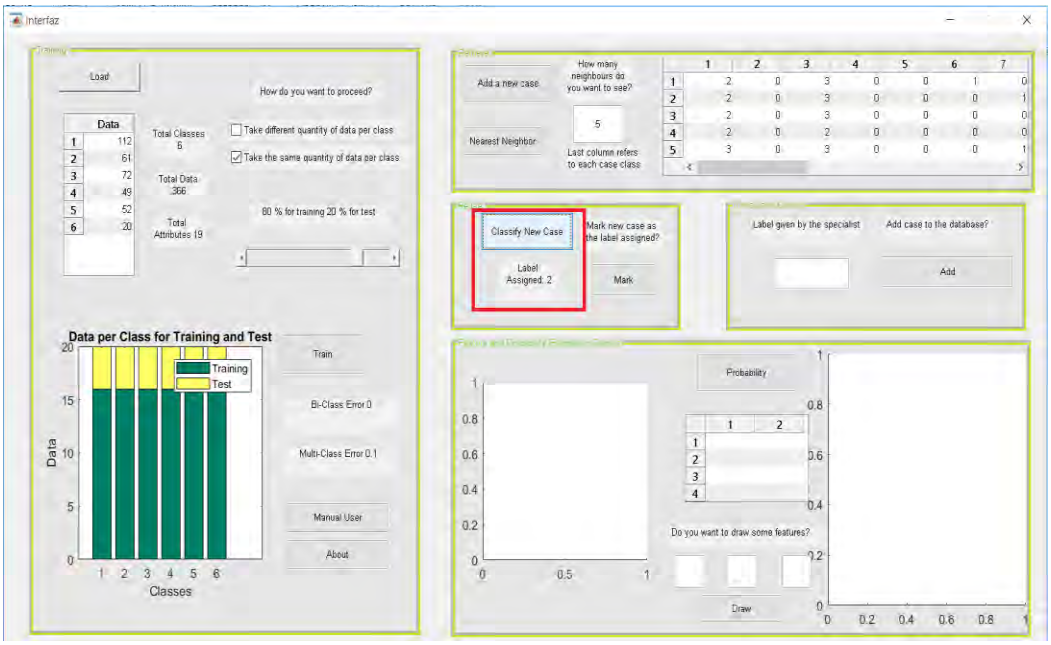
Figura 17. Visualización vecinos cercanos al nuevo caso



- Adaptación

Usando los clasificadores entrenados a presionar el botón “Classify New Case” se obtiene la clase generada por los dos clasificadores.

Figura 18. Etiqueta entregada por el sistema al nuevo caso.



Si el usuario se encuentra acorde con la etiqueta entregada por el sistema puede guardar el nuevo caso con dicha etiqueta presionando el botón Mark sin necesidad de escribirla en la siguiente etapa, el cuadro de texto quedara inhabilitado si se presiona el botón.

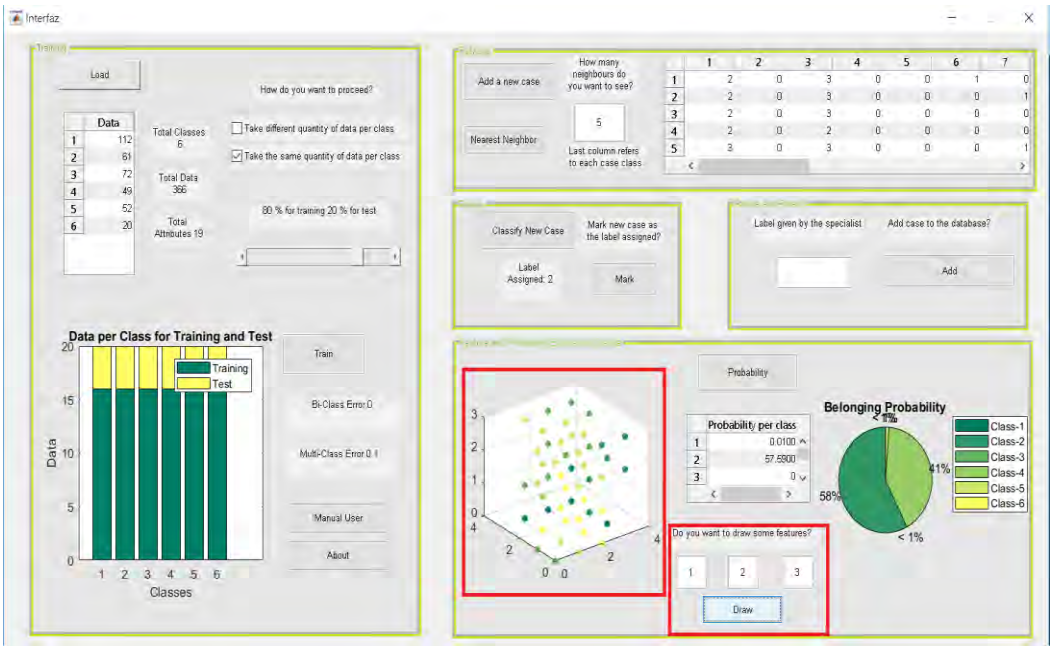
Figura 19. Botón para etiquetar el nuevo caso con la clase dada por el sistema.

- Estimación de probabilidad

Si se desea se puede observar, presionando el botón “Probability”, la probabilidad de que el nuevo caso pertenezca a una clase en particular.

Figura 20. Estimación de la probabilidad de pertenencia a cada clase

Figura 21. Graficas de las características introducidas por el usuario.



- Revisión y aprendizaje

Una vez la información ha sido analizada por el usuario, si lo desea, puede introducir el nuevo caso a la base para entrenamiento, con la clase que se concluyó describe claramente el nuevo caso presentado y la cual ha sido colocada dentro del cuadro en blanco.

Figura 22. Adición del nuevo caso con su respectiva etiqueta a la base de datos usada para entrenar al sistema.

Anexo H. Certificado de presentación congreso IWBBIO 2018

Figura 23. Certificado de presentación a la conferencia IWBBIO 2018

Anexo J. Articulo para congreso IWBBIO 2018

Case-Based Reasoning Systems for Medical Applications with Improved Adaptation and Recovery Stages

X. Blanco Valencia[1], D. Bastidas Torres[2(✉)], C. Piñeros Rodriguez[2], D. H. Peluffo-Ordóñez[2,3], M. A. Becerra[4], and A. E. Castro-Ospina[4]

[1] Universidad de Salamanca, Salamanca, Spain
[2] Universidad de Nariño, Pasto, Colombia
daba89@live.com
[3] Universidad Yachay Tech, Urcuquí, Ecuador
[4] Instituto Tecnológico Metropolitano, Medellín, Colombia

Abstract. Case-Based Reasoning Systems (CBR) are in constant evolution, as a result, this article proposes improving the retrieve and adaption stages through a different approach. A series of experiments were made, divided in three sections: a proper pre-processing technique, a cascade classification, and a probability estimation procedure. Every stage offers an improvement, a better data representation, a more efficient classification, and a more precise probability estimation provided by a Support Vector Machine (SVM) estimator regarding more common approaches. Concluding, more complex techniques for classification and probability estimation are possible, improving CBR systems performance due to lower classification error in general cases.

Keywords: Case-based reasoning · Preprocessing
Cascade classification · Probability

1 Introduction

Reasoning in humans is based on the process of remembering and applying rules, the product of various experiences that generate knowledge [1]. Case-based reasoning (CBR) is a problem solving approach that uses past experience to tackle current problems. Technically, CBR is a methodology that has proven to be appropriate for applying analogy strategies in unstructured domains and where knowledge acquisition is difficult [2]. Therefore, it is the ideal methodology for the development of support systems for medical diagnosis [3]. Through previous analysis, it can provide results that allow a better understanding of a patient and, therefore, a better diagnosis and a better diagnosis and treatment [4]. The life cycle of a CBR-based system consists of four main phases: to identify the

X. Blanco Valencia—This work is supported by Faculty of Engineering from University of Salamanca.

I. Rojas and F. Ortuño (Eds.): IWBBIO 2018, LNBI 10813, pp. 26–38, 2018.
https://doi.org/10.1007/978-3-319-78723-7_3

current problem and find a past case similar to the new case (retrieve), using the case and suggest a solution to the current problem (reuse/adaptation), evaluate the proposed solution (revise), and update the system to learn from experience (retain) [5]. In this paper, we propose an improvement to case-based reasoning systems by developing an estimate of the relation between new cases with existing classes and using multi-class cascade classifiers giving a better diagnostic assistance in medical settings.

This paper is organized as follows: Sect. 2 reviews some related works and outlines basics on CBR. Section 3 describes the operation of the proposed classification approach. Section 4 gathers some results and discussion. Finally, conclusions and final remarks are presented in, Sect. 5.

2 Related Works and Background

The origin of the CBR can be traced to Yale University and the Schank and Abelson's work in 1977 [6]. An early exploration of CBR in the medical field was conducted by Koton [7] and Bareiss [8] in the 1980s. The CBR is inspired by human reason, i.e., to solve a problem by applying previous experiences adapted to the current situation. A case (episodic experience) contains a problem, a solution, and its result.

Clinical practice can begin with some initial experiences (cases resolved), then, those experiences are used to solve a new problem where it can been involve some adjustment in the previous solutions to solve the new problem. Therefore, the CBR is a reasoning process which is medically accepted and seems to call attention increasingly [4,9,10]. Anderson has demonstrated how people use past cases as models for learning to solve problems, particularly in early learning. Other results like Kolodner, indicate that experts who know a lot about a particular subject, can remember events in their domain of expertise more easily than non-experts [11].

In the literature, there is a variety of works that apply the CBR methodology focused on the health sector [12–14], and in several of them, its evolution is studied through the last years. For example, Bichindaritz [15], refers to the CBR as an appropriate methodology for the care of the elderly and support for people with disabilities and in [16], it is concluded that automatic adaptation is a weakness especially in systems based on CBR in the medical field. It is [17] suggested working a reduction of dimensions along with CBR, in order to improve the increasingly large, complex and uncertain data systems of clinical environments. Often, with the increase in the number of classes, the complexity and the computational cost increase. In addition, difficulties in the classification may be present only for some classes [18,19].

The present article shows an alternative solution for the problems existing in the automatic adaptation stage.

3 Proposed Classification Approach

This proposal is aimed at improving the retrieve and adaptation stages of case-based reasoning systems through two processes. The first is focused on an appropriate preprocessing in order to improve the representation of the base of cases and to obtain better results in the classification; and a second process, where the recovery and adaptation stages are combined using cascade pattern recognition algorithms, which improves the result of the classification. Finally, it is proposed to estimate the probability density using support vector machines. This results in a CBR with a greater amount of resources that will give the expert enough support to make the best decision in a medical environment.

3.1 Oversampling and Undersampling Methodology for Class Balancing

Class imbalance problems can be addressed in different ways, being the most common one the oversampling technique, this technique consists in increasing the size of the class with the least quantity of cases, also called minority class, through adding synthetic samples and obtaining a number of records similar to the majority classes. These techniques are employed to avoid the over-training problems caused by the big difference between the sample amount per class. The increase of data in the minority class results in a better classification in exchange for a higher computational cost. The undersampling technique proposes to reduce the majority class to an equal or lower size compared to the minority class, this can be done in several ways, such as eliminating redundant samples, eliminating very close data to random samples by finding the nearest neighbor, deleting random, among others. These methods, as opposed to oversampling, remove unnecessary data, giving a lower computational cost, but this process can also remove relevant data, thus affecting the classification process.

In the present work, tests were performed with different balancing methods, such as, under and over sampling or a mixture of both. The classification error and the computational cost were the measurement and comparison patterns, respectively.

3.2 Cascade Classification Methodology

The systems based on CBR contemplate independent stages for the recovery and adaptation phase, in this work these stages were integrated into a single one, resulting in computational savings. The majority of CBR systems are built using the KNN algorithm as part of the retrieve stage, and usually the adaptation stage is avoided due to its complexity. This article proposes a complex technique based on sequential classification with classifiers of different types, entering a wide research field.

3.3 SVM Probability Density Methodology

As a complement and improvement of the adaptation stage, the class membership of a new case is predicted. The estimation of probability density by Parzen windows is one of the most studied and well-documented methods at the moment. Support Vector Machine has shown its capacity in application and pattern recognition in general. The method consists in drawing each case from an N dimension, where N is the number of features, then calculating a hyperplane separating each class. This proposal uses SVM as an alternative to other methods such as Parzen and KNN.

4 Results and Discussion

4.1 Database

All databases were obtained from the machine learning repository UCI [20]. Two databases with multiple medical diagnoses in the public domain are considered. Hypothyroidism with 5 features distributed around 3 classes and dermatology with 19 features distributed around 6 classes. With the feature selector technique, the hypothyroidism database was reduced from 29 to 5 features, and dermatology database with 33 features was reduced to 19 features making the classification procedure more optimal.

4.2 Methods

The average classification error was used as a measure of comparison between the different experiments.

1. **Preprocessing:** With the purpose of selecting the most relevant feature, techniques like CFS (Correlation Feature Selection)-Best first and InfoGain AttributeRanker were used, found in the data mining software WEKA. Next undersampling and oversampling procedures for class balance are applied, 6 experiments were made in order to identify this procedures using SMOTE, KNN-Undersampling, ADASYN and a combination of both SMOTE-KNNU algorithms.
2. **Cascade classifiers algorithms:** Using a different programming software than the data mining software WEKA, 5 classifiers were used (Naive Bayes, Parzen, Random Forest, KNN and SVM). Experiment 1 has 2 and 3 classifiers combination embedded sequentially without repeating the same one using 5 classifiers, where the classifiers are trained with the output of the classifier before. For experiment 2 a class is separated from the original database using a bi-class classifier. As a result a separate sequential classification form is used, being the first part a bi-class classification and the second part an embedded sequential 2 and 3 classifiers combination, with the same combination as experiment 1. This experiments are run 100 times for repeatability and reproducibility purposes, also every experiment were made with 70% from the database for training and 30% for test.

3. **Probability estimation:** Parzen Windows, KNN and SVM were used as probability estimator. 70% of the database was used for training and a 30% to obtain the success rate between the estimator output and the real class given by the database. Also execution time were measured between estimators.

4.3 Performance Measures

The preprocessing and cascade classification were compared using execution time and classification error, for probability estimation, similarity between the class estimation and the original class expressed as an percentage and execution time were used.

4.4 Experiments

1. **Pre-processing**
   (a) **Feature Selection Methods**
      i. **Experiment 1:** The execution time is high using multilayer perceptron classifier, being this 24.72 s for dermatology database and 22.68 s for hypothyroidism, every other classifier showed execution times lower than 1 s for every database. The highest classification errors can be observed using Naive Bayes in the dermatology database with 97.81% and random forest on hypothyroidism database with 99.31%.
      ii. **Experiment 2:** Feature selection technique cfseval-Best first is implemented, the same classifiers of experiment 1 were used. Classification errors can be observed in the following tables. Table 1 for dermatology and Table 2 for hypothyridism.
      iii. **Experiment 3:** The results using InfoGainAttributeval- Ranker can be seen in Tables 1 and 2.

   On Tables 1 and 2 all classifiers show a good classification process, success rate percentage is above 90% for both databases. For dermatology database (Table 1) with best first selector shows Naive Bayes as the best classifier with 97.81% success rate, as for hypothyroidism database (Table 2) with ranker selector, random forest is the best classifier with

Table 1. Dermatology database with best first and ranker as selectors

Classifier	Favorable classification %		Poor classification %		Time (s)	
	Best first	Ranker	Best first	Ranker	Best first	Ranker
NaiveBayes	**97.81**	97.26	2.18	2.73	0	0
Multilayer perceptron	96.44	95.90	3.55	4.09	**9.38**	**9.66**
KNN (1)	96.44	95.35	3.55	4.64	0	0
SVM (Linear Kernel)	97.26	97.26	2.73	2.73	0.06	0.06
Random forest	96.44	95.62	3.55	4.37	0.05	0.03

Table 2. Hypothyroidism database with best first and ranker as selectors

Classifier	Favorable Classification %		Poor Classification %		Time (s)	
	Best first	Ranker	Best first	Ranker	Best first	Ranker
NaiveBayes	94.64	94.72	5.35	5.27	0	0
Multilayer perceptron	96.10	96.26	3.89	3.73	**2.94**	**3.02**
KNN (1)	93.16	94.22	6.83	5.7794	0	0
SVM (Linear Kernel)	93.13	93.50	6.86	6.49	0.2	0.13
Random forest	95.78	**97.50**	4.21	2.49	0.45	0.45

97.50% success rate. Time wise there is a notorious decrease on the majority, for example Random Forest classifier for dermatology database showed an execution time of 24.60 s without selection method, by contrast, Table 1 execution time with best first selector was 9.38 s and 9.66 s for ranker selector. According to Table 2, Multilayer perceptron classifier execution time is reduced from 22.68 s Table 1 to 2.94 s and 3.02 s for best-first and ranker selectors, respectively. These results reaffirm that feature selection technique is a good option for data optimization, eliminating useless feature and longer execution times for different systems.

(b) **Balancing Methods**

i. **Experiment 1:** Using the databases with the most relevant features, the classifier based on KNN was applied, without a balancing method.

ii. **Experiment 2:** Focusing on the minimum amount of cases by class, the number of cases of the majority class is reduced with respect to the minority class, avoiding over classification and verifying if the data of the major class is indeed necessary for a good classification. Dermatology database presents a minimal amount of 20 cases of class 6, and a minimal amount of 50 cases of class 3 for Hypothyroidism Database.

iii. **Experiment 3:** Now applying a general preprocess in the database known as oversampling the most used technique is SMOTE, this algorithm makes new synthetic data between the original ones, the amount of data generated were programmed by 50 intervals to know if really is necessary an increase in the data overall being the amount of the major class the limit.

iv. **Experiment 4:** An undersampling method with KNN technique was applied by deleting unnecessary data without compromising the performance of the classification stage. Hypothyroidism database majority class with 1790 cases is reduced to 64, this value obtained by modifying nearest neighbor amount and distance trying to eliminate unnecessary data. Dermatology database class 1 with 112 cases was reduced to 20 cases.

v. **Experiment 5:** Here, it is used the hybrid balancing method SMOTE-KNNU, which used the same parameters that the

experiment 3 and 4. After this process dermatology database ends up with 72 cases for each class and hypothyroidism with 100.

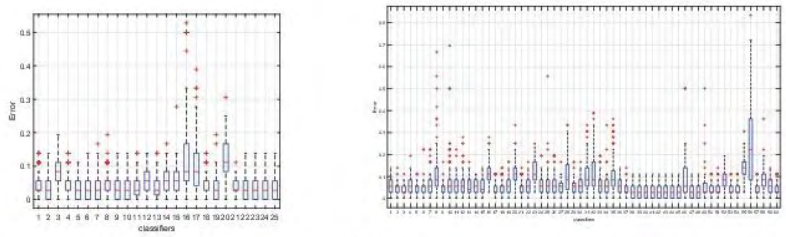
vi. **Experiment 6:** This experiment used as balancing method Adasyn (adaptive synthetic sampling), a extension of SMOTE, not only creates synthetic data on one point but in many around a center. Parameters as percentage of increasing data is managed. Hypothyroidism was the only database in which this algorithm was implemented due to the neighboring of data per each class. Results summary are shown on Table 3, where classification error is analyzed to choose the best option for both databases.

Table 3. Performance results in terms of error percentage % of wrong classifications.

Database	Experiment 1	Experiment 2	Experiment 3	Experiment 4	Experiment 5	Experiment 6
Hypothyroidism	5.39 ± 5.39	$\mathbf{25.26 \pm 8.78}$	9.20 ± 9.36	14.13 ± 5.96	$\mathbf{2.37 \pm 5.53}$	15.32 ± 9.91
Dermatology	$\mathbf{2.90 \pm 8.68}$	$\mathbf{4.81 \pm 8.82}$	4.57 ± 8.77	3.28 ± 9.87	$\mathbf{2.69 \pm 12.23}$	

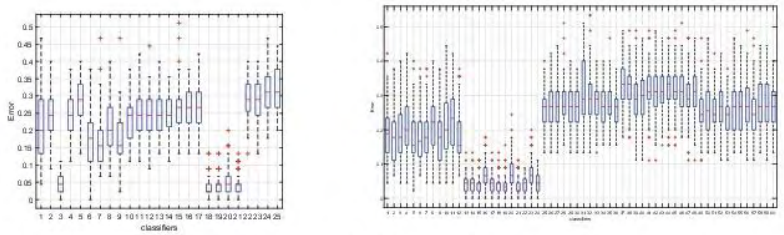
As regards on Table 3, on hypothyroidism database, the minimum error value was in experiment 5, where classification error percentage was 2.37%. Also classification errors are uneven, for example, experiment 2 got 25.26%, and experiment 4 got 14.14%. Experiment 1 shows a lower classification error, possibly result of an overtraining process, given the excessive amount of data for class 1 training the classifier almost exclusively. For dermatology database, there is a similar behavior in all 5 experiments, the difference between the best classification on experiment 2 with 4.81% and experiment 5 with 2.69% is not bigger than 3%. Another important remark resides on experiment 1 with 2.90% as the lowest classification error for all other tests. Also dermatology database does not contain big differences in the amount of data per classes, avoiding overtraining, making it a reasonable database to be used without a pre-processing technique.

2. **Cascades:** The following classifiers were used in the experiments: Parzen, SVM, Random Forest, Naive bayes and KNN.
   (a) **Experiment 1:** Two and three classifiers sequence were used, with 21 and 60 combinations respectively. Hypothyroidism database with a KNNU-SMOTE pre-processing and dermatology database without one are used to test every combination on a cascade classifier environment, classification error is shown by boxplots as follows:
   The first 5 boxes for dermatology database are individual classifiers, Fig. 1(a) shows classifier 5 as the best classifier (Naive Bayes) with the lowest error, an average of 0.04 and variance of 0.12 maximum, 6 to 25 combination displays no alteration regarding classifier 5 results. On triple combinations Fig. 1(b) shows no variation on classification error; the average was 0.04, combination number 3 shows a lower variance, although classifier 5 shows a value of 0.12. Figure 2(a) regarding hypothyroidism,



(a) *Double Classifiers Combination Error in Dermatology Database* (b) *Triple Classifiers Combination Error in Dermatology Database*

Fig. 1. Dermatology database performance



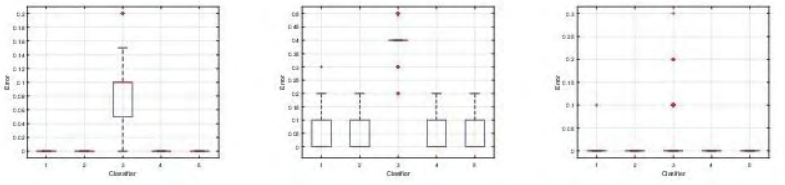
(a) *Double Classifiers Combination Error in Hypothyroidism Database* (b) *Triple Classifiers Combination Error in Dermatology Database*

Fig. 2. Hypothyroidism database performance

random Forest (classifier 3) proves to be the classifier with the lowest classification error with an average of 0.04 and a variance of 0.12, double combinations 18 to 21 shows low classification error, but there is no variety between combinations. Figure 2(b) presents a similar case, combination 13 to 24 shows an average error of 0.05 and other combinations offers way higher average error and variance. Sequentially implemented classifiers do not affect classification error significantly. So other methods were implemented, trying to reduce classification error the lowest possible.

(b) **Experiment 2:** A class was removed from the original database, so 5 binary classifiers were used for this purpose (SVM, Parzen, Random Forest, KNN and Naive Bayes), naming the desired class for removal 'class 1' and all the other ones as 'class 0', this way a second classifier, in this case a multiclass one, was trained for the remaining classes making the classification process a little bit easier. Classification error and execution time are shown on Figs. 3 and 4.

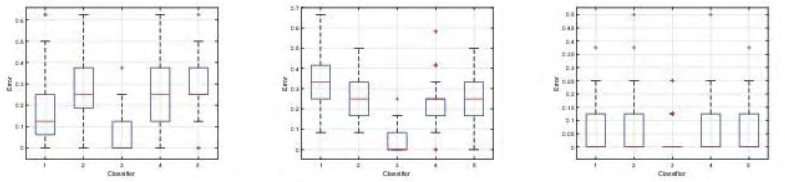
Figure 3(c), the best binary classifier for dermatology database was classifier 2 (Parzen) and the best class for removal was class 3, although classifier 1 and 2 presents the same classification error, variance on both cases are bigger than classifier 2 like 0.3. Similarly, Fig. 4(c) shows that



(a) *Class 1 Error per Classifier* (b) *Class 2 Error per Classifier* (c) *Class 3 Error per Classifier*

(d) *Class 4 Error per Classifier* (e) *Class 5 Error per Classifier* (f) *Class 6 Error per Classifier*

Fig. 3. Binary classifier error per class with dermatology database

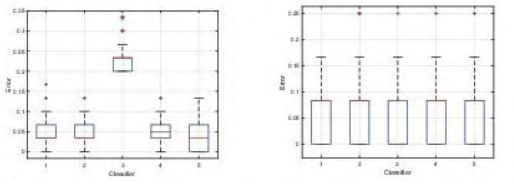


(a) *Class 1 Error per Classifier* (b) *Class 2 Error per Classifier* (c) *Class 3 Error per Classifier*

Fig. 4. Binary classifier error per class with hypothyroidism database

class 3 contains the lowest error and a variance of 0 with the third classifier (Random Forest). Results shown proves that a correct data separation might lead to a better classification, thus, a better adaption improvement.

(c) **Experiment 3:** Once a class and a binary classifier were chosen, tests for a multiclass classifiers were made, using the same classifier as experiment 2. Results are shown on Fig. 5:

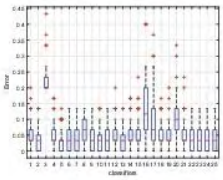


(a) *Single Combination Classifiers Error in Dermatology Database* (b) *Single Combination Classifier Error in Hypothyroidism Database*

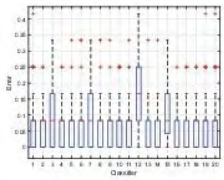
Fig. 5. Single combination multiclass classifier performance with one less class

There is no significant change with experiment 1, average error and variance are similar. On the contrary hypothyroidism database shows an average error of 0 and a variance a little higher with 0.17 not showing a big improvement.

(d) **Experiment 4:** Now a double combination of classifiers is added for a multiclass classifier, using the same combination as experiment 1 but with one less class to classify. Results are shown on Fig. 6.



(a) *Double Combination Classifiers Error in Dermatology Database*

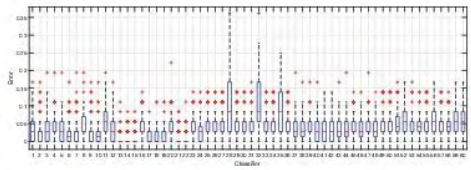


(b) *Double Combination Classifiers Error in Hypothyroidism Database*

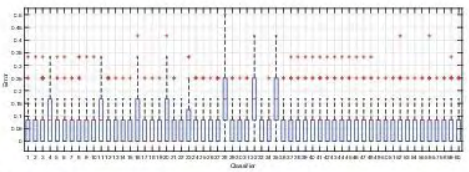
Fig. 6. Double combination multiclass classifier performance with one less class

Hypothyroidism database keeps 0 as an average error on 13, 14, 16 and 17 combination. Dermatology database does not show any improvements regarding experiment 1.

(e) **Experiment 5:** A triple combination of classifiers is analysed, same as experiment 1, 60 combination are implemented trained with one less class database. Results are shown on Fig. 7.

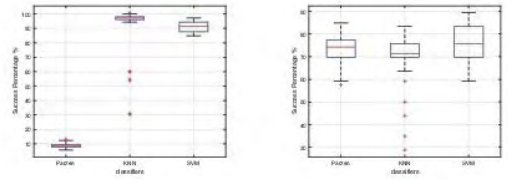


(a) *Triple Combination Classifiers Error in Dermatology Database*



(b) *Triple Combination Classifiers Error in Hypothyroidism Database*

Fig. 7. Triple combination classifiers error



(a) *Probability estimator Success Rate Dermatology Database*

(b) *Probability estimator Success Rate Hypothyroidism Database*

Fig. 8. Probability density estimation

Most relevant result is seen on dermatology database (Fig. 7a), a lower average error was obtained on combination 13, 14, 15, 20 and 21, with a few high error values but lower variance in general. Any of the 5 combination mentioned before could improve the classifying process. On Hypothyroidism database there is no change on any values compared to any experiment made.

3. **Probability**
   (a) **Experiment 1:** The 3 most representative classifiers were used as density probability estimators: Parzen Windows, KNN and SVM.

The results were compared with the original class of each case, counting only the good estimation over the bad ones. For every test made, values like average and minimal classification error and executed time were taken, as their respective graphics, error and time vs times the test was made. Every test was executed a 100 times, analyzing how much the results differ from one cycle to another and the average value.

Probability estimation was introduced. As shown in Fig. 8, for analysis success percentage was measured and graphed on boxplots, Parzen windows estimator shows a good efficiency on hypothyroidism database Fig. 8(a) with an average of 75% success rate and low variance, instead dermatology database Fig. 8(b) has success of 8%, something uncommon for such recognized estimator. Later KNN estimator was taken, another recognized probability estimator, lower classification error was given, hypothyroidism database average has a 72% success rate on dermatology and 95% success rate was found but with a higher variance, for example 30% to 50% success rate was recorded. SVM estimator gave the lowest result and variance than others on both databases on average 75% success rate for hypothyroidism and 93% for dermatology.

5 Conclusion

- Using more complex tools in pattern recognition systems like cascade classification and probability density estimation, proves to improve precision and accuracy in general, showing lower classification error and high success rate

for probability estimation, this composition in a CBR system, focused on medical environments, provide two independent techniques for a better diagnosis and understanding of different problems in this area.
- Cascade classification improves classification error on one classifier. Although a larger amount of classifiers does not mean a better classification process, database nature and behavior are very important in the process of lowering classification error, like in hypothyroidism lowering classification error is a very complex process, although in dermatology there were results as 0 for classification error. This means cascade classification could be more necessary on some system than others making the use of other techniques relevant.
- Probability estimation using SVM as estimator proved to get better results for both databases than other estimators, although execution time was increased a more precise an accurate system is preferred over a faster and lighter one, especially for medical environments.

References

1. Leake, D.B.: CBR in context: the present and future. In: Case-Based Reasoning, Experiences, Lessons and Future Directions, pp. 1–30 (1996)
2. Kolodner, J.L.: Maintaining organization in a dynamic long-term memory. Cogn. Sci. **7**(4), 243–280 (1983)
3. Abecker, A.: Corporate memories for knowledge management in industrial practice: prospects and challenges. J. Univ. Comput. Sci. **3**(8), 929–954 (1997)
4. Aamodt, A., Plaza, E.: Case-based reasoning: foundational issues, methodological variations, and system approaches. AI Commun. **7**(1), 39–59 (1994)
5. Pal, S.K., Shiu, S.C.: Foundations of Soft Case-Based Reasoning, vol. 8. Wiley, Hoboken (2004)
6. Schank, R.C., Abelson, R.P.: Scripts, Plans, Goals and Understanding: An Inquiry into Human Knowledge Structures. Oxford, England (1977)
7. Koton, P.: Using experiences for in learning and problem solving. Engineering and Computer Science mMIT/LCS/TR-441 (1989)
8. Bareiss, R.: Exemplar Based Knowledge Acquisition: A Unified Approach to Concept Representati on, Classification, and Learning. Academic Press Professional Inc., San Diego (1989)
9. Anderson, J.R.: The Architecture of Cognition. Harvard University Press, Cambridge (1983)
10. Kolodner, J.L.: Maintaining organization in a dynamic long-term memory*. Cogn. Sci. **7**(4), 243–280 (1983)
11. Shiu, S.C., Pal, S.K.: Case-based reasoning: concepts, features and soft computing. Appl. Intell. **21**(3), 233–238 (2004)
12. Paz, J.F.D., Bajo, J., Vera, V., Corchado, J.M.: MicroCBR: a case-based reasoning architecture for the classification of microarray data. Appl. Soft Comput. **11**(8), 4496–4507 (2011)
13. Paz, J.F.D., Bajo, J., López, V.F., Corchado, J.M.: Biomedic organizations: an intelligent dynamic architecture for KDD. Inf. Sci. **224**, 49–61 (2013)
14. De Paz, J.F., Rodríguez, S., Bajo, J., Corchado, J.M.: Case-based reasoning as a decision support system for cancer diagnosis: a case study. Int. J. Hybrid Intell. Syst. **6**(2), 97–110 (2009)

15. Bichindaritz, I., Marling, C.: Case-based reasoning in the health sciences: what's next? Artif. Intell. Med. **36**(2), 127–135 (2006)
16. Juárez, J., Campos, M., Gomariz, A., Palma, J., Marin, R.: A reuse-based CBR system evaluation in critical medical scenarios. In: 21st International Conference on Tools with Artificial Intelligence, ICTAI 2009, pp. 261–268, November 2009
17. Montani, S.: How to use contextual knowledge in medical case-based reasoning systems: a survey on very recent trends. Artif. Intell. Med. **51**(2), 125–131 (2011)
18. Krawczyk, B., Woźniak, M., Herrera, F.: On the usefulness of one-class classifier ensembles for decomposition of multi-class problems. Pattern Recogn. **48**(12), 3969–3982 (2015)
19. Kang, S., Cho, S., Kang, P.: Multi-class classification via heterogeneous ensemble of one-class classifiers. Eng. Appl. Artif. Intell. **43**, 35–43 (2015)
20. Lichman, M.: UCI Machine Learning Repository (2013)

Anexo K. Página Web

A continuación, se muestra una página web creada con el fin de compartir el desarrollo, códigos realizados y artículos presentados en la trayectoria de este proyecto, además de tener un sitio fijo en el cual se pueda encontrar la información relacionada en cualquier momento que sea necesario.

La página se puede encontrar en el siguiente link:

https://sites.google.com/site/degreethesisdiegopeluffo/improvedcbr

Figura 24. Pagina web" CBR with improved Adaptation and recovery stages"