



UNIVERSIDAD JUÁREZ AUTÓNOMA DE TABASCO

DIVISIÓN ACADÉMICA DE INFORMÁTICA Y SISTEMAS



8

Detección de anomalías en mastografías digitales usando el proceso KDD

Trabajo recepcional bajo la modalidad de Tesis

Que para obtener el grado de

Maestro en Sistemas Computacionales

Presenta

Eddy Sánchez de la Cruz

Asesor

M.I.S. Homero Alpuín Jiménez

Cuerpo Académico

Sistemas Inteligentes

4

Línea de Generación y Aplicación del Conocimiento

Optimización, Clasificación y Minería de Datos

Cunduacán, Tabasco

Noviembre 2012



UNIVERSIDAD JUÁREZ AUTÓNOMA DE TABASCO

DIVISIÓN ACADÉMICA DE INFORMÁTICA Y SISTEMAS



8

Detección de anomalías en mastografías digitales usando el proceso KDD

Trabajo recepcional bajo la modalidad de Tesis

Que para obtener el grado de

Maestro en Sistemas Computacionales

Presenta

Eddy Sánchez de la Cruz

Asesor

M.I.S. Homero Alpuín Jiménez

Revisores

M. en C. Guillermo de los Santos Torres

M. en C. José Luis Gómez Ramos

Dra. María del Pilar Pozos Parra

M.A. Pablo Payró Campos

M.S.C. Oscar Alberto Chávez Bosquez

Cunduacán, Tabasco

Noviembre 2012

Carta de Autorización

A quien corresponda:

El que suscribe, autoriza por medio del presente escrito a la Universidad Juárez Autónoma de Tabasco para que utilice tanto física como digitalmente la tesis de grado denominada, "**Detección de anomalías en mastografías digitales usando el proceso KDD**", de la cual soy autor y titular de los Derechos de Autor.

La finalidad del uso por parte de la Universidad Juárez Autónoma de Tabasco de la tesis antes mencionada, será única y exclusivamente para difusión, educación y sin fines de lucro; autorización que se hace de manera enunciativa más no limitativa para subirla a la Red Abierta de Bibliotecas Digitales (RABID) y a cualquier otra red académica con las que la Universidad tenga relación institucional.

Por lo antes manifestado, libero a la Universidad Juárez Autónoma de Tabasco de cualquier reclamación legal que pudiera ejercer respecto al uso y manipulación de la tesis mencionada y para los fines estipulados en este documento.

Se firma la presente autorización en la ciudad de Villahermosa, Tabasco a los 5 días del mes de noviembre del año 2012.

Autorizo



Eddy Sánchez de la Cruz



UNIVERSIDAD JUÁREZ AUTÓNOMA DE TABASCO

"ESTUDIO EN LA DUDA. ACCIÓN EN LA FE"

DIVISIÓN ACADÉMICA DE INFORMÁTICA Y SISTEMAS



Oficio No. 2523/12/DAIS/D
Octubre 10 de 2012

MIS. Homero Alpuín Jiménez
Profesor-Investigador
Presente

De acuerdo al artículo 44 fracción 3 del Reglamento de Estudios de Posgrado, de la Universidad Juárez Autónoma de Tabasco, me permito informar a Usted, que ha sido designado Director del trabajo de Tesis titulado "**Detección de anomalías en mastografías digitales usando el proceso KDD**", a realizar por el **C. Eddy Sánchez de la Cruz**, para obtener el grado de Maestro en Sistemas Computacionales.

Sin otro particular, aprovecho la ocasión para enviarle un afectuoso saludo.

Atentamente


Lic. Eduardo Cruces Gutiérrez
Director

Universidad Juárez Autónoma de Tabasco
División Académica de Informática y Sistemas

c.c.p. Lic. Martha Patricia Silva Payró.- Coordinadora de Investigación y Posgrado.
Archivo.
Consecutivo.

Miembro CUMEX desde 2008

UNA ALIANZA DE CALIDAD POR LA EDUCACIÓN SUPERIOR

Carretera Cunduacán-Jalpa Km. 1, Colonia Esmeralda, C.P. 86690. Cunduacán, Tabasco, México.
E-mail: direccion.dais@ujat.mx
Teléfonos: (993) 358 1500 ext. 6727; (914) 336 0616; Fax: (914) 336 0870



Recibi
20/10/2012



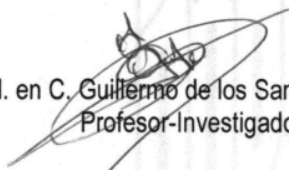
Universidad Juárez Autónoma de Tabasco

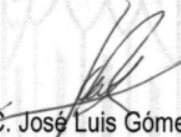
"Estudio en la duda, acción en la fe"

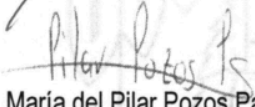
División Académica de Informática y Sistemas

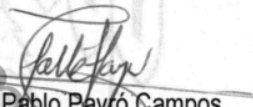
En la Universidad Juárez Autónoma de Tabasco, de acuerdo al Reglamento de Estudios de Posgrado vigente, se revisó el trabajo de investigación titulado "**Detección de anomalías en mastografías digitales usando el proceso KDD**", realizado por el LIA. Eddy Sánchez de la Cruz, para obtener el Grado de Maestro en Sistemas Computacionales bajo la modalidad de Tesis.

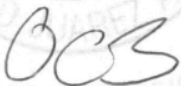
Los integrantes del jurado, después de revisar el trabajo, lo declararon aceptado.


M. en C. Guillermo de los Santos Torres
Profesor-Investigador


M. en C. José Luis Gómez Ramos
Profesor-Investigador


Dra. María del Pilar Pozos Parra
Profesora-Investigadora


M.A. Pablo Payró Campos
Profesor-Investigador


MSC. Oscar Alberto Chávez Bosquez
Profesor-Investigador



UNIVERSIDAD JUÁREZ AUTÓNOMA DE TABASCO

"ESTUDIO EN LA DUDA. ACCIÓN EN LA FE"

DIVISIÓN ACADÉMICA DE INFORMÁTICA Y SISTEMAS



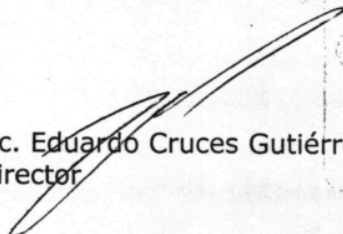
Oficio No.2453/12/DAIS/D
Octubre 4 de 2012

LIC. EDDY SÁNCHEZ DE LA CRUZ
PRESENTE

En virtud de que cumple satisfactoriamente los requisitos establecidos en el Reglamento de Estudios de Posgrado vigente en la Universidad, informo a Usted que se autoriza la impresión del trabajo de investigación **"Detección de anomalías en mastografías digitales usando el proceso KDD"**, para presentar Examen de Grado de la Maestría en Sistemas Computacionales, bajo la modalidad de Tesis.

Sin otro particular, aprovecho la ocasión para enviarle un afectuoso saludo.

Atentamente


Lic. Eduardo Cruces Gutiérrez
Director

c.c.p. Lic. Martha Patricia Silva Payró.- Coordinadora de Investigación y Posgrado.
Archivo.
Consecutivo.

Dedicatoria

A Mirta, mi *Dulcinea del Toboso*, por creer en mí.

A Dora mi madre, ejemplo de abnegación y lucha. Esto te lo debo a ti.

A mi padre (Arnulfo) y hermanas (Are y Vero), mi héroe y villanas favoritos :-D

A la Dra. Pilar Pozos-Parra, cuando la conocí simplemente recordé lo que alguien dijo del escritor ruso León Tolstói: «*Es el espíritu más grande del siglo XIX*».

Agradecimientos

A Dios por darme vida, fuerza, consejos, consuelo, esperanza y jalones de oreja.

A mi asesor, Mtro. Homero Alpuín, por guiarme en el desarrollo de la tesis.

A mis compañeros de posgrado: Darío, Efraín, Genaro, Gerardo, Germán, Lázaro y Pedro. Por su amistad.

Al Ing. Rubicel Cruz Romero y la Mtra. Martha Patricia por el apoyo brindado en todo momento en DAIS.

A mis revisores de tesis, por sus acertados comentarios y valiosas aportaciones.

Al Dr. Humberto Ochoa del IIT-UACJ por guiarme durante la estancia de investigación

Índice general

1. Generalidades	1
1.1. Introducción	1
1.2. Planteamiento del problema	2
1.2.1. Hipótesis	2
1.3. Objetivos	3
1.4. Justificación	3
1.5. Diseño de la investigación	4
2. Marco teórico	6
2.1. El proceso de Descubrimiento de Conocimiento en base de datos y la minería de datos	6
2.1.1. Definición de Knowledge Discovery in Databases (KDD)	6
2.1.2. Definición de Data Mining (DM)	6
2.1.3. Descubrimiento de conocimiento de bases de datos y Minería de datos	6
2.1.4. Tipos de datos	7
2.1.5. Clasificadores de aplicación en minería de datos	8
2.2. Imágenes digitales	14
2.2.1. Mastografías digitales	15
2.2.2. Ruido en imágenes	15
2.3. Método de Mejía para reducir ruido y realzar anomalías en mastografías digitales	17
2.3.1. Comentarios al método de Mejía	18
2.4. Análisis de imágenes médicas	18
2.5. Descripción del modelo de Desarrollo Rápido de Aplicaciones (DRA)	19
3. Desarrollo de la investigación	21
3.1. Análisis de requerimientos	25
3.1.1. Requerimientos	25
3.2. Semi-automatización del proceso KDD	26

ÍNDICE GENERAL

4. Pruebas y resultados	33
4.1. Algoritmos de clasificación para minería de datos	33
4.1.1. Selección del clasificador	35
4.2. Experimentos y análisis	36
4.3. Evaluación	42
5. Conclusiones y trabajos futuros	44

Índice de figuras

1.1. Gráfica del diseño de la investigación. Fuente: elaboración propia basada en (Olmos-Pineda y González-Bernal, 2007).	5
2.1. Proceso KDD. Fuente: (Olmos-Pineda y González-Bernal, 2007).	7
2.2. Imagen digital bidimensional. Fuente: Elaboración propia.	15
2.3. Ruidos que afectan una imagen. Fuente: (Azouzi y Vilaplana, 2006).	16
2.4. Diagrama de bloques del método de Mejía. Fuente: (Mejía, 2010).	17
2.5. Anormalidades en recuadros, rodeadas por fondo (tejido circundante del seno). Fuente: (Mejía, 2010).	18
2.6. Toma de una mastografía. Fuente: (Sánchez, 2010).	19
2.7. Zona que presenta anomalía. Fuente: Elaboración propia.	19
2.8. Modelo DRA. Fuente: (Pressman, 2006).	20
3.1. Filtrado. Fuente: (Sánchez, 2010).	23
3.2. Segmentación de la zona de interés. Fuente: elaboración propia	23
3.3. Histograma de la zona segmentada y normalizada. Fuente: Elaboración propia.	24
3.4. Herramienta de MD Weka. Fuente: (University Waikato, 2011).	26
3.5. Flujo de información. Fuente: elaboración propia.	27
3.6. Modelado de datos. Fuente: (Sánchez, 2010).	27
3.7. Módulo «Pantalla inicio». Fuente: (Sánchez, 2010).	28
3.8. Menú de opciones. Fuente: (Sánchez, 2010).	28
3.9. Módulo «Filtrar». Fuente: (Sánchez, 2010).	29
3.10. Módulo «Segmentar». Fuente: (Sánchez, 2010).	30
3.11. Tipo de anomalía. Fuente: (Sánchez, 2010).	31
3.12. Módulo «Clasificar». Fuente: (Sánchez, 2010).	32
4.1. Gráfica de resultados de los datasets. Fuente: elaboración propia.	40

Índice de tablas

2.1. Método bayesiano	9
2.2. Método de funciones de clasificación	10
2.3. Método perezos	10
2.4. Metclasificadores (1)	11
2.5. Metclasificadores (2)	12
2.6. Método varios	12
2.7. Método de generación de reglas	13
2.8. Método de generación de árboles de decisión	14
3.1. DBs de mastografías digitales de acceso público (1)	21
3.2. DBs de mastografías digitales de acceso público (2)	22
3.3. DB MIAS	22
4.1. Pruebas / Clasificadores (1)	34
4.2. Pruebas / Clasificadores (2)	35
4.3. Porcentajes más altos	36
4.4. Experimento con el primer conjunto de datos	37
4.5. Experimento con el segundo conjunto de datos	38
4.6. Experimento con el tercer conjunto de datos	39
4.7. Experimento con el cuarto conjunto de datos	40
4.8. Instancias incorrectamente clasificadas	41
4.9. Clasificación de instancias M	42
4.10. Comparación de resultados obtenidos con otros trabajos previamente reportados en la literatura.	43

Índice de Algoritmos

1. LogitBoost	25
---------------	----

Resumen

El cáncer de mama es la neoplasia más frecuente en las mujeres y se manifiesta en un crecimiento no controlado de células malignas en el tejido mamario. La mastografía o mamografía es un método para detectar anomalías en la mama.

El problema que se aborda en esta investigación es la disminución de *falsos negativos*, que son los tumores que si existen, pero que no son detectados por el especialista en una mastografía.

Como aporte de esta investigación, se propone el uso del proceso de descubrimiento en bases de datos *KDD* (*Knowledge Discovery in Databases*) para detectar anomalías en mastografías digitales. Dentro del proceso KDD se utiliza, para filtrar las imágenes mastográficas, el *método de Mejía*, el cual busca obtener, en una nueva representación de la imagen, la extracción de las características que son difíciles de visualizar y procesar en la imagen original (Mejía et al., 2009). Y para la detección se utiliza el clasificador *Decorate*, el cual se basa en una construcción de conjuntos de clasificadores mediante el uso de ejemplos de entrenamiento artificiales (Witten y Frank, 2005).

Las pruebas arrojan 95 % de instancias correctamente clasificadas, estos resultados son altamente esperanzadores, ya que su impacto se traduce en que a una persona se le puede detectar a tiempo alguna anomalía cancerígena y recibir tratamiento oportuno.

Capítulo 1

Generalidades

1.1. Introducción

El análisis de datos es una tarea que consiste en buscar o encontrar tendencias o variaciones de comportamiento en los datos, de tal manera que esta información resulte de utilidad para los usuarios finales. A estas tendencias o variaciones se les conoce como patrón (Witten y Frank, 2005). Si los patrones son útiles y de relevancia para quien lo interpreta, entonces se le llama conocimiento. Lo anterior se logra mediante el proceso de Descubrimiento de Conocimiento en Bases de Datos (KDD por las siglas en inglés de *Knowledge Discovery in Databases*). En un principio, el análisis de los datos se realizaba de forma manual, empleando técnicas estadísticas. Sin embargo, actualmente esta forma de análisis resulta inviable por la gran cantidad de datos que puede contener una Base de Datos (DB por las siglas en inglés de *Data Base*), además de que existen una gran cantidad de formatos para los datos, como tablas (bases de datos relacionales), secuencias (Bandyopadhyay et al., 2005), grafos (Olmos et al., 2005), imágenes (González, 2007) o audio, lo cual aumenta la complejidad de un análisis manual. Bajo estas circunstancias es como surge la Minería de Datos (DM por las siglas en inglés de *Data Mining*), que es un proceso automático o semiautomático que busca descubrir patrones ocultos en un conjunto de datos y que además, sean potencialmente útiles para los usuarios de las DB (Witten y Frank, 2005).

La DM es una fase dentro del proceso KDD, sin embargo muchas veces en la literatura KDD y DM se refieren a lo mismo. La DM se ha aplicado al área de salud para reconocer patrones en imágenes médicas (Olmos-Pineda y González-Bernal, 2007). La idea principal es realzar y extraer la información contenida en las imágenes para su interpretación y análisis por parte de los humanos (Pajares-Martinsanz, 2005). Debido a la complejidad de extraer información de las imágenes se han empleado diferentes estrategias para atacar el problema tales como Redes Neuronales Artificiales (RNA) (Méndez et al., 2007), redes bayesianas (Coto, 2003), lógica difusa (Valencia, 2004), razona-

miento basado en casos (Coto, 2003), algoritmos bioinspirados (Sánchez y Torres, 2009) e híbridos de éstos. La medicina se ha beneficiado ya que para los especialistas las imágenes son de gran utilidad para poder encontrar, mediante un análisis de éstas, posibles anomalías y diagnosticar enfermedades. La dificultad por parte del especialista se hace notar debido a que la revisión de la imagen no garantiza el encuentro de la totalidad de las anormalidades, es por ello que esta investigación sirve como herramienta de soporte en la decisión médica para encontrar posibles regiones anómalas.

La metodología aplicada en esta investigación consta de cinco fases, las cuales son un compendio del proceso KDD, y son: fase de selección, fase de preparación de los datos, fase de transformación de los datos, fase de minería de datos y fase de evaluación de patrones.

1.2. Planteamiento del problema

En el análisis de imágenes de mastografía muchas veces se presentan *falsos negativos*. Un falso negativo ocurre si el especialista analiza una placa mastográfica e indica que no existe tumor cancerígeno cuando realmente sí existe. (Cheng et al., 2007) señalan que después de hecha la revisión en la imagen por el especialista, entre un 25% de los tumores visibles no son detectados (falso negativo). Esto significa que si a un paciente se le detectan tres tumores cancerígenos y se los extirpan, existe el riesgo que aún tenga alojado otro tumor que no fue detectado y que sigue expandiéndose; además que puede ocurrir una metástasis¹. Cabe aclarar que ese tumor puede ser detectado en otra sesión, sin embargo esa detección será tardía, y un retardo en este caso se traduce en una pérdida humana.

Es de suma importancia entonces disminuir los falsos negativos, ya que si esto se logra el impacto social sería de grandes alcances: el especialista tendría mayor certeza de encontrar las anormalidades en el momento oportuno y el paciente sería tratado a tiempo.

1.2.1. Hipótesis

Dado que el porcentaje de un diagnóstico efectivo en mastografías es de 75% (Torres-Arreola y Doubova, 2006), y que el porcentaje de un diagnóstico efectivo en imágenes médicas digitales, aplicando el proceso KDD, es de 90% (Olmos-Pineda y González-Bernal, 2007), se plantea que:

¹Metástasis: Propagación de células cancerígenas alojadas en la mama a otro órgano del cuerpo.

Al usar el proceso KDD en mastografías digitales se puede reducir de 25 % a 10 % el porcentaje de dar un falso negativo, y con ello obtener un porcentaje de 90 % en la detección de anormalidades cancerígenas.

1.3. Objetivos

Objetivo general

Aplicar el proceso KDD para apoyar el reconocimiento de anormalidades en mastografías digitales con la finalidad de disminuir los falsos negativos.

Objetivos específicos

1. Obtener una base de datos de mastografías digitales de acceso público.
2. Aplicar el proceso de mejoramiento de datos en mastografías digitales.
3. Transformar los datos (imágenes) a formato de entidad relación para construir una base de conocimiento.
4. Desarrollar un prototipo de sistema que aplique DM para la detección. El sistema será específico para la base de datos del objetivo 1.
5. Evaluar la detección que se obtuvo comparándola con el estado del arte.

1.4. Justificación

La incorporación de la DM al área médica sirve para analizar imágenes en la detección de anormalidades y diagnosticar tumores cancerígenos (Coto, 2003). Al reducir de 25 % a 10 % el porcentaje de dar un falso negativo existe la posibilidad de elevar la esperanza de vida de un paciente potencial de cáncer de mama, ya que el tumor que es detectado por el especialista en una primera revisión, pasa a ser sospechoso y con ello se hace necesario la toma de otra mastografía desde otro ángulo para confirmar la presencia de la anormalidad.

Cabe enfatizar que la temprana detección de un tumor cancerígeno en la mama puede ayudar a tratar el problema a tiempo (Sánchez-Domínguez y Paz-Morales, 2006). Por lo anterior esta investigación es de gran valía para la población que padece cáncer de mama y aún no lo han detectado. Los resultados de esta investigación pueden servir para que el especialista tenga una segunda opinión en el análisis de una mastografía digital y con ello aumentar la fiabilidad de detectar un tumor cancerígeno.

La aportación de esta investigación es el uso de el proceso KDD en mastografías digitales, empleando además el método propuesto en (Mejía et al., 2009) en la fase de limpieza y el clasificador *Decorate* para la detección de la anormalidad cancerígena.

1.5. Diseño de la investigación

A continuación se describen los pasos de forma breve, ya que tanto la explicación detallada como las herramientas utilizadas se abordan en el apartado 3.

Descripción del diseño de la investigación:

1. Teniendo una imagen, se filtra para reducir el ruido.
2. Se segmenta la zona de interés y se normaliza.
3. Después de normalizada, se obtiene el histograma de frecuencia de niveles de gris.
4. Se exporta cada histograma a una tabla para crear la base de conocimiento.
5. Se prueban los algoritmos de clasificación.
6. Se utiliza el clasificador *Decorate* por encontrar el mejor desempeño.
7. Se desarrolla el prototipo para 1) implementar el clasificador *Decorate*, 2) semi-automatizar el proceso KDD y 3) hacer la detección de la anormalidad cancerígena.
8. Teniendo nuevas mastografías digitales, se segmentan para obtener muestras y hacer nuevas pruebas con el fin de corroborar la correcta detección del prototipo.

Representación gráfica el diseño de la investigación.

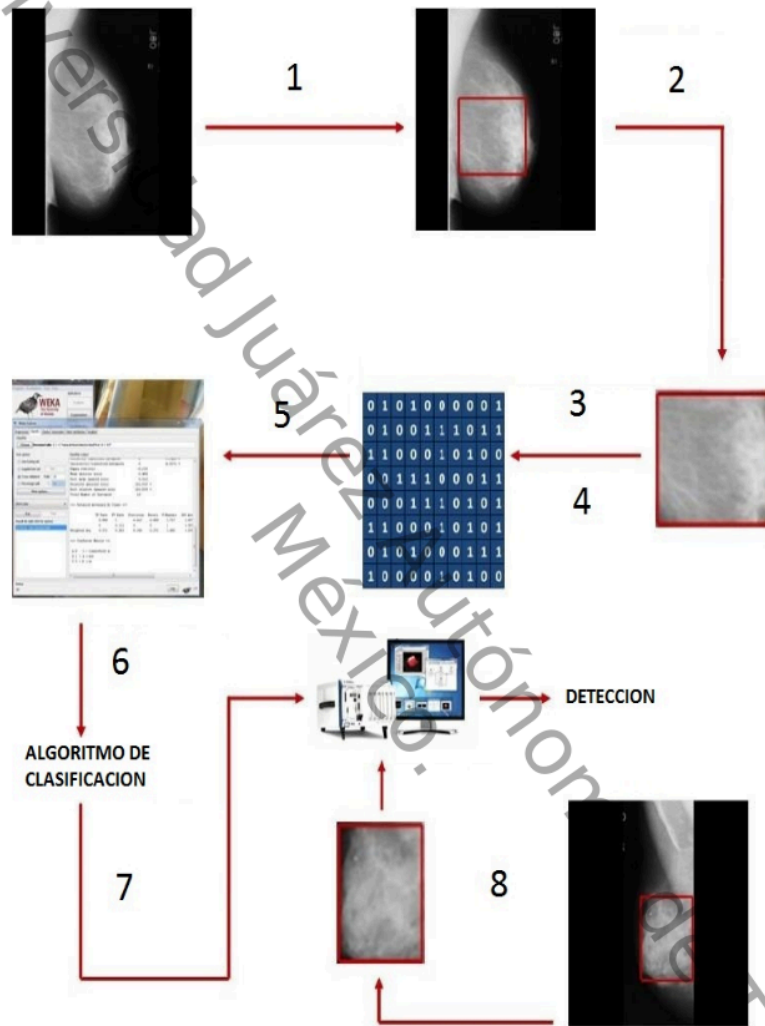


Figura 1.1: Gráfica del diseño de la investigación. Fuente: elaboración propia basada en (Olmos-Pineda y González-Bernal, 2007).

Capítulo 2

Marco teórico

2.1. El proceso de Descubrimiento de Conocimiento en base de datos y la minería de datos

2.1.1. Definición de Knowledge Discovery in Databases (KDD)

(Saldana y Flores, 2005) dicen que KDD puede definirse como una serie de pasos que permiten identificar patrones poco obvios dentro de los datos.

(Ramos et al., 2007) definen KDD como el proceso no trivial de identificación de los datos de patrones válidos de las bases de datos para interpretar grandes cantidades de información y encontrar las relaciones o patrones entre los datos, que se encuentran en las bases de datos.

2.1.2. Definición de Data Mining (DM)

(Ng y Pei, 2007) describe a la DM como una disciplina que combina técnicas de inteligencia artificial, aprendizaje computacional, probabilidad, estadística y base de datos para extraer información y conocimientos útiles a partir de grandes cantidades de datos.

(Martínez, 2003) dice que la DM se fundamenta en el concepto de escarbar en la información almacenada para descubrir elementos de utilidad desde grandes cantidades de datos almacenadas, con el objetivo de detectar patrones de comportamiento consistentes o relacionales entre los diferentes campos de una base de datos para aplicarlos a nuevos conjuntos de datos.

2.1.3. Descubrimiento de conocimiento de bases de datos y Minería de datos

Como ya se mencionó en el apartado 1.1 el proceso KDD comúnmente se similariza con la DM, sin embargo, teniendo en mente las definiciones

anteriores, podemos decir que el proceso KDD es una secuencia de fases cuya finalidad es descubrir patrones de comportamiento; ahora, la DM apoya fuertemente este descubrimiento de patrones, es decir, la DM es una fase dentro del proceso KDD. Sin la DM tal descubrimiento de patrones no se lograría y, a su vez, la DM por si sola no funcionaría, pues los datos con los que va a tratar necesitan un tratamiento de mejora y análisis, el cual le brindan las otras fases del proceso KDD. Por lo cual se concluye, según (Olmos-Pineda y González-Bernal, 2007), que MD es una fase dentro del proceso KDD, tal como se ve en la Figura 2.1

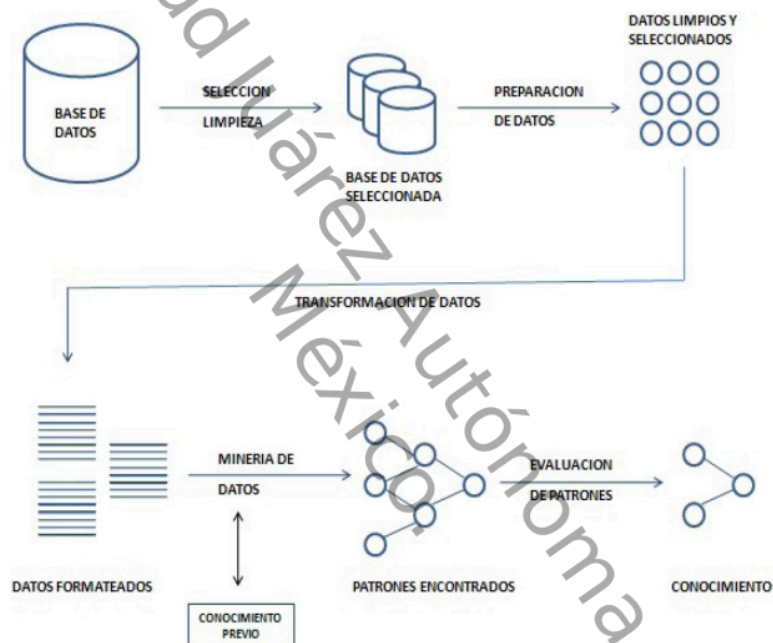


Figura 2.1: Proceso KDD. Fuente: (Olmos-Pineda y González-Bernal, 2007).

2.1.4. Tipos de datos

Es importante saber dónde se puede aplicar el proceso KDD. A continuación se muestra una clasificación, según (Larranaga y Inza, 2007), de los tipos de datos con los que se puede trabajar:

- Bases de datos relacionales.
 - Colección de relaciones (tablas). Tabla como conjunto de atributos (variables, columnas, campos) conteniendo tuplas (casos, filas, registros).

- Presentación tabular: atributo-valor (vista minable).
- Bases de datos espaciales: datos geográficos, imágenes médicas, redes de transporte o tráfico.
- Bases de datos temporales: distintos instantes o intervalos temporales.
- Bases de datos documentales: documentos de texto, variables desde palabras hasta resúmenes
- Bases de datos multimedia: imágenes, audio y video.
- La World Wide Web: repositorio de información más grande y diverso en la actualidad.
 - Minería del contenido: encontrar patrones en las páginas web.
 - Minería de la estructura: estudia los hipervínculos y URLs.
 - Minería del uso: análisis de la navegación

2.1.5. Clasificadores de aplicación en minería de datos

La DM se vale de diferentes clasificadores para el descubrimiento de patrones, estos clasificadores se encuentran agrupados en métodos. Los clasificadores pueden combinarse para obtener resultados satisfactorios. En este apartado se mencionan los clasificadores que Weka proporciona, los cuales fueron usados para pruebas en esta investigación:

Método bayesiano (*Bayes*)

Dentro de éste método se encuentran los clasificadores (ver Tabla 2.1) que se basan en las redes bayesianas.

Redes bayesianas Una red bayesiana (Fernández, 2009) es un grafo acíclico dirigido en el que cada nodo representa una variable y cada arco, representa una dependencia probabilística, en la cual se especifica la probabilidad condicional de cada variable dados sus padres, la variable a la que apunta el arco es dependiente (causa-efecto) de la que está en el origen de éste. La topología o estructura de la red nos da información sobre las dependencias probabilísticas entre las variables pero también sobre las independencias condicionales de una variable (o conjunto de variables) dada otra u otras variables, dichas independencias, simplifican la representación del conocimiento (menos parámetros) y el razonamiento (propagación de las probabilidades).

Teorema de Bayes Sean A y B dos sucesos aleatorios cuyas probabilidades se denotan por $p(A)$ y $p(B)$ respectivamente, verificándose que

$p(B) > 0$. Supongamos conocidas las probabilidades *a priori* de los sucesos A y B , es decir, $p(A)$ y $p(B)$, así como la probabilidad condicionada del suceso B dado el suceso A , es decir $p(B|A)$. La probabilidad *a posteriori* del suceso A conocido que se verifica el suceso B , es decir $p(A|B)$, puede calcularse a partir de la siguiente fórmula:

$$p(A|B) = \frac{p(A, B)}{p(B)} = \frac{p(A)p(B|A)}{p(B)} = \frac{p(A)p(B|A)}{\sum_A p(A)p(B|A)} \quad (2.1)$$

Método	Clasificador	Función
Bayes	BayesNet	Redes bayesianas de aprendizaje
	ComplementNaiveBayes	Construye un complemento de Naïve Bayes
	DMNBtext	Clase para la construcción y el uso de un clasificador Naïve Bayes discriminativo-Multinomial
	NaiveBayes	Probabilístico estándar de un clasificador bayesiano
	NaiveBayesMultinomial	Versión multinomial de Naïve Bayes
	NaiveBayesMultinomial Updateable	Clasificador Naïve Bayes incremental que aprende de un caso a la vez
	NaiveBayesUpdate	Este clasificador utiliza una precisión predeterminada de 0.1 para los atributos numéricos

Tabla 2.1: Método bayesiano

Método de funciones de clasificación (*Functions*)

La categoría de las funciones de la Tabla 2.2 incluye un grupo diverso de clasificadores que se pueden escribir como ecuaciones matemáticas de una manera bastante natural (Witten y Frank, 2005).

Método	Clasificador	Función
Functions	Logistic	Construir los modelos lineales de regresión logística
	MultilayerPerceptron	Red neuronal propagación hacia atrás
	RBFNetwork	Implementa una red de funciones de base radial
	SimpleLogistic	Construir los modelos lineales de regresión logística con la selección de atributos incorporados
	SMOreg	Implementa un algoritmo de optimización de mínimos de Alex Smola y Bernhard Scholkopf para la formación de un modelo de regresión

Tabla 2.2: Método de funciones de clasificación

Método perezos (*Lazy*)

Los clasificadores de éste método (ver Tabla 2.3) clasifican una nueva instancia, utilizando las k instancias más cercanas a un elemento $\langle a_1, \dots, a_n \rangle$ que deseamos clasificar. Se utiliza la función de distancia euclidiana (Witten y Frank, 2005):

$$d(\langle a_1, \dots, a_n \rangle, \langle b_1, \dots, b_n \rangle) = (\sum (a_i - b_i)^2)^{1/2} \quad (2.2)$$

Método	Clasificador	Función
Lazy	IB1	Básico del vecino más cercano basado en instancia de aprendizaje
	IBk	Clasificador k-vecino más cercano
	Kstar	Vecino más cercano con función de distancia generalizada
	LWL	Clasificador general de aprendizaje a nivel local ponderado

Tabla 2.3: Método perezos

Metaclasificadores (*Meta*)

Los metaclasificadores (ver Tabla 2.4 y 2.5), surgen con el objetivo de mejorar la precisión de las predicciones (Witten y Frank, 2005), ha surgido

CAPÍTULO 2. MARCO TEÓRICO

un interés creciente en la definición de métodos que combinan varios modelos clasificatorios. Se trata, entonces, de combinar las predicciones del conjunto de modelos, normalmente por votación, para clasificar nuevos ejemplos. La precisión obtenida por esta combinación supera, generalmente, la precisión de cada componente individual del conjunto.

Método	Clasificador	Función
Meta	AdaBoostM1	Clase para generar un clasificador nominal utilizando el método de AdaBoostM1
	AttributeSelectedClassifier	Reducir la dimensionalidad de los datos mediante la selección de atributos
	Bagging	Clase para el ensacado de un clasificador para reducir la varianza, trabaja para la la regresión también
	ClassificationViaClustering	Un simple meta-clasificador que utiliza una agrupación para la clasificación
	ClassificationViaRegression	Realizar la clasificación utilizando un método de regresión
	CVPParameterSelection	Realiza la selección de parámetros de validación cruzada
	Dagging	Crea un número de desunión, y se alimenta de cada conjunto de datos
	Decorate	Construcción de conjuntos de clasificadores mediante el uso de ejemplos de entrenamiento artificiales
	END	Un meta-clasificador para el manejo de multi-clase mediante la construcción y el conjunto de dicotomías anidados
	FilteredClassifier	Ejecuta un clasificador de los datos filtrados
	MultiBoostAB	Combina <i>boosting</i> y <i>bagging</i> mediante el método MultiBoosting
	MultiClassClassifier	Usa un clasificador de dos clases para conjunto de datos multiclase
	MultiScheme	Utiliza validación cruzada para seleccionar un clasificador de varios candidatos

Tabla 2.4: Metaclasificadores (1)

CAPÍTULO 2. MARCO TEÓRICO

Método	Clasificador	Función
Meta	OrdinalClassClassifier	Aplica algoritmos de clasificación estándar para problemas con un valor de clase ordinal
	RacedIncrementalLogitBoost	Clasificador para el aprendizaje incremental de grandes conjuntos de datos
	RandomCommitee	Construye un conjunto de clasificadores base aleatoria
	RandomSubSpace	Este método construye un clasificador de árbol de decisión basado en que mantiene una mayor precisión en los datos de formación y mejora la precisión de la generalización
	RotationForest	Puede hacer clasificación y regresión en función de la base de aprendizaje
	Stacking	Combina varios clasificadores utilizando el método de apilamiento
	StackingC	Versión más eficiente de apilamiento
	Vote	Combina clasificadores utilizando el promedio de las estimaciones de probabilidad o predicciones numéricas

Tabla 2.5: Metaclasificadores (2)

Método varios *Misc*

En esta categoría (ver Tabla 2.6) se incluyen dos clasificadores simples: Hyperpipes, para problemas de clasificación discreta, y VFI (intervalos de votación característica) que clasifica los casos de prueba mediante el voto (Witten y Frank, 2005).

Método	Clasificador	Función
Misc	HyperPipes	Muy simple, aprende rápido sobre la base de hipervolumenes en el espacio de instancias
	VFI	Característica de votación: método de los intervalos, simple y rápido

Tabla 2.6: Método varios

Método de generación de reglas (*Rules*)

En estos clasificadores (ver Tabla 2.7) se descubren patrones de la forma «SI X entonces Y» (Witten y Frank, 2005).

Método	Clasificador	Función
Rules	ConjunctiveRule	Aprendizaje con reglas sencillas conjuntivas
	DecisionTable	Crea una tabla simple de decisión de la mayoría de los clasificados
	DTNB	Clase para la construcción y el uso de una tabla de decisión con clasificador bayesiano híbrido
	JRip	Usa el algoritmo RIPPER para reglas de inducción rápidas y eficaces
	NNge	Método del vecino más cercano de generación de reglas que utilicen ejemplares no-interesados generalizados
	OneR	Clasificador 1R
	PART	Obtiene reglas de los árboles de decisión parcial construido con J4.8
	Ridor	Regla de aprendizaje odulación hacia abajo
	ZeroR	Predecir la clase de la mayoría (si es nominal) o el valor medio (si es numérico)

Tabla 2.7: Método de generación de reglas

Método de generación de árboles de decisión (*Trees*)

Un árbol de decisión (Witten y Frank, 2005) es una estructura que se compone de nodos (internos y hojas) y de arcos. Sus nodos internos están caracterizados por uno o varios atributos de prueba y de estos nodos se desprenden uno o más arcos. Cada uno de estos arcos tiene asociado un valor del atributo de prueba y estos valores determinan qué camino seguir en el recorrido del árbol. Los nodos hoja contienen información que permite determinar la pertenencia del objeto a una clase. Las características principales de un árbol de decisión son: construcción sencilla, no necesita determinar de antemano parámetros para su construcción, puede tratar problemas multi-clase de la misma forma en que trabaja con problemas de dos clases, facilidad para ser representado mediante un conjunto de reglas y la fácil interpretación de su estructura (ver Tabla 2.8).

Método	Clasificador	Función
Trees	BFTree	Clase para la construcción de un clasificador de árbol de decisión primero-mejor
	DecisionStump	Construye árboles de decisión de un nivel
	FT	Clase para construir árboles de función, los cuales son árboles de clasificación que tienen funciones de regresión logística en los nodos internos y/o hojas
	J48	Aprendizaje mediante árbol de decisión C4.5 (C4.5 implementa la revisión 8)
	J48graft	Clase para generar un injerto (podado o sin podar) de un árbol de decisión C4.5
	LADTree	Clase para generar un árbol de decisión multi-clase, utilizando el algoritmo LogitBoost
	LMT	Construye árboles de modelo logístico
	NBTree	Construye un árbol de decisión con clasificadores Naive Bayes en las hojas
	RandomForest	Construye bosques al azar
	RandomTree	Construye un árbol que tiene en cuenta un determinado número de rasgos al azar en cada nodo
	REPTree	Aprendizaje rápido de árboles que usa poda de error reducido
	SimpleCart	Clase que implementa poda de complejidad de costo mínimo

Tabla 2.8: Método de generación de árboles de decisión

2.2. Imágenes digitales

Una imagen digital bidimensional (Figura 2.2) es una función que a cada par de coordenadas (x, y) asocia un valor relativo a alguna propiedad del punto que representa (por ejemplo su brillo o su matiz).

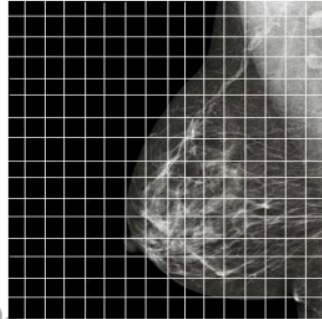


Figura 2.2: Imagen digital bidimensional. Fuente: Elaboración propia.

2.2.1. Mastografías digitales

En una mastografía digital se registra la mastografía convencional en código de computadora en vez de registrarla en película de radiografía. Las ventajas que ofrecen son:

- Las mastografías digitales pueden almacenarse y localizarse electrónicamente, lo cual hace que se puedan tener consultas a larga distancia con más facilidad con otros expertos.
- Ya que las mastografías digitales pueden ser ajustadas por el radiólogo, es posible notar diferencias sutiles entre los tejidos.
- La mastografía digital puede reducir el número de procedimientos necesarios de seguimiento, por ejemplo, biopsias.
- Se promueve la conservación ecológica, ya que se reduce la producción de tantas placas mastográficas que después de ser utilizadas van a la basura.

2.2.2. Ruido en imágenes

La mayoría de las veces, los datos originales en una imagen o en una secuencia de video contienen información que es irrelevante para una aplicación dada. En tales casos, los datos se deben simplificar, removiendo toda aquella información irrelevante. La simplificación controla la cantidad y naturaleza de la información que es preservada. Además los datos simplificados deben contener áreas fáciles de segmentar. Por ejemplo, la simplificación puede reducir la complejidad de texturas, eliminando ruido o eliminar objetos muy pequeños para un tamaño dado (Azouzi y Vilaplana, 2006).

Todas las imágenes tienen cierta cantidad de ruido (ver Figura 2.3), la cual se puede deber a la cámara o al medio de transmisión de la señal. Generalmente el ruido se manifiesta como píxeles aislados que toman un nivel de gris diferente al de sus vecinos. Los tipos de ruidos más comunes son los siguientes (Elizondo, 2002):

- **Gaussiano:** Son pequeñas variaciones en la imagen; generalmente se debe a diferentes ganancias en la cámara, ruido en los digitalizadores, perturbaciones en la transmisión, etc. Se considera que el valor final del pixel sería el ideal más una cantidad correspondiente al error que puede describirse como una variable aleatoria gaussiana.
- **Impulsional:** Aquí el valor que toma el pixel no tiene relación con el valor ideal, sino con el valor del ruido que toma valores muy altos o bajos (puntos blancos y/o negros) causados por una saturación del sensor o por un valor mínimo captado, si se ha perdido la señal en ese punto. Se encuentran también al trabajar con objetos a altas temperaturas, ya que las cámaras tienen una ganancia en el infrarrojo que no es detectable por el ojo humano; por ello las partes más calientes de un objeto pueden llegar a saturar un pixel.



Figura 2.3: Ruidos que afectan una imagen. Fuente: (Azouzi y Vilaplana, 2006).

Para resolver el problema de ruido se utilizan los filtros. En el siguiente apartado se explica a detalle el método de Mejía (Mejía et al., 2009) para reducir ruido y resaltar anomalías cancerígenas en mastografías digitales.

2.3. Método de Mejía para reducir ruido y realzar anomalías en mastografías digitales

(Mejía et al., 2009) realzan las anomalías del seno de la mujer, en mastografías digitales, para que éstas sean más fáciles de distinguir a simple vista por el radiólogo y con ello disminuir la posibilidad de que pasen desapercibidas.

El método que propone (Mejía et al., 2009) está basado en el enfoque clásico de los métodos que emplean transformadas para el procesamiento de imágenes. Éste, en forma general, se puede describir en los siguientes pasos:

1. Aplicar una transformación a la imagen que se va a analizar.
2. Realizar el procesamiento y análisis de las características de interés.
3. Aplicar la transformada inversa para regresar a la imagen original.

Este proceso se ilustra en el diagrama de bloques de la Figura 2.4

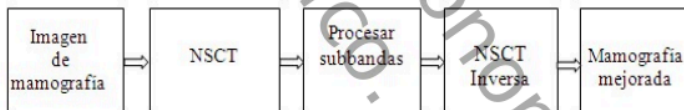


Figura 2.4: Diagrama de bloques del método de Mejía. Fuente: (Mejía, 2010).

La entrada al sistema es la mastografía digital, esta imagen se transforma al dominio de la Transformada de contorno no submuestreada (NSCT por las siglas en inglés de *NonSubsampled Contourlet Transform*).

Después de procesar las subbandas, se aplica la transformada inversa para obtener finalmente una imagen mejorada. En este caso, se tienen dos regiones que se desean diferenciar en la mamografía: 1) las anomalías y 2) el fondo, tal como se muestra en la Figura 2.5. En la región donde se encuentran las anomalías se desea que la intensidad del pixel aumente, para que su presencia sea evidente, y en la región del fondo se desea que la intensidad disminuya (Mejía et al., 2009).

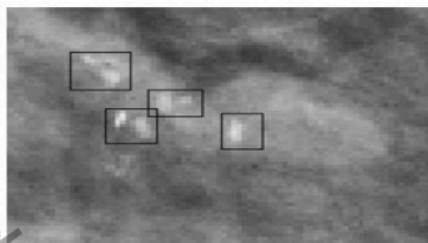


Figura 2.5: Anormalidades en recuadros, rodeadas por fondo (tejido circundante del seno). Fuente: (Mejía, 2010).

2.3.1. Comentarios al método de Mejía

El método presentado en (Mejía et al., 2009) es capaz de lograr un contraste entre las anormalidades cancerígenas y el fondo en mastografías digitales. Por ello se eligió este método para realizar el filtro de las imágenes.

2.4. Análisis de imágenes médicas

(Delrieux, 2007) dice que dado el notorio avance de la radiología computacional y el diagnóstico basado en imágenes, es indudablemente un área de oportunidad para hacer investigación de vanguardia y que sus resultados benefician, sobre todo, a la población.

El análisis de imágenes médicas es un tema de gran relevancia por el hecho de brindar soporte y apoyo al diagnóstico médico. Se han hecho trabajos para detectar anormalidades en imágenes obtenidas de mastografía (Figura 2.6), empleando algoritmos que implementan diferentes estrategias, tales como Redes Neuronales Artificiales (RNA) (Méndez et al., 2007), redes bayesianas (Coto, 2003), lógica difusa (Valencia, 2004), razonamiento basado en casos (Coto, 2003) e híbridos de éstos.

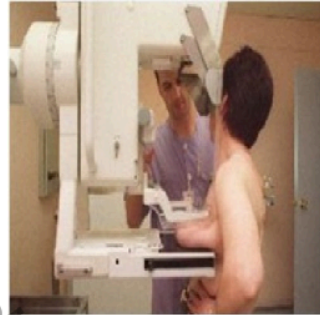


Figura 2.6: Toma de una mastografía. Fuente: (Sánchez, 2010)

Sabiendo que una imagen es, desde el punto de vista computacional, una matriz construida por píxeles, lo que se requiere es encontrar en esa matriz zonas que presenten anomalías (Figura 2.7). En las imágenes obtenidas por mastografía pueden apreciarse diferentes tipos de formaciones o anomalías celulares que revelen la presencia de un tumor maligno.

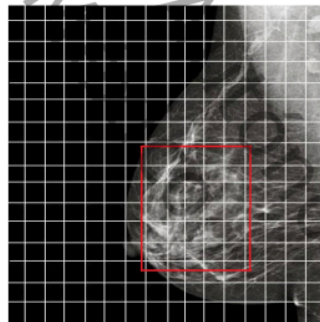


Figura 2.7: Zona que presenta anomalía. Fuente: Elaboración propia.

2.5. Descripción del modelo de Desarrollo Rápido de Aplicaciones (DRA)

El Desarrollo Rápido de Aplicaciones (Pressman, 2006) es un modelo de proceso de desarrollo del software lineal secuencial que enfatiza un ciclo de desarrollo extremadamente corto (Figura 2.8). DRA, así mismo, es un modelo a «alta velocidad» en el que se logra el desarrollo rápido utilizando un enfoque de construcción basado en componentes. Si se comprenden bien los requisitos y se limita el ámbito del proyecto, el proceso DRA permite al equipo de desarrollo crear un «sistema completamente funcional» den-

tro de periodos cortos de tiempo. Para esta investigación, el enfoque DRA comprende las siguientes fases:

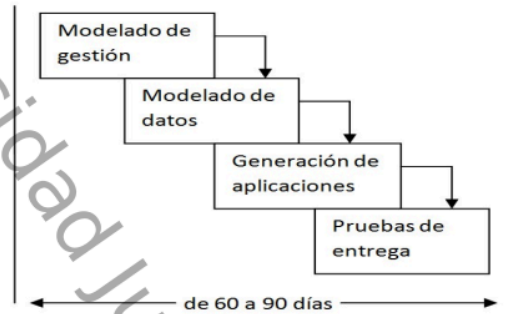


Figura 2.8: Modelo DRA. Fuente: [Pressman, 2006](#).

- **Modelado de gestión:** el flujo de información entre las funciones de gestión se modela de forma que responda a las siguientes preguntas: ¿Qué información conduce el proceso de gestión? ¿Qué información se genera? ¿Quién la genera? ¿A dónde va la información? ¿Quién la procesa?
- **Modelado de datos:** el flujo de información definido como parte de la fase de modelado de gestión se refina como un conjunto de objetos de datos necesarios para apoyar la empresa. Se definen las características (llamadas atributos) de cada uno de los objetos y las relaciones entre estos objetos.
- **Generación de aplicaciones:** El DRA asume la utilización de técnicas de cuarta generación. En lugar de crear software con lenguajes de programación de tercera generación, el proceso DRA trabaja para volver a utilizar componentes de programas ya existentes (cuando es posible) o a crear componentes reutilizables (cuando sea necesario). En todos los casos se utilizan herramientas automáticas para facilitar la construcción del software.
- **Pruebas de entrega:** Como el proceso DRA enfatiza la reutilización, ya se han comprobado muchos de los componentes de los programas. Esto reduce tiempo de pruebas. Sin embargo, se deben probar todos los componentes nuevos y se deben ejercitar todas las interfaces a fondo.

CAPÍTULO 3. DESARROLLO DE LA INVESTIGACIÓN

Nombre	Descripción
LLNL/UCSF	Base de datos creada por Lawrence Livermore National Laboratories (LLNL) y el Departamento de Radiología de la Universidad de California en San Francisco (UCSF). Fuente: mammo-db-help@llnl.gov
Nijmegen	Base de datos creada por el National Expert and Training Centre for Breast Cancer Screening y por el Departamento de Radiología de la Universidad de Nijmegen (Holanda). Fuente: nico@mbfys.kun.nl

Tabla 3.2: DBs de mastografías digitales de acceso público (2)

- Se seleccionó la base de datos reducida de acceso público MIAS¹ (Tabla 3.3), la cual ha sido usada para prueba de investigaciones en (Rangayyan et al., 1997), (Antonie et al., 2001), (Frag y Mashali, 2005), (Quintero et al., 2008), (Mejía et al., 2009) y la cual también recomienda la University of South Florida en (Heath et al., 2010):

Nombre	Descripción
MIAS	Base de datos reducida que contiene las mismas imágenes que la versión original, pero reducidas a un tamaño de 1024 x 1024 píxeles.

Tabla 3.3: DB MIAS

2. Fase de preparación de los datos:

- En esta parte del proceso KDD se utilizó el método de Mejía, descrito en el apartado 2.3 para la limpieza de los datos. Este método está basado en Nonsubsampled Contourlet Transform NSCT y el filtro de Prewitt.
- Tal como se describe en (Mejía et al., 2009) este método de filtrado es usado satisfactoriamente para resaltar anomalías en mastografías digitales. En la Figura 3.1 se muestra una imagen antes y después de aplicar el proceso de filtrado. En la imagen filtrada se puede apreciar cómo se resaltan los píxeles más brillantes que revelan la existencia de la anomalía.

¹Para ver una descripción detallada ir al enlace <http://peipa.essex.ac.uk/info/mias.html>

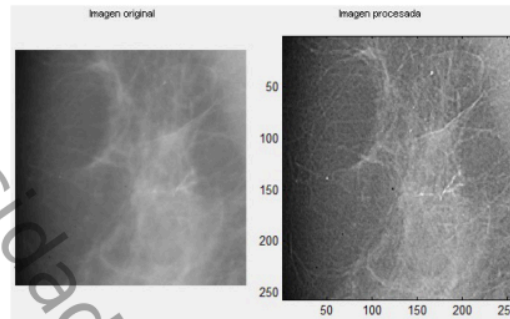


Figura 3.1: Filtrado. Fuente: (Sánchez, 2010)

- Teniendo la imagen filtrada se procede a segmentar la zona de interés como se muestra a continuación:

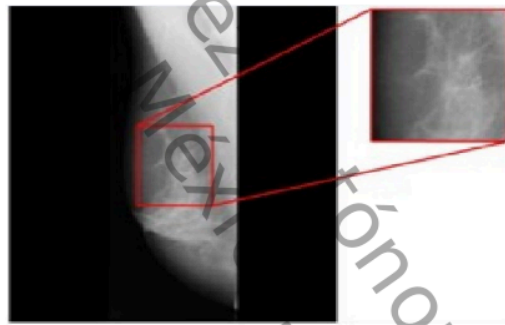


Figura 3.2: Segmentación de la zona de interés. Fuente: elaboración propia

- Finalmente, en esta fase del proceso KDD, se normaliza la zona de interés de cada imagen con la finalidad que todas tengan las mismas dimensiones.

3. Fase de transformación de los datos:

- En esta tercera fase del proceso KDD, se obtiene el histograma de cada imagen normalizada. Los histogramas describen la frecuencia de distribución de los niveles de gris. El eje x representa los niveles de gris que va desde 0 a 255, el eje y representa el número de píxeles que corresponden a cada nivel de gris. Un ejemplo de este tipo de histograma es mostrado en la Figura 3.3

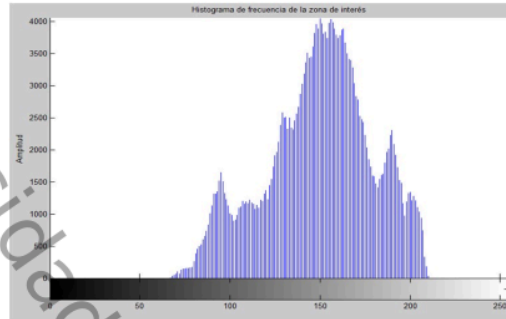


Figura 3.3: Histograma de la zona segmentada y normalizada. Fuente: Elaboración propia.

- Para construir la base de entrenamiento se tomaron 100 instancias (histogramas), 50 con cáncer y 50 sin cáncer. De estas 100 instancias 15 son benignas (B), 35 son normales (N), sin cáncer y 50 son malignas (M). Finalmente, en la fase de transformación de los datos, se construyó un archivo con la base de entrenamiento en formato CSV. Los archivos CSV (*comma-separated values*) son un tipo de documento en formato abierto sencillo para representar datos en forma de tabla, en las que las columnas se separan por comas y las filas por saltos de línea. Un ejemplo de una fila de este tipo de archivo se muestra a continuación:

```
520417,26413,16747,14412,11561,9346,10517,7801,5836,
5303,4872,4381,4115,3718,3575,3153,2767,2609,2424,
2350,2235,2126,2226, ..., B
```

4. Fase de minería de datos:

- Para realizar la detección de patrones se usó el software WEKA (*Waikato Environment for Knowledge Analysis*) v. 3.6.1 (University Waikato, 2011). Weka contiene diferentes métodos de clasificación (ver apartado 3.1.1).
- Se hicieron pruebas con cada clasificador de cada método y el mejor resultado se obtuvo con el clasificador *Decorate* (ver apartado 4.1). *Decorate* es un meta clasificador que se puede combinar con diferentes clasificadores base. En esta investigación se utilizó como clasificador base el árbol de decisión multiclase: LADTree. Este clasificador base utiliza el algoritmo LogitBoost, el algoritmo se muestra a continuación:

Algoritmo 1: LogitBoost

Result: If $p(1|a) > 0,5$ predecir la primera clase **Else** la segunda
begin
 for $j = 1$ hasta t **do**
 for $a[i]$ **do**
 Asignar el valor objetivo para la regresión a
 $z[i] = (y[i] - p(1|a[i])) / (p(1|a[i]) * (1 - p(1|a[i])))$
 Asignar el peso de la instancia a
 $[i] = p(1|a[i]) * (1 - p(1|a[i]))$
 Ajustar un modelo de regresión f_j a los datos con valores
 de clase $z[i]$ y pesos $w[i]$

5. Fase de análisis de los datos obtenidos:

- Esta fase se describe en el apartado [4.2](#)

3.1. Análisis de requerimientos

En este apartado se describe el análisis de requerimientos realizado para la elaboración del prototipo del sistema y se explica el porqué de la utilización de las tecnologías.

3.1.1. Requerimientos

Este prototipo de sistema ayuda al especialista (radiólogo) a tener una segunda opinión en la detección de anomalías cancerígenas en una mastografía digital.

A continuación se mencionan los requerimientos necesarios para un correcto manejo y funcionamiento del prototipo.

Requerimientos en la interacción humano-computadora

- Diseñar interfaces amigables y de fácil manejo para el especialista.
- Mantener una combinación de uso intuitivo entre las interfaces.
- Colocar los botones de forma ergonómica para el uso del usuario.
- Procurar presentar imágenes no distorsionadas o difíciles de interpretar.
- Presentar mensajes de error o de información que sean específicos y entendibles.

Requerimientos funcionales del sistema

- Contar con al menos 512 MB de memoria RAM para el proceso de tratamiento de imágenes.
- Contar con el programa para minería de datos Weka 3.6.4.
- Contar con el entorno de programación Matlab R2009a.
- Contar con el Bloc de notas para visualizar resultados.

Herramienta de minería de datos Weka

Weka es una colección de clasificadores para tareas de MD. Weka también contiene herramientas para pre-procesamiento de datos, clasificación, regresión, agrupamiento, reglas de asociación y visualización. Weka es un software de uso libre bajo la licencia pública GNU (University Waikato, 2011).

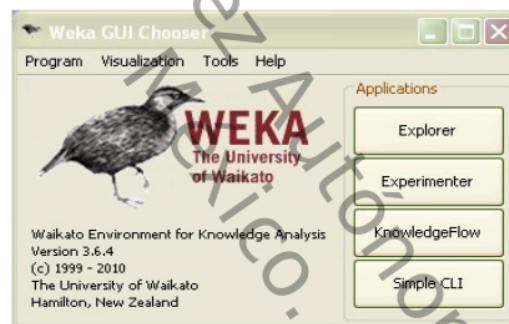


Figura 3.4: Herramienta de MD Weka. Fuente: (University Waikato, 2011)

Weka contiene diferentes métodos de clasificación, tales como: método bayesiano (*bayes*), método de funciones de clasificación (*functions*), método perezosos (*lazy*), metaclasificadores (*meta*), método de generación de reglas (*rules*), método de generación de árboles de decisión, y método varios (*misc*). Estos métodos fueron descritas en el apartado 2.1.5

3.2. Semi-automatización del proceso KDD

- En base al *modelado de gestión* del modelo DRA, el flujo de información para obtener la detección de la anomalía en un mastografía por parte del radiólogo, se representa con el siguiente *rich picture*² (Figura 3.5):

²Rich Picture: término inglés que define la forma de representar una idea, un problema o un proceso mediante la utilización de imágenes o dibujos.



Figura 3.5: Flujo de información. Fuente: elaboración propia.

- Siguiendo el modelo DRA, se realizó el *modelado de datos* en el cual se refina el modelado de gestión como una tabla de datos necesaria para que el especialista tenga una segunda opinión del análisis de la mastografía y con ello aumente la fiabilidad de la detección de la anormalidad, un ejemplo de esta tabla se muestra a continuación:

```

=== Predictions on test data ===

inst#    actual    predicted error prediction (PIX4)
1        1:B      2:N      +    0.5 (0)
2        2:N      2:N      +    0.5 (0)
3        3:M      2:N      +    0.5 (0)
4        1:B      3:M      +    0.639 (0)
  
```

Figura 3.6: Modelado de datos. Fuente: (Sánchez, 2010)

En esta tabla vemos cuatro instancias, las primeras tres son de uso necesario para el correcto funcionamiento del prototipo; ahora bien, la nueva muestra, que siempre será la instancia cuatro (inst# 4), es la que interesa al especialista, en esta muestra el especialista predijo que la zona que él consideró sospechosa representaba una anormalidad de primer tipo, es decir, de

CAPÍTULO 3. DESARROLLO DE LA INVESTIGACIÓN

tipo benigna (1:B) (círculo azul), sin embargo, el clasificador indica que esa normalidad puede ser de tipo tres, o sea, maligna (3:M) (círculo rojo). Esto ayuda al especialista pues ahora requiere tomar otra mastografía desde otro ángulo para corroborar si la anormalidad es realmente de tipo 3:M.

- Como el *modelado de generación de aplicaciones* del modelo DRA recomienda crear módulos e integrar módulos útiles para el fin que se desea, en la presente investigación se reutiliza un módulo para el proceso de filtrado y se crean tres más, uno para la pantalla de inicio, otro para la segmentación y otro para la clasificación. A continuación se presentan y se describen:

1.- Módulo «Pantalla de inicio» :

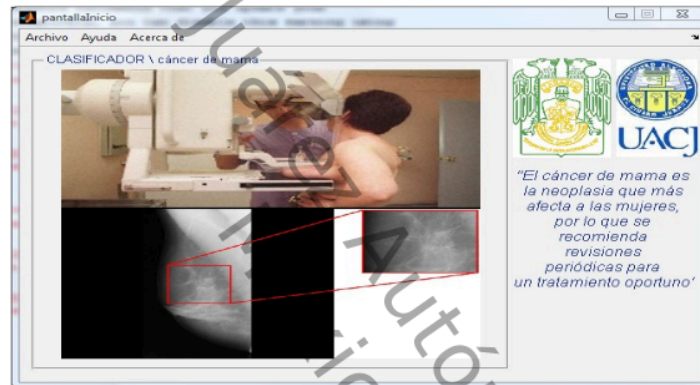


Figura 3.7: Módulo «Pantalla inicio». Fuente: (Sánchez, 2010).

En este módulo el usuario puede seleccionar las siguientes opciones del menú:

Archivo	Ayuda	Acerca de
↳ Filtrar	↳ Manual	
↳ Segmentar		
↳ Clasificar		

Figura 3.8: Menú de opciones. Fuente: (Sánchez, 2010).

2.- Módulo «Filtrar» :

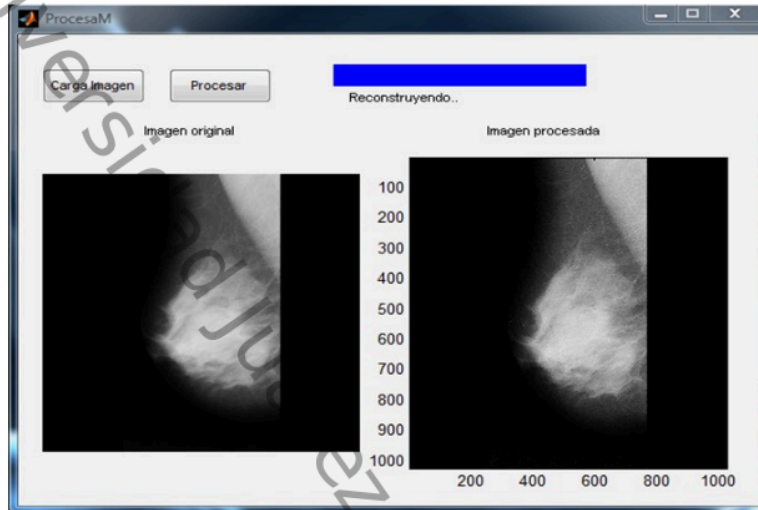


Figura 3.9: Módulo «Filtrar». Fuente: (Sánchez, 2010).

Este módulo se integró al prototipo ya que con este método de filtrado se obtienen excelentes resultados en el realce de anomalías, este trabajo se presentó en la *8th Mexican International Conference on Artificial Intelligence* (MICA I 2009). El método funciona usando the Nonsubsampled Contourlet Transform (NSCT) y el filtro de Prewitt. El método se basa en el enfoque clásico utilizado en los métodos de transformación para procesamiento de imágenes (Mejía et al., 2009). El diagrama del método se muestra en la Figura 2.4.

3.- Módulo «Segmentar»:

En este módulo el especialista segmenta la zona que él considera presenta anomalía, tal como se ve en la Figura 3.10.

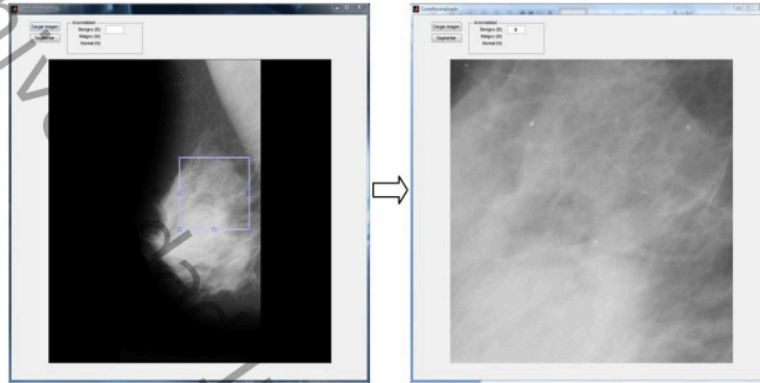


Figura 3.10: Módulo «Segmentar». Fuente: (Sánchez, 2010).

Para la segmentación se emplea el siguiente código:

```
segment = imcrop(handles.myImage);
```

en la variable `segment` se almacena la zona segmentada. Luego, esa zona de interés se normaliza para que todas las imágenes tengan las mismas dimensiones y se almacena en la variable `norm`. El código usado para la normalización es como sigue:

```
norm = imresize (segment, [512 512]);
```

Después de ser normalizada la zona de interés, se obtiene su histograma de frecuencia de niveles de gris, que va de 0 a 255 (ver Figura 3.3). Este histograma se almacena en una hoja de cálculo llamada `Histograma.xlsx`, esto para propósito de construir la base de testeo. El código usado es el siguiente:

```
xlsx = imhist (norm)
xlswrite('Histograma.xlsx',xlsx,'Hoja1','A5:IV5, A5:IV5')
```

Luego se pregunta al radiólogo si la anomalía que detectó es de tipo benigno (B), maligno (M) o normal (N).

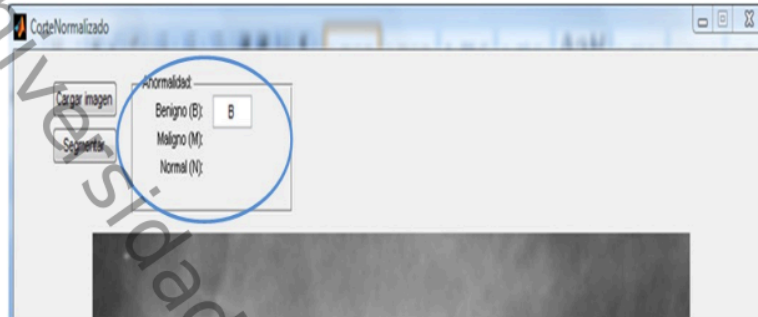


Figura 3.11: Tipo de anormalidad. Fuente: (Sánchez, 2010).

Y según sea el tipo se almacena en la tabla con el siguiente código:

```
anormB = ['B']; % indica que la anormalidad es "benigna"
anormN = ['N']; % indica que la anormalidad es "normal"
anormM = ['M']; % indica que la anormalidad es "maligna"

val = get(hObject,'String'); % Se obtiene valor de la caja de texto

if (val == anormB) % Se compara el valor obtenido con el tipo de
    anormalidad
        xlswrite('Histograma.xlsx', anormB, 'Hoja1', 'IW5'); % Se
            inserta "B"
        en la celda [256,6] de la tabla "Histogramas"
    else if (val == anormN) % Se compara el valor introducido con
        el rango
        xlswrite('Histograma.xlsx', anormN, 'Hoja1', 'IW5');
        % Se inserta "SI"
        en la celda [256,6] de la tabla "Histogramas"
    else if (val == anormM) % Se compara el valor introducido
        con el rango
        xlswrite('Histograma.xlsx', anormM, 'Hoja1', 'IW5');
        % Se inserta
        "SI" en la celda [256,6] de la tabla "Histogramas"
    end
end
end
```

Finalmente, para este módulo se recomienda abrir el archivo `Histograma.xlsx` y guardarlo en formato CSV (delimitado por comas), para su posterior uso.

4.- Módulo «Clasificar»:

En este módulo se selecciona el archivo `Histograma.csv` para implementar la clasificación o detección de la anomalía (ver Figura 3.12):

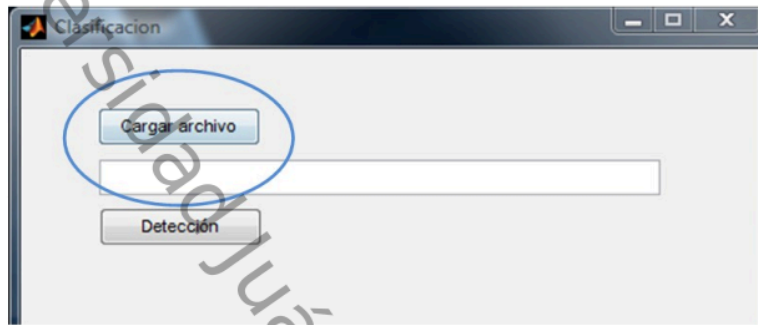


Figura 3.12: Módulo «Clasificar». Fuente: (Sánchez, 2010).

La clasificación se activa pulsando el botón «Detección», el que indica la llamada al clasificador *Decorate*, el cual se combina, como se explicó en la fase 4 con el clasificador LADTree y este, a su vez, utiliza el algoritmo base LoogitBoost (ver algoritmo 1).

```
!java -classpath weka.jar weka.classifiers.meta.Decorate -W
weka.classifiers.trees.LADTree -t Htraining.csv -T
Histograma.csv -p 5 > detect.txt
```

Finalmente se lanza el resultado³ para análisis del especialista. El código es el siguiente:

```
winopen ('detect.txt')
```

- Como el proceso DRA enfatiza la reutilización y la constante prueba de los componentes, para esta última fase de **Pruebas de entrega**, ya se ha probado el componente o módulo de filtrado, los módulos de segmentación y clasificación se depurarán siguiendo las observaciones del especialista.

³Este resultado fue explicado en el *modelado de datos* del modelo DRA

Capítulo 4

Pruebas y resultados

Es importante aclarar que a) para construir la base de entrenamiento se usó un conjunto de datos de 100 instancias; luego, b) para elegir el clasificador que presentara mejor desempeño se utilizó un conjunto de datos de 20 instancias; finalmente c) para realizar los experimentos con el clasificador elegido se utilizaron cuatro conjuntos de datos de 20 instancias cada conjunto, 80 en total para los experimentos. Cada conjunto de dato fue obtenido de manera aleatoria de la base de datos MIAS.

4.1. Algoritmos de clasificación para minería de datos

Se probó cada clasificador de cada método; los métodos son: método bayesiano (*bayes*), método de funciones de clasificación (*functions*), método perezosos (*lazy*), metaclassificadores (*meta*), método de generación de reglas (*rules*), método de generación de árboles de decisión, y método varios (*misc*). Estos métodos fueron descritos en el apartado 2.1.5

Estas pruebas registraron porcentajes variados, tal como se puede ver en la Tabla 4.1 y la Tabla 4.2. Cabe recalcar que estas pruebas se realizaron para seleccionar el clasificador que presentara mejor resultado. Las pruebas se hicieron con 20 instancias, que se tomaron aleatoriamente de la base de datos MIAS, contra las 100 instancias que se usaron para entrenar el clasificador, ya que de esta forma es como trabaja las instancias la herramienta de MD Weka.

¹Los detalles de cada prueba aparecen en el Anexo 1

CAPÍTULO 4. PRUEBAS Y RESULTADOS

Método	Clasificador	ICC*	%
Bayes	BayesNet	5	25 %
	ComplementNaiveBayes	12	60 %
	DMNBtext	7	35 %
	NaiveBayes	13	65 %
	NaiveBayesMultinomial	12	60 %
	NaiveBayesMultinomialUpdateable	12	60 %
	NaiveBayesUpdate	13	65 %
Functions	Logistic	14	70 %
	MultilayerPerceptron	13	65 %
	RBFNetwork	5	25 %
	SimpleLogistic	14	70 %
	SMO	13	65 %
Lazy	IB1	12	60 %
	IBk	12	60 %
	KStar	5	25 %
	LWL	11	55 %
Meta	AdaBoostM1	5	25 %
	AttributeSelectedClassifier	5	25 %
	Bagging	12	60 %
	ClassificationViaClustering	6	30 %
	ClassificationViaRegression	11	55 %
	CVParameterSelection	5	25 %
	Dagging	10	50 %
	Decorate	19	95 %
	END	12	60 %
	FilteredClassifier	5	25 %
	MultiBoostAB	5	25 %
	MultiClassClassifier	10	50 %
	MultiScheme	5	25 %
	OrdinalClassClassifier	12	60 %
	RacedIncrementalLogitBoost	5	25 %
	RandomCommittee	15	75 %
	RandomSubSpace	5	25 %
	RotationForest	11	55 %
	Stacking	5	25 %
	StackingC	5	25 %
	Vote	5	25 %

* ICC ← Instancias Correctamente Clasificadas

Tabla 4.1: Pruebas / Clasificadores (1)

CAPÍTULO 4. PRUEBAS Y RESULTADOS

Método	Clasificador	ICC*	%
Misc	HyperPipes	10	50 %
	VFI	5	25 %
Rules	ConjunctiveRule	5	25 %
	DecisionTable	5	25 %
	DTNB	5	25 %
	JRip	6	30 %
	NNge	9	45 %
	OneR	11	55 %
	PART	12	60 %
	Ridor	9	45 %
	ZeroR	5	25 %
Trees	BFTree	5	25 %
	DecisionStump	5	25 %
	FT	12	60 %
	J48	11	55 %
	J48graft	11	55 %
	LADTree	10	50 %
	LMT	14	70 %
	NBTree	11	55 %
	RandomForest	17	85 %
	RandomTree	11	55 %
	REPTree	5	25 %
	SimpleCart	5	25 %
	UserClassifier	5	25 %

* ICC ← Instancias Correctamente Clasificadas

Tabla 4.2: Pruebas / Clasificadores (2)

4.1.1. Selección del clasificador

De las pruebas presentadas en el apartado anterior, Los porcentajes más altos se obtuvieron con los clasificadores *Decorate*, *RandomCommitee* y *RandomForest* (ver Tabla 4.3). Y de estos tres clasificadores el mejor resultado se obtiene con *Decorate*. Con este clasificador se obtuvo 95 % de instancias correctamente clasificadas.

Método	Clasificador	ICC*	%
Meta	Decorate	19	95 %
Meta	RandomCommitee	15	75 %
Trees	RandomForest	17	85 %

* ICC $\leftarrow$ Instancias Correctamente Clasificadas

Tabla 4.3: Porcentajes más altos

4.2. Experimentos y análisis

Los experimentos, utilizando el clasificador seleccionado (*Decorate*), se realizaron utilizando cuatro conjuntos de datos de veinte instancias cada uno. Las tablas 4.4, 4.5, 4.6 y 4.7 muestran los resultados.

CAPÍTULO 4. PRUEBAS Y RESULTADOS

inst#	actual	predicted	error
1	1:B	1:B	
2	1:B	1:B	
3	2:N	2:N	
4	2:N	2:N	
5	3:M	3:M	
6	2:N	2:N	
7	2:N	2:N	
8	1:B	1:B	
9	2:N	2:N	
10	3:M	3:M	
11	2:N	2:N	
12	3:M	3:M	
13	2:N	2:N	
14	1:B	1:B	
15	2:N	2:N	
16	3:M	3:M	
17	3:M	3:M	
18	2:N	2:N	
19	1:B	1:B	
20	2:N	1:B	+
Correctly Classified 19 < -- 95 % Incorrectly Classified 1 < -- 5 %			

En el primer conjunto de datos tenemos 20 instancias, de las cuales 5 son benignas (B), 10 normales (N), y 5 malignas (M). Se obtuvo 95% de instancias correctamente clasificadas, es decir, 19 de 20 instancias.

Tabla 4.4: Experimento con el primer conjunto de datos

CAPÍTULO 4. PRUEBAS Y RESULTADOS

inst#	actual	predicted	error
1	2:N	2:N	+
2	3:M	3:M	
3	3:M	3:M	
4	2:N	2:N	
5	1:B	1:B	
6	2:N	1:B	
7	1:B	1:B	
8	1:B	1:B	
9	2:N	2:N	
10	2:N	2:N	
11	3:M	3:M	
12	2:N	2:N	
13	2:N	2:N	
14	1:B	1:B	
15	2:N	2:N	
16	3:M	3:M	
17	2:N	2:N	
18	3:M	3:M	
19	1:B	1:B	
20	1:B	1:B	
Correctly Classified 19 < -- 95 %			
Incorrectly Classified 1 < -- 5 %			

En el segundo conjunto de datos tenemos 20 instancias, de las cuales 6 son benignas (B), 9 normales (N), y 5 malignas (M). Se obtuvo 95% de instancias correctamente clasificadas, es decir, 19 de 20 instancias.

Tabla 4.5: Experimento con el segundo conjunto de datos

CAPÍTULO 4. PRUEBAS Y RESULTADOS

inst#	actual	predicted	error
1	1:B	2:N	+
2	2:N	2:N	
3	3:M	3:M	
4	2:N	2:N	
5	2:N	2:N	
6	2:N	2:N	
7	3:M	3:M	
8	2:N	2:N	
9	1:B	3:M	+
10	2:N	2:N	
11	3:M	3:M	
12	3:M	3:M	
13	2:N	2:N	
14	1:B	1:B	
15	1:B	1:B	
16	3:M	3:M	
17	2:N	2:N	
18	3:M	3:M	
19	1:B	1:B	
20	1:B	1:B	
Correctly Classified 18 < -- 90 %			
Incorrectly Classified 2 < -- 10 %			

En el tercer conjunto de datos tenemos 20 instancias, de las cuales 6 son benignas (B), 8 normales (N), y 6 malignas (M). Se obtuvo 90% de instancias correctamente clasificadas, es decir, 18 de 20 instancias.

Tabla 4.6: Experimento con el tercer conjunto de datos

CAPÍTULO 4. PRUEBAS Y RESULTADOS

inst#	actual	predicted	error
1	1:B	1:B	
2	2:N	2:N	
3	3:M	3:M	
4	3:M	3:M	
5	3:M	3:M	
6	2:N	2:N	
7	2:N	2:N	
8	3:M	3:M	
9	2:N	2:N	
10	2:N	2:N	
11	2:N	2:N	
12	2:N	2:N	
13	3:M	3:M	
14	2:N	2:N	
15	2:N	2:N	
16	2:N	2:N	
17	2:N	2:N	
18	2:N	2:N	
19	2:N	2:N	
20	1:B	1:B	
Correctly Classified 20 < 100 %			
Incorrectly Classified 0 < 0 %			

Finalmente, en el cuarto conjunto de datos tenemos 20 instancias, de las cuales 2 son benignas (B), 13 normales (N), y 5 malignas (M). Se obtuvo 100 % de instancias correctamente clasificadas, es decir, 20 de 20 instancias.

Tabla 4.7: Experimento con el cuarto conjunto de datos

A continuación se muestra de manera gráfica en la Figura 4.1 lo expuesto anteriormente:

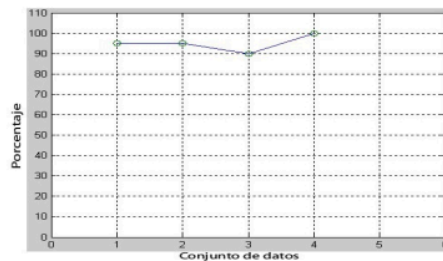


Figura 4.1: Gráfica de resultados de los datasets. Fuente: elaboración propia.

En el primer conjunto de datos se obtiene 95%; en el segundo conjunto de datos se obtiene 95%; en el tercero se obtiene 90%, y en el cuarto se obtiene 100 %. Estos resultados fluctúan entre 90 % y 100 %, dando un promedio de 95 % de instancias correctamente clasificadas. Esto muestra que el clasificador *Decorate*, usado en el proceso KDD, es muy fiable en la detección de anomalías cancerígenas. Esto se corrobora, ya que tenemos cuatro instancias incorrectamente clasificadas (ver Tabla 4.8). 2 instancias

3 en benignas (B) y 2 son normales (N).

Las instancias malignas (M) indican que sí existe anomalía. En la Tabla 4.8 3 demos ver que ninguna instancia maligna es incorrectamente clasificada, lo cual implica que el clasificador realiza un correcto diagnóstico de todas las instancias malignas, tal como se muestra en la Tabla 4.9. Y la razón por la que las instancias malignas nos interesan es porque dichas instancias representan una anomalía cancerígena.

dataset	inst#	actual	predicted	error
1	20	2:N	1:B	+
2	6	2:N	1:B	+
3	1	1:B	2:N	+
3	9	1:B	3:M	+

Tabla 4.8: Instancias incorrectamente clasificadas

dataset	inst#	actual	predicted	error
1	5	3:M	3:M	
1	10	3:M	3:M	
1	12	3:M	3:M	
1	16	3:M	3:M	
1	17	3:M	3:M	
2	2	3:M	3:M	
2	3	3:M	3:M	
2	11	3:M	3:M	
2	16	3:M	3:M	
2	18	3:M	3:M	
3	3	3:M	3:M	
3	7	3:M	3:M	
3	11	3:M	3:M	
3	12	3:M	3:M	
3	16	3:M	3:M	
3	18	3:M	3:M	
4	3	3:M	3:M	
4	4	3:M	3:M	
4	5	3:M	3:M	
4	8	3:M	3:M	
4	13	3:M	3:M	

Tabla 4.9: Clasificación de instancias M

4.3. Evaluación

En los últimos años se han presentado varias aportaciones para ayudar al radiólogo en el diagnóstico de cáncer de mama. Las pruebas en estos trabajos reportados se han realizado con diferentes DB. En la Tabla 4.10, vemos que (Lizcano et al., 2005) obtuvo 95.35 %, sin embargo, la DB nos dice que estos resultados son propios para la comunidad española y, además, el tamaño de la muestra es pequeño. (Quintero et al., 2008) obtuvo 91 % utilizando la misma DB que se usa en esta investigación, sin embargo, el tamaño de la muestra es muy pequeño. En (Calot et al., 2008) se obtuvo 73 % utilizando una DB diferente, resultados realmente bajos y, además, no se menciona el tamaño de la muestra. (Mejía et al., 2009), utilizaron una muestra pequeña y los resultados no los presenta en porcentaje, dadas las métricas de medición usadas. Finalmente, en esta investigación, se obtiene 95 %, resultados satisfactorios, teniendo en cuenta que la DB utilizada goza de amplia aprobación por la comunidad científica (Rangayyan et al., 1997), (Antonie et al., 2001), (Farag y Mashali, 2005), (Quintero

CAPÍTULO 4. PRUEBAS Y RESULTADOS

et al., 2008), (Mejía et al., 2009), (Heath et al., 2010), y que el tamaño de la muestra es absolutamente considerable si se compara con los tamaños de las muestras de las investigaciones antes mencionadas.

Método	año	DB utilizada	Tamaño muestra	% de ICC*
SDAC	2011	MIAS	80	95 %
(Mejía et al., 2009)	2009	MIAS	12	
(Calot et al., 2008)	2008	DDSM	s/n	73 %
		(Heath et al., 2010)		
(Quintero et al., 2008)	2008	MIAS	30	91 %
(Lizcano et al., 2005)	2005	Hospital Puerta de Hierro de Madrid	43	95.35 %

* ICC ← Instancias Correctamente Clasificadas

Tabla 4.10: Comparación de resultados obtenidos con otros trabajos previamente reportados en la literatura.

Capítulo 5

Conclusiones y trabajos futuros

Conclusiones

En esta investigación se combinó el método de filtrado de (Mejía et al., 2009) y el clasificador *Decorate* en el proceso KDD para detectar anomalías en mastografías digitales. Los resultados (ver apartado 4) muestran una detección satisfactoria usando la base de datos MIAS (Tabla 3.3).

Cada objetivo planteado fue alcanzado satisfactoriamente, ya que:

1. Se obtuvo la base de datos de imágenes de mastografía de acceso público MIAS (Tabla 3.3).
2. Se aplicó el método de filtrado de (Mejía et al., 2009) para tratamiento de mejora de los datos (imágenes).
3. Se transformaron correctamente los datos (imágenes) a formato entidad relación, en la fase 3 del proceso KDD, para construir una base de conocimiento.
4. Se desarrolló un prototipo de sistema (ver apartado 3.2) que aplica DM para la detección de anomalías cancerígenas. Esta fase de la investigación se concluyó en el Instituto de Ingeniería y Tecnología de la Universidad Autónoma de Ciudad Juárez IIT-UACJ (Sánchez, 2010).
5. Se evaluó la detección del clasificador comparándola con el estado del arte (ver apartado 4.3).

Dado todo el trabajo anterior, se logró alcanzar el objetivo general, el cual se cita de manera textual y dice:

CAPÍTULO 5. CONCLUSIONES Y TRABAJOS FUTUROS

Aplicar el proceso KDD para reconocer anomalías en imágenes de mastografías con la finalidad de disminuir los falsos negativos.

Los falsos negativos son las anomalías que sí existen y que no pueden ser detectadas a simple vista por el radiólogo. El objetivo general quedó comprobado, ya que como puede verse en la Tabla 4.9 todas las instancias malignas fueron correctamente clasificadas. Y con esto, la hipótesis, que dice que:

Al usar el proceso KDD en mastografías digitales se puede reducir de 25 % a 10 % el porcentaje de dar un falso negativo, y con ello obtener un porcentaje de 90 % en la detección de anomalías cancerígenas.

La hipótesis quedó comprobada, usando la base de datos MIAS, ya que como se explica en el apartado 4.2 el promedio de las pruebas es de 95 %. Es decir, 5 % mayor de los planteado en la hipótesis.

Los resultados de esta investigación son altamente esperanzadores, ya que su impacto se traduce en que a una persona se le puede detectar a tiempo alguna anomalía cancerígena y recibir tratamiento oportuno. Además, que se puede evitar una metástasis, y con ello mejorar la calidad de vida e incluso salvar la vida de una persona con cáncer de mama potencial.

Los resultados obtenidos han sido presentados en la *Seventh Latin American Workshop on Non-Monotonic Reasoning* (LANMR 2011) (Sánchez et al., 2011); en la *9na. Mexican International Conference on Artificial Intelligence* (MICA I 2010) (Sánchez et al., 2010a); en el *7mo. Congreso Nacional y 4to. Internacional de Informática y Sistemas* (CONAIS 2010) (Sánchez et al., 2010b); en la *Semana de divulgación y video científico* (SDVC 2010) y en la *ExpoCiencias Veracruz-Tabasco 2010*.

Trabajos futuros

El trabajo de investigación realizado en esta tesis abre nuevas líneas de investigación futuras, ya que existen puntos que ampliar y mejorar asociados a los resultados presentados. Como trabajos futuros se propone:

- Usar otra base de datos de mastografías digitales para verificar que se mantenga el promedio de fiabilidad alrededor de 95 %, aplicando el proceso KDD. De preferencia poder obtener una base de datos de la región.
- Utilizar otro método de filtrado de datos (que pudiera surgir y mejorar al aquí empleado) con el cual se obtengan imágenes más nítidas y en

consecuencia un mejor resultado al ya obtenido.

- Automatizar completamente el proceso KDD, es decir, que el prototipo propuesto pase de ser semi-automático a automático.
- Utilizar otro entorno de programación que pudiera ayudar a desarrollar interfaces más amigables e intuitivas.
- Además, teniendo en cuenta los buenos resultados aquí obtenidos e inspirados en (Gallegos-Funes et al., 2009), quienes primero trabajaron con mastografías digitales y luego con imágenes microscópicas de papanicolaou, se pretende migrar el proceso KDD para ayudar en la detección del cáncer cervicouterino.

Bibliografía

- Antoni, M., Zaiane, O., y Coman, A. (2001). Application of data mining techniques for medical image classification. *In Proc. Of Second Intl. Workshop on Multimedia Data Mining (MDM/KDD 2001) in conjunction with Seventh ACM SIGKDD*, pages 94–101.
- Azouzi, S. H. y Vilaplana, V. (2006). Editor de imágenes basado en regiones. Aplicación en entorno Matlab. *Escuela Universitaria de Ingeniería Técnica Industrial de Terrasa*, page 122.
- Bandyopadhyay, S., Maulik, U., Holder, L. B., y Cook, D. J. (2005). Advanced methods for knowledge discovery from complex data. *Springer*, pages 3–42.
- Calot, E., Merlino, H., y García-Martínez, P. B. R. (2008). Clasificación de tumores en mamografías mediante uso combinado de RBP y filtros sobel. *Laboratorio de Sistemas Inteligentes. Facultad de Ingeniería. UBA*.
- Cheng, H., Shi, X., Min, R., Hu, L., Cai, X., y Du, H. (2007). Approaches for automated detection and classification of masses in mammograms. *Revista Médica del Instituto Mexicano del Seguro Social*, 45:157–166.
- Coto, E. (2003). Métodos de segmentación de imágenes médicas. *Lecturas en ciencias de la computación*, pages 9–16.
- Delrieux, C. (2007). Las ciencias de las imágenes en argentina. *Departamento de Ing. Eléctrica y Computadoras. Universidad Nacional del Sur*.
- Elizondo, J. J. E. (2002). Fundamentos de procesamiento de imágenes. *Instituto Tecnológico de Ciudad Madero*, pages 10–13.
- Farag, A. y Mashali, S. (2005). DCT based features for the detection of microcalcifications in digital mammograms. *Micro-NanoMechatronics and Human Science, 2003 IEEE International Symposium*, 1(8823932):352–355.
- Fernández, E. (2009). Análisis de clasificador bayesiano. *Laboratorio de sistemas inteligentes* www.fi.uba.ar/laboratorios/lsi.

- Gallegos-Funes, F., Gómez-Mayorga, M., Lopez-Bonilla, J., y Cruz-Santiago, R. (2009). Rank M-Type Radial Basis Function (RMRBF) Neural Network for Pap Smear Microscopic Image Classification. *C. Roy Keys Inc.* <http://redshift.vif.com> 16:4.
- González, A. J. (2007). Extracción de características de imágenes de células de médula ósea para la clasificación de leucemias agudas. *Instituto Nacional de Astrofísica, óptica y Electrónica*.
- Heath, M., Bowyer, K., Kopans, D., Moore, R., y Kegelmeyer, W. (2010). The Digital Database for Screening Mammography (DDSM). *Medical Physics Publishing. ISBN: 1-930524-00-5*, pages 212–218. University of South Florida Digital Mammography Home Page <http://marathon.csee.usf.edu/Mammography/Database.html>
- Larranaga, P. y Inza, I. (2007). Introducción a la minería de datos. *Intelligent Systems Group. Universidad del País Vasco*.
- Lizcano, J. A. M., Pérez, L. M., y Ávila, C. S. (2005). Sistema para la detección automatizada de microcalcificaciones en mamografía digitalizada utilizando la transformada contourlet. *Congreso de Métodos Numéricos en Ingeniería. ISBN: 978-607-7557-71-5*, pages 2–11.
- Martínez, A. V. (2003). Minería de datos. Una introducción. *Conciencia tecnológica. Instituto Tecnológico de Aguascalientes*.
- Mejía, J. M. (2010). Realce de microcalcificaciones en mamografías digitales utilizando la Transformada Contourlet. *Universidad Autónoma de Ciudad Juárez*.
- Mejía, J. M., de Jesús Ochoa Domínguez, H., Vergara-Villegas, O. O., Sánchez, V. G. C., y Maynez, L. O. (2009). The Nonsampled Contourlet Transform for Enhancement of Microcalcifications in Digital Mammograms. *Advances in Artificial Intelligence*, 5845:292–302.
- Méndez, N. D. D., Chavarro, J. C. C. P., y Laverde, R. M. (2007). Seguridad inteligente. *Science Et Technica*, 13:389–394.
- Ng, R. T. y Pei, J. (2007). Special issue on data mining for health informatics. *ACM Digital library*, 9(1).
- Olmos, I., González, J. A., y Osorio, M. (2005). Mining common patterns on graphs. *Springer*, pages 41–48.
- Olmos-Pineda, I. y González-Bernal, J. (2007). Minería de datos. *Semana de Informática. Instituto Tecnológico de Puebla*.
- Pajares-Martinsanz, G. (2005). Visión por computador. Imágenes digitales y aplicaciones. *Alfaomega*.

- Pressman, R. (2006). Ingeniería de software. un enfoque práctico. *McGraw-Hill*.
- Quintero, L. V., Jiménez, L. B., Torres, C., y Mattos, L. (2008). Transformada wavelet y técnicas de filtrado no lineal aplicadas a la detección de microcalcificaciones en mamografías digitales. *Memorias. XIII Simposio de Tratamiento de Señales, Imágenes y Visión Artificial STSIVA*. ISSN 978-958-8477-00-8, II:23-26.
- Ramos, J. L. G., Jiménez, H. A., Osorio, C. V., y Martínez, M. A. T. (2007). Descubrimiento del conocimiento en base de datos utilizando herramienta de software libre, caso: DAIS. *Avances en Informática y Sistemas Computacionales*, 2:23-51.
- Rangayyan, R., El-Faramawy, N., Desautels, L., y Alim, O. (1997). Measures of Acutance and Shape for Classification of Breast Tumors. *IEEE Transactions on Medical Imaging*, 16:799.
- Saldana, J. F. R. y Flores, R. G. (2005). El proceso de descubrimiento de conocimiento en base de datos. *Ingenierías*, 8(26):37-47.
- Sánchez, E. (2010). Prototipo de software para detectar anomalías cancerígenas en mastografías digitales. *Reporte del proyecto de investigación*, page 9. Instituto de Ingeniería y Tecnología de la Universidad Autónoma de Ciudad Juárez (IIT-UACJ).
- Sánchez, E., Alpuín-Jiménez, H., de Jesús Ochoa Domínguez, H., y Pozos-Parra, P. (2011). SDCA: System to Detect Cancerous Abnormalities. *Proceedings of the Seventh Latin American Workshop on Non-Monotonic Reasoning 2011*. ISSN 1613-0073, 804:115-122.
- Sánchez, E., Pozos-Parra, P., y Alpuín-Jiménez, H. (2010a). Cancer detection using the KDD process. *Advances in Soft Computing Algorithms*. ISSN: 1870-4079, 49:109-117.
- Sánchez, E., Pozos-Parra, P., y Alpuín-Jiménez, H. (2010b). Detección de cáncer de mama usando el proceso KDD en mastografías digitales. *Avances en Informática y Sistemas Computacionales*. ISBN: 978-607-7557-71-5, V:40-51.
- Sánchez, E. y Torres, J. P. (2009). SIA en el diagnóstico del cáncer. *Avances en Informática y Sistemas Computacionales*. ISBN: 978-607-7557-24-1, IV:28-40.
- Sánchez-Domínguez, H. y Paz-Morales, M. (2006). Prevalencias de factores de riesgo para cáncer de mama en población urbana y rural. *Salud en Tabasco*, 12(2):410-414.

Torres-Arreola, L. y Doubova, S. V. (2006). Approaches for automated detection and classification of masses in mammograms. *Pattern Recognition*, 39:264–668.

University Waikato (2011). Weka 3: Data Mining Software in Java. *Machine Learning Group at University of Waikato*. <http://www.cs.waikato.ac.nz/~ml/weka/index.html>.

Valencia, C. (2004). Evaluación de algoritmos basados en lógica difusa aplicados al preproceso y detección de bordes en imágenes digitales. *Springer*.

Witten, I. H. y Frank, E. (2005). *Data Mining. Practical Machine Learning Tools and Techniques*. Morgan Kaufmann Publishers. ISBN: 0-12-088407-0.

Anexo 1

Pruebas de selección del algoritmo de DM

Las pruebas fueron realizadas con una base de testeo de 20 instancias, de las cuales 5 son benignas (B), 10 normales (N), y 5 malignas (M). Se probó con cada clasificador de cada método. Los resultados se muestran a continuación:

Método bayesiano (*bayes*)

BayesNet

inst#	actual	predicted	error
1	1:B	3:M	+
2	1:B	3:M	+
3	2:N	3:M	+
4	2:N	3:M	+
5	3:M	3:M	
6	2:N	3:M	+
7	2:N	3:M	+
8	1:B	3:M	+
9	2:N	3:M	+
10	3:M	3:M	
11	2:N	3:M	+
12	3:M	3:M	
13	2:N	3:M	+
14	1:B	3:M	+
15	2:N	3:M	+
16	3:M	3:M	
17	3:M	3:M	
18	2:N	3:M	+
19	1:B	3:M	+
20	2:N	3:M	+
Correctly Classified 5 < -- 25 %			
Incorrectly Classified 15 < -- 75 %			

Con el clasificador *BayesNet* se obtuvo 25 % de instancias correctamente clasificadas, es decir, 5 de 20 instancias.

ComplementNaiveBayes

inst #	actual	predicted	error
1	1:B	2:N	+
2	1:B	2:N	+
3	2:N	2:N	
4	2:N	1:B	+
5	3:M	3:M	
6	2:N	2:N	
7	2:N	2:N	
8	1:B	2:N	+
9	2:N	1:B	+
10	3:M	1:B	+
11	2:N	2:N	
12	3:M	3:M	
13	2:N	2:N	
14	1:B	2:N	+
15	2:N	2:N	
16	3:M	3:M	
17	3:M	3:M	
18	2:N	2:N	
19	1:B	2:N	+
20	2:N	2:N	
Correctly Classified 12 < -- 60 %			
Incorrectly Classified 8 < -- 40 %			

Con el clasificador *ComplementNaiveBayes* se obtuvo 60 % de instancias correctamente clasificadas, es decir, 12 de 20 instancias.

DMNBtext

inst #	actual	predicted	error
1	1:B	2:N	+
2	1:B	3:M	+
3	2:N	2:N	
4	2:N	2:N	
5	3:M	2:N	+
6	2:N	3:M	+
7	2:N	3:M	+
8	1:B	3:M	+
9	2:N	3:M	+
10	3:M	3:M	
11	2:N	2:N	
12	3:M	2:N	+
13	2:N	3:M	+
14	1:B	3:M	+
15	2:N	2:N	
16	3:M	3:M	
17	3:M	1:B	+
18	2:N	3:M	+
19	1:B	3:M	+
20	2:N	2:N	
Correctly Classified 7 < -- 35 %			
Incorrectly Classified 13 < -- 65 %			

Con el clasificador *DMNBtext* se obtuvo 35 % de instancias correctamente clasificadas, es decir, 7 de 20 instancias.

NaiveBayes

inst #	actual	predicted	error
1	1:B	1:B	
2	1:B	2:N	+
3	2:N	2:N	
4	2:N	2:N	
5	3:M	3:M	
6	2:N	2:N	
7	2:N	3:M	+
8	1:B	2:N	+
9	2:N	1:B	+
10	3:M	2:N	+
11	2:N	2:N	
12	3:M	3:M	
13	2:N	2:N	
14	1:B	2:N	+
15	2:N	2:N	
16	3:M	3:M	
17	3:M	3:M	
18	2:N	2:N	
19	1:B	2:N	+
20	2:N	2:N	
Correctly Classified 13 < -- 65 %			
Incorrectly Classified 7 < -- 35 %			

Con el clasificador *NaiveBayes* se obtuvo 65 % de instancias correctamente clasificadas, es decir, 13 de 20 instancias.

NaiveBayesMultinomial

inst #	actual	predicted	error
1	1:B	2:N	+
2	1:B	2:N	+
3	2:N	2:N	
4	2:N	1:B	+
5	3:M	3:M	
6	2:N	2:N	
7	2:N	2:N	
8	1:B	2:N	+
9	2:N	1:B	+
10	3:M	1:B	+
11	2:N	2:N	
12	3:M	3:M	
13	2:N	2:N	
14	1:B	2:N	+
15	2:N	2:N	
16	3:M	3:M	
17	3:M	3:M	
18	2:N	2:N	
19	1:B	2:N	+
20	2:N	2:N	
Correctly Classified 12 < -- 60 %			
Incorrectly Classified 8 < -- 40 %			

Con el clasificador *NaiveBayesMultinomial* se obtuvo 60 % de instancias correctamente clasificadas, es decir, 12 de 20 instancias.

NaiveBayesMultinomialUpdateable

inst #	actual	predicted	error
1	1:B	2:N	+
2	1:B	2:N	+
3	2:N	2:N	
4	2:N	1:B	+
5	3:M	3:M	
6	2:N	2:N	
7	2:N	2:N	
8	1:B	2:N	+
9	2:N	1:B	+
10	3:M	1:B	+
11	2:N	2:N	
12	3:M	3:M	
13	2:N	2:N	
14	1:B	2:N	+
15	2:N	2:N	
16	3:M	3:M	
17	3:M	3:M	
18	2:N	2:N	
19	1:B	2:N	+
20	2:N	2:N	
Correctly Classified 12 < -- 60 %			
Incorrectly Classified 8 < -- 40 %			

Con el clasificador *NaiveBayesMultinomialUpdateable* se obtuvo 60 % de instancias correctamente clasificadas, es decir, 12 de 20 instancias.

NaiveBayesUpdate

inst #	actual	predicted	error
1	1:B	1:B	
2	1:B	2:N	+
3	2:N	2:N	
4	2:N	2:N	
5	3:M	3:M	
6	2:N	2:N	
7	2:N	3:M	+
8	1:B	2:N	+
9	2:N	1:B	+
10	3:M	2:N	+
11	2:N	2:N	
12	3:M	3:M	
13	2:N	2:N	
14	1:B	2:N	+
15	2:N	2:N	
16	3:M	3:M	
17	3:M	3:M	
18	2:N	2:N	
19	1:B	2:N	+
20	2:N	2:N	
Correctly Classified 13 < -- 65 %			
Incorrectly Classified 7 < -- 35 %			

Con el clasificador *NaiveBayesUpdate* se obtuvo 65 % de instancias correctamente clasificadas, es decir, 13 de 20 instancias.

Método funciones de clasificación (*functions*)

Logistic

inst #	actual	predicted	error
1	1:B	3:M	
2	1:B	1:B	+
3	2:N	2:N	
4	2:N	3:M	
5	3:M	2:N	
6	2:N	1:B	
7	2:N	3:M	+
8	1:B	1:B	+
9	2:N	2:N	+
10	3:M	3:M	+
11	2:N	1:B	
12	3:M	3:M	
13	2:N	2:N	
14	1:B	1:B	+
15	2:N	2:N	
16	3:M	3:M	
17	3:M	3:M	
18	2:N	2:N	
19	1:B	1:B	+
20	2:N	2:N	
Correctly Classified 14 < -- 70 %			
Incorrectly Classified 6 < -- 30 %			

Con el clasificador *Logistic* se obtuvo 70% de instancias correctamente clasificadas, es decir, 14 de 20 instancias.

MultilayerPerceptron

inst #	actual	predicted	error
1	1:B	3:M	+
2	1:B	2:N	+
3	2:N	2:N	
4	2:N	2:N	
5	3:M	3:M	
6	2:N	3:M	+
7	2:N	2:N	
8	1:B	2:N	+
9	2:N	2:N	
10	3:M	3:M	
11	2:N	2:N	
12	3:M	3:M	
13	2:N	2:N	
14	1:B	2:N	+
15	2:N	2:N	
16	3:M	3:M	
17	3:M	3:M	
18	2:N	3:M	+
19	1:B	2:N	+
20	2:N	2:N	
Correctly Classified 13 < -- 65 %			
Incorrectly Classified 7 < -- 35 %			

Con el clasificador *MultilayerPerceptron* se obtuvo 65% de instancias correctamente clasificadas, es decir, 13 de 20 instancias.

RBFNetwork

inst #	actual	predicted	error
1	1:B	3:M	+
2	1:B	3:M	+
3	2:N	3:M	+
4	2:N	3:M	+
5	3:M	3:M	
6	2:N	3:M	+
7	2:N	3:M	+
8	1:B	3:M	+
9	2:N	3:M	+
10	3:M	3:M	
11	2:N	3:M	+
12	3:M	3:M	
13	2:N	3:M	+
14	1:B	3:M	+
15	2:N	3:M	+
16	3:M	3:M	
17	3:M	3:M	
18	2:N	3:M	+
19	1:B	3:M	+
20	2:N	3:M	+
Correctly Classified 5 < -- 25 %			
Incorrectly Classified 15 < -- 75 %			

Con el clasificador *RBFNetwork* se obtuvo 25 % de instancias correctamente clasificadas, es decir, 5 de 20 instancias.

SimpleLogistic

inst #	actual	predicted	error
1	1:B	3:M	+
2	1:B	2:N	+
3	2:N	2:N	
4	2:N	2:N	
5	3:M	3:M	
6	2:N	2:N	
7	2:N	2:N	
8	1:B	2:N	+
9	2:N	3:M	+
10	3:M	3:M	
11	2:N	2:N	
12	3:M	3:M	
13	2:N	2:N	
14	1:B	2:N	+
15	2:N	2:N	
16	3:M	3:M	
17	3:M	3:M	
18	2:N	2:N	
19	1:B	2:N	+
20	2:N	2:N	
Correctly Classified 14 < -- 70 %			
Incorrectly Classified 6 < -- 30 %			

Con el clasificador *SimpleLogistic* se obtuvo 70 % de instancias correctamente clasificadas, es decir, 14 de 20 instancias.

SMO

inst #	actual	predicted	error
1	1:B	3:M	+
2	1:B	2:N	+
3	2:N	2:N	
4	2:N	2:N	
5	3:M	3:M	
6	2:N	3:M	+
7	2:N	2:N	
8	1:B	2:N	+
9	2:N	2:N	
10	3:M	3:M	
11	2:N	1:B	+
12	3:M	3:M	
13	2:N	2:N	
14	1:B	2:N	+
15	2:N	2:N	
16	3:M	3:M	
17	3:M	3:M	
18	2:N	2:N	
19	1:B	2:N	+
20	2:N	2:N	
Correctly Classified 13 < -- 65 %			
Incorrectly Classified 7 < -- 35 %			

Con el clasificador *SMO* se obtuvo 65% de instancias correctamente clasificadas, es decir, 13 de 20 instancias.

Método perezosos (*lazy*)

IB1

inst #	actual	predicted	error
1	1:B	3:M	+
2	1:B	2:N	+
3	2:N	2:N	
4	2:N	3:M	+
5	3:M	3:M	
6	2:N	1:B	+
7	2:N	2:N	
8	1:B	2:N	+
9	2:N	2:N	
10	3:M	3:M	
11	2:N	2:N	
12	3:M	3:M	
13	2:N	2:N	
14	1:B	2:N	+
15	2:N	2:N	
16	3:M	3:M	
17	3:M	3:M	
18	2:N	3:M	+
19	1:B	2:N	+
20	2:N	2:N	
Correctly Classified 12 < -- 60 %			
Incorrectly Classified 8 < -- 40 %			

Con el clasificador *IB1* se obtuvo 60 % de instancias correctamente clasificadas, es decir, 12 de 20 instancias.

IBk

inst #	actual	predicted	error
1	1:B	3:M	+
2	1:B	2:N	+
3	2:N	2:N	
4	2:N	3:M	+
5	3:M	3:M	
6	2:N	1:B	+
7	2:N	2:N	
8	1:B	2:N	+
9	2:N	2:N	
10	3:M	3:M	
11	2:N	2:N	
12	3:M	3:M	
13	2:N	2:N	
14	1:B	2:N	+
15	2:N	2:N	
16	3:M	3:M	
17	3:M	3:M	
18	2:N	3:M	+
19	1:B	2:N	+
20	2:N	2:N	
Correctly Classified 12 < -- 60 %			
Incorrectly Classified 8 < -- 40 %			

Con el clasificador *IBk* se obtuvo 60 % de instancias correctamente clasificadas, es decir, 12 de 20 instancias.

KStar

inst #	actual	predicted	error
1	1:B	1:B	
2	1:B	1:B	
3	2:N	1:B	+
4	2:N	1:B	+
5	3:M	1:B	+
6	2:N	1:B	+
7	2:N	1:B	+
8	1:B	1:B	
9	2:N	1:B	+
10	3:M	1:B	+
11	2:N	1:B	+
12	3:M	1:B	+
13	2:N	1:B	+
14	1:B	1:B	
15	2:N	1:B	+
16	3:M	1:B	+
17	3:M	1:B	+
18	2:N	1:B	+
19	1:B	1:B	
20	2:N	1:B	+
Correctly Classified 5 < -- 25 %			
Incorrectly Classified 15 < -- 75 %			

Con el clasificador *KStar* se obtuvo 25 % de instancias correctamente clasificadas, es decir, 5 de 20 instancias.

LWL

inst #	actual	predicted	error
1	1:B	1:B	
2	1:B	2:N	+
3	2:N	2:N	
4	2:N	3:M	+
5	3:M	1:B	+
6	2:N	2:N	
7	2:N	3:M	+
8	1:B	2:N	+
9	2:N	2:N	
10	3:M	2:N	+
11	2:N	2:N	
12	3:M	3:M	
13	2:N	2:N	
14	1:B	2:N	+
15	2:N	2:N	
16	3:M	3:M	
17	3:M	3:M	
18	2:N	3:M	+
19	1:B	2:N	+
20	2:N	2:N	
Correctly Classified 11 < -- 55 %			
Incorrectly Classified 9 < -- 45 %			

Con el clasificador *LWL* se obtuvo 55 % de instancias correctamente clasificadas, es decir, 11 de 20 instancias.

Metaclasificadores (*meta*)

AdaBoostM1

inst#	actual	predicted	error
1	1:B	3:M	+
2	1:B	3:M	+
3	2:N	3:M	+
4	2:N	3:M	+
5	3:M	3:M	
6	2:N	3:M	+
7	2:N	3:M	+
8	1:B	3:M	+
9	2:N	3:M	+
10	3:M	3:M	
11	2:N	3:M	+
12	3:M	3:M	
13	2:N	3:M	+
14	1:B	3:M	+
15	2:N	3:M	+
16	3:M	3:M	
17	3:M	3:M	
18	2:N	3:M	+
19	1:B	3:M	+
20	2:N	3:M	+
Correctly Classified 5 < -- 25 %			
Incorrectly Classified 15 < -- 75 %			

Con el clasificador *AdaBoostM1* se obtuvo 25% de instancias correctamente clasificadas, es decir, 5 de 20 instancias.

AttributeSelectedClassifier

inst#	actual	predicted	error
1	1:B	3:M	+
2	1:B	3:M	+
3	2:N	3:M	+
4	2:N	3:M	+
5	3:M	3:M	
6	2:N	3:M	+
7	2:N	3:M	+
8	1:B	3:M	+
9	2:N	3:M	+
10	3:M	3:M	
11	2:N	3:M	+
12	3:M	3:M	
13	2:N	3:M	+
14	1:B	3:M	+
15	2:N	3:M	+
16	3:M	3:M	
17	3:M	3:M	
18	2:N	3:M	+
19	1:B	3:M	+
20	2:N	3:M	+
Correctly Classified 5 < -- 25 %			
Incorrectly Classified 15 < -- 75 %			

Con el clasificador *AttributeSelectedClassifier* se obtuvo 25% de instancias correctamente clasificadas, es decir, 5 de 20 instancias.

Bagging

inst #	actual	predicted	error
1	1:B	2:N	+
2	1:B	2:N	+
3	2:N	2:N	
4	2:N	3:M	+
5	3:M	3:M	
6	2:N	3:M	+
7	2:N	2:N	
8	1:B	2:N	+
9	2:N	2:N	
10	3:M	3:M	
11	2:N	2:N	
12	3:M	3:M	
13	2:N	2:N	
14	1:B	2:N	+
15	2:N	2:N	
16	3:M	3:M	
17	3:M	3:M	
18	2:N	3:M	
19	1:B	2:N	+
20	2:N	2:N	+
Correctly Classified 12 < -- 60 %			
Incorrectly Classified 8 < -- 40 %			

Con el clasificador *Bagging* se obtuvo 60 % de instancias correctamente clasificadas, es decir, 12 de 20 instancias.

ClassificationViaClustering

inst #	actual	predicted	error
1	1:B	2:N	+
2	1:B	2:N	+
3	2:N	3:M	+
4	2:N	3:M	+
5	3:M	3:M	
6	2:N	2:N	
7	2:N	3:M	+
8	1:B	2:N	+
9	2:N	3:M	+
10	3:M	3:M	
11	2:N	2:N	
12	3:M	3:M	
13	2:N	3:M	+
14	1:B	2:N	+
15	2:N	3:M	+
16	3:M	3:M	
17	3:M	2:N	+
18	2:N	3:M	+
19	1:B	2:N	+
20	2:N	3:M	+
Correctly Classified 6 < -- 30 %			
Incorrectly Classified 14 < -- 70 %			

Con el clasificador *ClassificationViaClustering* se obtuvo 30 % de instancias correctamente clasificadas, es decir, 6 de 20 instancias.

ClassificationViaRegretion

inst #	actual	predicted	error
1	1:B	3:M	+
2	1:B	2:N	+
3	2:N	2:N	
4	2:N	3:M	+
5	3:M	3:M	
6	2:N	3:M	+
7	2:N	2:N	
8	1:B	2:N	+
9	2:N	2:N	
10	3:M	2:N	+
11	2:N	1:B	+
12	3:M	3:M	
13	2:N	2:N	
14	1:B	2:N	+
15	2:N	2:N	
16	3:M	3:M	
17	3:M	3:M	
18	2:N	2:N	
19	1:B	2:N	+
20	2:N	2:N	
Correctly Classified 11 < -- 55 %			
Incorrectly Classified 9 < -- 45 %			

Con el clasificador *ClassificationViaRegretion* se obtuvo 55 % de instancias correctamente clasificadas, es decir, 11 de 20 instancias.

CVParameterSelection

inst #	actual	predicted	error
1	1:B	3:M	+
2	1:B	3:M	+
3	2:N	3:M	+
4	2:N	3:M	+
5	3:M	3:M	
6	2:N	3:M	+
7	2:N	3:M	+
8	1:B	3:M	+
9	2:N	3:M	+
10	3:M	3:M	
11	2:N	3:M	+
12	3:M	3:M	
13	2:N	3:M	+
14	1:B	3:M	+
15	2:N	3:M	+
16	3:M	3:M	
17	3:M	3:M	
18	2:N	3:M	+
19	1:B	3:M	+
20	2:N	3:M	+
Correctly Classified 5 < -- 25 %			
Incorrectly Classified 15 < -- 75 %			

Con el clasificador *CVParameterSelection* se obtuvo 25 % de instancias correctamente clasificadas, es decir, 5 de 20 instancias.

Dagging

inst#	actual	predicted	error
1	1:B	2:N	+
2	1:B	2:N	+
3	2:N	2:N	
4	2:N	2:N	
5	3:M	3:M	
6	2:N	3:M	+
7	2:N	3:M	+
8	1:B	2:N	+
9	2:N	2:N	
10	3:M	3:M	
11	2:N	2:N	
12	3:M	3:M	
13	2:N	3:M	+
14	1:B	2:N	+
15	2:N	3:M	+
16	3:M	3:M	
17	3:M	3:M	
18	2:N	3:M	+
19	1:B	2:N	+
20	2:N	2:N	
Correctly Classified 10 < -- 50 %			
Incorrectly Classified 10 < -- 50 %			

Con el clasificador *Dagging* se obtuvo 50 % de instancias correctamente clasificadas, es decir, 10 de 20 instancias.

Decorate

inst#	actual	predicted	error
1	1:B	1:B	
2	1:B	1:B	
3	2:N	2:N	
4	2:N	2:N	
5	3:M	3:M	
6	2:N	2:N	
7	2:N	2:N	
8	1:B	1:B	
9	2:N	2:N	
10	3:M	3:M	
11	2:N	2:N	
12	3:M	3:M	
13	2:N	2:N	
14	1:B	1:B	
15	2:N	2:N	
16	3:M	3:M	
17	3:M	3:M	
18	2:N	2:N	
19	1:B	1:B	
20	2:N	1:B	+
Correctly Classified 19 < -- 95 %			
Incorrectly Classified 1 < -- 5 %			

Con el clasificador *Decorate* se obtuvo 95 % de instancias correctamente clasificadas, es decir, 19 de 20 instancias.

END

inst #	actual	predicted	error
1	1:B	2:N	+
2	1:B	3:M	+
3	2:N	2:N	
4	2:N	2:N	
5	3:M	1:B	+
6	2:N	3:M	+
7	2:N	2:N	
8	1:B	3:M	+
9	2:N	2:N	
10	3:M	3:M	
11	2:N	2:N	
12	3:M	3:M	
13	2:N	2:N	
14	1:B	3:M	+
15	2:N	2:N	
16	3:M	3:M	
17	3:M	3:M	
18	2:N	3:M	+
19	1:B	3:M	+
20	2:N	2:N	
Correctly Classified 12 < -- 60 %			
Incorrectly Classified 8 < -- 40 %			

Con el clasificador *END* se obtuvo 60 % de instancias correctamente clasificadas, es decir, 12 de 20 instancias.

FilteredClassifier

inst #	actual	predicted	error
1	1:B	3:M	+
2	1:B	3:M	+
3	2:N	3:M	+
4	2:N	3:M	+
5	3:M	3:M	
6	2:N	3:M	+
7	2:N	3:M	+
8	1:B	3:M	+
9	2:N	3:M	+
10	3:M	3:M	
11	2:N	3:M	+
12	3:M	3:M	
13	2:N	3:M	+
14	1:B	3:M	+
15	2:N	3:M	+
16	3:M	3:M	
17	3:M	3:M	
18	2:N	3:M	+
19	1:B	3:M	+
20	2:N	3:M	+
Correctly Classified 5 < -- 25 %			
Incorrectly Classified 15 < -- 75 %			

Con el clasificador *FilteredClassifier* se obtuvo 25 % de instancias correctamente clasificadas, es decir, 5 de 20 instancias.

Grading

inst#	actual	predicted	error
1	1:B	3:M	+
2	1:B	3:M	+
3	2:N	3:M	+
4	2:N	3:M	+
5	3:M	3:M	
6	2:N	3:M	+
7	2:N	3:M	+
8	1:B	3:M	+
9	2:N	3:M	+
10	3:M	3:M	
11	2:N	3:M	+
12	3:M	3:M	
13	2:N	3:M	+
14	1:B	3:M	+
15	2:N	3:M	+
16	3:M	3:M	
17	3:M	3:M	
18	2:N	3:M	+
19	1:B	3:M	+
20	2:N	3:M	+
Correctly Classified 5 < -- 25 %			
Incorrectly Classified 15 < -- 75 %			

Con el clasificador *Grading* se obtuvo 25 % de instancias correctamente clasificadas, es decir, 5 de 20 instancias.

MultiBoostAB

inst#	actual	predicted	error
1	1:B	3:M	+
2	1:B	3:M	+
3	2:N	3:M	+
4	2:N	3:M	+
5	3:M	3:M	
6	2:N	3:M	+
7	2:N	3:M	+
8	1:B	3:M	+
9	2:N	3:M	+
10	3:M	3:M	
11	2:N	3:M	+
12	3:M	3:M	
13	2:N	3:M	+
14	1:B	3:M	+
15	2:N	3:M	+
16	3:M	3:M	
17	3:M	3:M	
18	2:N	3:M	+
19	1:B	3:M	+
20	2:N	3:M	+
Correctly Classified 5 < -- 25 %			
Incorrectly Classified 15 < -- 75 %			

Con el clasificador *MultiBoostAB* se obtuvo 25 % de instancias correctamente clasificadas, es decir, 5 de 20 instancias.

MultiClassClassifier

inst#	actual	predicted	error
1	1:B	3:M	+
2	1:B	3:M	+
3	2:N	2:N	
4	2:N	3:M	+
5	3:M	2:N	+
6	2:N	3:M	+
7	2:N	3:M	+
8	1:B	3:M	+
9	2:N	2:N	
10	3:M	3:M	
11	2:N	1:B	+
12	3:M	3:M	
13	2:N	2:N	
14	1:B	3:M	+
15	2:N	2:N	
16	3:M	3:M	
17	3:M	3:M	
18	2:N	2:N	
19	1:B	3:M	+
20	2:N	2:N	
Correctly Classified 10 < -- 50 %			
Incorrectly Classified 10 < -- 50 %			

Con el clasificador *MultiClassClassifier* se obtuvo 50 % de instancias correctamente clasificadas, es decir, 10 de 20 instancias.

MultiScheme

inst#	actual	predicted	error
1	1:B	3:M	+
2	1:B	3:M	+
3	2:N	3:M	+
4	2:N	3:M	+
5	3:M	3:M	
6	2:N	3:M	+
7	2:N	3:M	+
8	1:B	3:M	+
9	2:N	3:M	+
10	3:M	3:M	
11	2:N	3:M	+
12	3:M	3:M	
13	2:N	3:M	+
14	1:B	3:M	+
15	2:N	3:M	+
16	3:M	3:M	
17	3:M	3:M	
18	2:N	3:M	+
19	1:B	3:M	+
20	2:N	3:M	+
Correctly Classified 5 < -- 25 %			
Incorrectly Classified 15 < -- 75 %			

Con el clasificador *MultiScheme* se obtuvo 25 % de instancias correctamente clasificadas, es decir, 5 de 20 instancias.

OrdinalClassClassifier

inst #	actual	predicted	error
1	1:B	2:N	+
2	1:B	3:M	+
3	2:N	2:N	
4	2:N	2:N	
5	3:M	1:B	+
6	2:N	3:M	+
7	2:N	2:N	
8	1:B	3:M	+
9	2:N	2:N	
10	3:M	3:M	
11	2:N	3:M	+
12	3:M	3:M	
13	2:N	2:N	
14	1:B	3:M	+
15	2:N	2:N	
16	3:M	3:M	
17	3:M	3:M	
18	2:N	2:N	
19	1:B	3:M	+
20	2:N	2:N	
Correctly Classified 12 < -- 60 %			
Incorrectly Classified 8 < -- 40 %			

Con el clasificador *OrdinalClassClassifier* se obtuvo 60% de instancias correctamente clasificadas, es decir, 12 de 20 instancias.

RacedIncrementalLogitBoost

inst #	actual	predicted	error
1	1:B	3:M	+
2	1:B	3:M	+
3	2:N	3:M	+
4	2:N	3:M	+
5	3:M	3:M	
6	2:N	3:M	+
7	2:N	3:M	+
8	1:B	3:M	+
9	2:N	3:M	+
10	3:M	3:M	
11	2:N	3:M	+
12	3:M	3:M	
13	2:N	3:M	+
14	1:B	3:M	+
15	2:N	3:M	+
16	3:M	3:M	
17	3:M	3:M	
18	2:N	3:M	+
19	1:B	3:M	+
20	2:N	3:M	+
Correctly Classified 5 < -- 25 %			
Incorrectly Classified 15 < -- 75 %			

Con el clasificador *RacedIncrementalLogitBoost* se obtuvo 25% de instancias correctamente clasificadas, es decir, 5 de 20 instancias.

RandomCommitee

inst #	actual	predicted	error
1	1:B	3:M	+
2	1:B	2:N	+
3	2:N	2:N	
4	2:N	2:N	
5	3:M	3:M	
6	2:N	2:N	
7	2:N	2:N	
8	1:B	2:N	
9	2:N	2:N	+
10	3:M	3:M	
11	2:N	2:N	
12	3:M	3:M	
13	2:N	2:N	
14	1:B	2:N	+
15	2:N	2:N	
16	3:M	3:M	
17	3:M	3:M	
18	2:N	2:N	
19	1:B	2:N	+
20	2:N	2:N	
Correctly Classified 15 < -- 75 %			
Incorrectly Classified 5 < -- 25 %			

Con el clasificador *RandomCommitee* se obtuvo 75 % de instancias correctamente clasificadas, es decir, 15 de 20 instancias.

RandomSubSpace

inst #	actual	predicted	error
1	1:B	3:M	+
2	1:B	3:M	+
3	2:N	3:M	+
4	2:N	3:M	+
5	3:M	3:M	
6	2:N	3:M	+
7	2:N	3:M	+
8	1:B	3:M	+
9	2:N	3:M	+
10	3:M	3:M	
11	2:N	3:M	+
12	3:M	3:M	
13	2:N	3:M	+
14	1:B	3:M	+
15	2:N	3:M	+
16	3:M	3:M	
17	3:M	3:M	
18	2:N	3:M	+
19	1:B	3:M	+
20	2:N	3:M	+
Correctly Classified 5 < -- 25 %			
Incorrectly Classified 15 < -- 75 %			

Con el clasificador *RandomSubSpace* se obtuvo 25 % de instancias correctamente clasificadas, es decir, 5 de 20 instancias.

RotationForest

inst #	actual	predicted	error
1	1:B	3:M	+
2	1:B	2:N	+
3	2:N	2:N	
4	2:N	2:N	
5	3:M	3:M	
6	2:N	2:N	
7	2:N	3:M	+
8	1:B	2:N	+
9	2:N	2:N	
10	3:M	3:M	
11	2:N	1:B	+
12	3:M	3:M	
13	2:N	3:M	+
14	1:B	2:N	+
15	2:N	2:N	
16	3:M	3:M	
17	3:M	3:M	
18	2:N	3:M	
19	1:B	2:N	+
20	2:N	2:N	+
Correctly Classified 11 < -- 55 %			
Incorrectly Classified 9 < -- 45 %			

Con el clasificador *RotationForest* se obtuvo 55% de instancias correctamente clasificadas, es decir, 11 de 20 instancias.

Stacking

inst #	actual	predicted	error
1	1:B	3:M	+
2	1:B	3:M	+
3	2:N	3:M	+
4	2:N	3:M	+
5	3:M	3:M	
6	2:N	3:M	+
7	2:N	3:M	+
8	1:B	3:M	+
9	2:N	3:M	+
10	3:M	3:M	
11	2:N	3:M	+
12	3:M	3:M	
13	2:N	3:M	+
14	1:B	3:M	+
15	2:N	3:M	+
16	3:M	3:M	
17	3:M	3:M	
18	2:N	3:M	+
19	1:B	3:M	+
20	2:N	3:M	+
Correctly Classified 5 < -- 25 %			
Incorrectly Classified 15 < -- 75 %			

Con el clasificador *Stacking* se obtuvo 25% de instancias correctamente clasificadas, es decir, 5 de 20 instancias.

StackingC

inst#	actual	predicted	error
1	1:B	3:M	+
2	1:B	3:M	+
3	2:N	3:M	+
4	2:N	3:M	+
5	3:M	3:M	
6	2:N	3:M	+
7	2:N	3:M	+
8	1:B	3:M	+
9	2:N	3:M	+
10	3:M	3:M	
11	2:N	3:M	+
12	3:M	3:M	
13	2:N	3:M	+
14	1:B	3:M	+
15	2:N	3:M	+
16	3:M	3:M	
17	3:M	3:M	
18	2:N	3:M	+
19	1:B	3:M	+
20	2:N	3:M	+
Correctly Classified 5 < -- 25 %			
Incorrectly Classified 15 < -- 75 %			

Con el clasificador *StackingC* se obtuvo 25 % de instancias correctamente clasificadas, es decir, 5 de 20 instancias.

Vote

inst#	actual	predicted	error
1	1:B	3:M	+
2	1:B	3:M	+
3	2:N	3:M	+
4	2:N	3:M	+
5	3:M	3:M	
6	2:N	3:M	+
7	2:N	3:M	+
8	1:B	3:M	+
9	2:N	3:M	+
10	3:M	3:M	
11	2:N	3:M	+
12	3:M	3:M	
13	2:N	3:M	+
14	1:B	3:M	+
15	2:N	3:M	+
16	3:M	3:M	
17	3:M	3:M	
18	2:N	3:M	+
19	1:B	3:M	+
20	2:N	3:M	+
Correctly Classified 5 < -- 25 %			
Incorrectly Classified 15 < -- 75 %			

Con el clasificador *Vote* se obtuvo 25 % de instancias correctamente clasificadas, es decir, 5 de 20 instancias.

Método varios (*misc*)

HyperPipes

inst#	actual	predicted	error
1	1:B	3:M	+
2	1:B	2:N	+
3	2:N	2:N	
4	2:N	2:N	
5	3:M	3:M	
6	2:N	2:N	
7	2:N	3:M	+
8	1:B	2:N	+
9	2:N	2:N	
10	3:M	1:B	+
11	2:N	3:M	+
12	3:M	3:M	
13	2:N	3:M	+
14	1:B	2:N	+
15	2:N	2:N	
16	3:M	3:M	
17	3:M	3:M	
18	2:N	3:M	+
19	1:B	2:N	+
20	2:N	2:N	
Correctly Classified 10 < -- 50 %			
Incorrectly Classified 10 < -- 50 %			

Con el clasificador *HyperPipes* se obtuvo 50 % de instancias correctamente clasificadas, es decir, 10 de 20 instancias.

VFI

inst#	actual	predicted	error
1	1:B	1:B	
2	1:B	2:N	+
3	2:N	1:B	+
4	2:N	1:B	+
5	3:M	3:M	
6	2:N	1:B	+
7	2:N	3:M	+
8	1:B	2:N	+
9	2:N	1:B	+
10	3:M	1:B	+
11	2:N	1:B	+
12	3:M	3:M	
13	2:N	3:M	+
14	1:B	2:N	+
15	2:N	3:M	+
16	3:M	3:M	
17	3:M	3:M	
18	2:N	3:M	+
19	1:B	2:N	+
20	2:N	1:B	+
Correctly Classified 5 < -- 25 %			
Incorrectly Classified 15 < -- 75 %			

Con el clasificador *VFI* se obtuvo 25 % de instancias correctamente clasificadas, es decir, 5 de 20 instancias.

Método de generación de reglas (*rules*)

ConjunctiveRule

inst#	actual	predicted	error
1	1:B	3:M	+
2	1:B	3:M	+
3	2:N	3:M	+
4	2:N	3:M	+
5	3:M	3:M	
6	2:N	3:M	+
7	2:N	3:M	+
8	1:B	3:M	+
9	2:N	3:M	+
10	3:M	3:M	
11	2:N	3:M	+
12	3:M	3:M	
13	2:N	3:M	+
14	1:B	3:M	+
15	2:N	3:M	+
16	3:M	3:M	
17	3:M	3:M	
18	2:N	3:M	+
19	1:B	3:M	+
20	2:N	3:M	+
Correctly Classified 5 < -- 25 %			
Incorrectly Classified 15 < -- 75 %			

Con el clasificador *ConjunctiveRule* se obtuvo 25 % de instancias correctamente clasificadas, es decir, 5 de 20 instancias.

DecisionTable

inst#	actual	predicted	error
1	1:B	3:M	+
2	1:B	3:M	+
3	2:N	3:M	+
4	2:N	3:M	+
5	3:M	3:M	
6	2:N	3:M	+
7	2:N	3:M	+
8	1:B	3:M	+
9	2:N	3:M	+
10	3:M	3:M	
11	2:N	3:M	+
12	3:M	3:M	
13	2:N	3:M	+
14	1:B	3:M	+
15	2:N	3:M	+
16	3:M	3:M	
17	3:M	3:M	
18	2:N	3:M	+
19	1:B	3:M	+
20	2:N	3:M	+
Correctly Classified 5 < -- 25 %			
Incorrectly Classified 15 < -- 75 %			

Con el clasificador *DecisionTable* se obtuvo 25 % de instancias correctamente clasificadas, es decir, 5 de 20 instancias.

DTNB

inst#	actual	predicted	error
1	1:B	3:M	+
2	1:B	3:M	+
3	2:N	3:M	+
4	2:N	3:M	+
5	3:M	3:M	
6	2:N	3:M	+
7	2:N	3:M	+
8	1:B	3:M	+
9	2:N	3:M	+
10	3:M	3:M	
11	2:N	3:M	+
12	3:M	3:M	
13	2:N	3:M	+
14	1:B	3:M	+
15	2:N	3:M	+
16	3:M	3:M	
17	3:M	3:M	
18	2:N	3:M	+
19	1:B	3:M	+
20	2:N	3:M	+
Correctly Classified 5 < -- 25 %			
Incorrectly Classified 15 < -- 75 %			

Con el clasificador *DTNB* se obtuvo 25 % de instancias correctamente clasificadas, es decir, 5 de 20 instancias.

JRip

inst#	actual	predicted	error
1	1:B	3:M	+
2	1:B	3:M	+
3	2:N	3:M	+
4	2:N	3:M	+
5	3:M	3:M	
6	2:N	3:M	+
7	2:N	3:M	+
8	1:B	3:M	+
9	2:N	2:N	
10	3:M	3:M	
11	2:N	3:M	+
12	3:M	3:M	
13	2:N	3:M	+
14	1:B	3:M	+
15	2:N	3:M	+
16	3:M	3:M	
17	3:M	3:M	
18	2:N	3:M	+
19	1:B	3:M	+
20	2:N	3:M	+
Correctly Classified 6 < -- 30 %			
Incorrectly Classified 14 < -- 70 %			

Con el clasificador *JRip* se obtuvo 30 % de instancias correctamente clasificadas, es decir, 6 de 20 instancias.

NNge

inst #	actual	predicted	error
1	1:B	3:M	+
2	1:B	2:N	+
3	2:N	2:N	
4	2:N	3:M	+
5	3:M	3:M	
6	2:N	3:M	+
7	2:N	3:M	+
8	1:B	2:N	+
9	2:N	2:N	
10	3:M	3:M	
11	2:N	1:B	+
12	3:M	3:M	
13	2:N	3:M	
14	1:B	2:N	+
15	2:N	2:N	+
16	3:M	3:M	
17	3:M	3:M	
18	2:N	3:M	+
19	1:B	2:N	+
20	2:N	2:N	
Correctly Classified 9 < -- 45 %			
Incorrectly Classified 11 < -- 55 %			

Con el clasificador *NNge* se obtuvo 45 % de instancias correctamente clasificadas, es decir, 9 de 20 instancias.

OneR

inst #	actual	predicted	error
1	1:B	3:M	+
2	1:B	1:B	
3	2:N	3:M	+
4	2:N	3:M	+
5	3:M	3:M	
6	2:N	2:N	
7	2:N	3:M	+
8	1:B	1:B	
9	2:N	3:M	+
10	3:M	2:N	+
11	2:N	2:N	
12	3:M	3:M	
13	2:N	3:M	+
14	1:B	1:B	
15	2:N	2:N	
16	3:M	3:M	
17	3:M	3:M	
18	2:N	3:M	+
19	1:B	1:B	
20	2:N	3:M	+
Correctly Classified 11 < -- 55 %			
Incorrectly Classified 9 < -- 45 %			

Con el clasificador *OneR* se obtuvo 55 % de instancias correctamente clasificadas, es decir, 11 de 20 instancias.

PART

inst #	actual	predicted	error
1	1:B	3:M	+
2	1:B	2:N	+
3	2:N	2:N	
4	2:N	2:N	
5	3:M	2:N	+
6	2:N	3:M	+
7	2:N	3:M	+
8	1:B	2:N	+
9	2:N	2:N	
10	3:M	3:M	
11	2:N	2:N	
12	3:M	3:M	
13	2:N	2:N	
14	1:B	2:N	+
15	2:N	2:N	
16	3:M	3:M	
17	3:M	3:M	
18	2:N	2:N	
19	1:B	2:N	+
20	2:N	2:N	
Correctly Classified 12 < -- 60 %			
Incorrectly Classified 8 < -- 40 %			

Con el clasificador *PART* se obtuvo 60 % de instancias correctamente clasificadas, es decir, 12 de 20 instancias.

Ridor

inst #	actual	predicted	error
1	1:B	3:M	+
2	1:B	2:N	+
3	2:N	3:M	+
4	2:N	3:M	+
5	3:M	3:M	
6	2:N	3:M	+
7	2:N	3:M	+
8	1:B	2:N	+
9	2:N	2:N	
10	3:M	3:M	
11	2:N	2:N	
12	3:M	3:M	
13	2:N	2:N	
14	1:B	2:N	+
15	2:N	2:N	
16	3:M	3:M	
17	3:M	3:M	
18	2:N	3:M	+
19	1:B	2:N	+
20	2:N	3:M	+
Correctly Classified 9 < -- 45 %			
Incorrectly Classified 11 < -- 55 %			

Con el clasificador *Ridor* se obtuvo 45 % de instancias correctamente clasificadas, es decir, 9 de 20 instancias.

ZeroR

inst#	actual	predicted	error
1	1:B	3:M	+
2	1:B	3:M	+
3	2:N	3:M	+
4	2:N	3:M	+
5	3:M	3:M	
6	2:N	3:M	+
7	2:N	3:M	+
8	1:B	3:M	+
9	2:N	3:M	+
10	3:M	3:M	
11	2:N	3:M	+
12	3:M	3:M	
13	2:N	3:M	+
14	1:B	3:M	+
15	2:N	3:M	+
16	3:M	3:M	
17	3:M	3:M	
18	2:N	3:M	+
19	1:B	3:M	+
20	2:N	3:M	+
Correctly Classified 5 -- 25 %			
Incorrectly Classified 15 <-- 75 %			

Con el clasificador *ZeroR* se obtuvo 25 % de instancias correctamente clasificadas, es decir, 5 de 20 instancias.

Método de generación de árboles de decisión (*trees*)

BFTree

inst#	actual	predicted	error
1	1:B	3:M	+
2	1:B	3:M	+
3	2:N	3:M	+
4	2:N	3:M	+
5	3:M	3:M	
6	2:N	3:M	+
7	2:N	3:M	+
8	1:B	3:M	+
9	2:N	3:M	+
10	3:M	3:M	
11	2:N	3:M	+
12	3:M	3:M	
13	2:N	3:M	+
14	1:B	3:M	+
15	2:N	3:M	+
16	3:M	3:M	
17	3:M	3:M	
18	2:N	3:M	+
19	1:B	3:M	+
20	2:N	3:M	+
Correctly Classified 5 < -- 25 %			
Incorrectly Classified 15 < -- 75 %			

Con el clasificador *BFTree* se obtuvo 25 % de instancias correctamente clasificadas, es decir, 5 de 20 instancias.

DecisionStump

inst#	actual	predicted	error
1	1:B	3:M	+
2	1:B	3:M	+
3	2:N	3:M	+
4	2:N	3:M	+
5	3:M	3:M	
6	2:N	3:M	+
7	2:N	3:M	+
8	1:B	3:M	+
9	2:N	3:M	+
10	3:M	3:M	
11	2:N	3:M	+
12	3:M	3:M	
13	2:N	3:M	+
14	1:B	3:M	+
15	2:N	3:M	+
16	3:M	3:M	
17	3:M	3:M	
18	2:N	3:M	+
19	1:B	3:M	+
20	2:N	3:M	+
Correctly Classified 5 < -- 25 %			
Incorrectly Classified 15 < -- 75 %			

Con el clasificador *DecisionStump* se obtuvo 25 % de instancias correctamente clasificadas, es decir, 5 de 20 instancias.

FT

inst #	actual	predicted	error
1	1:B	3:M	+
2	1:B	2:N	+
3	2:N	2:N	
4	2:N	3:M	+
5	3:M	3:M	
6	2:N	3:M	+
7	2:N	2:N	
8	1:B	2:N	+
9	2:N	2:N	
10	3:M	3:M	
11	2:N	1:B	+
12	3:M	3:M	
13	2:N	2:N	
14	1:B	2:N	+
15	2:N	2:N	
16	3:M	3:M	
17	3:M	3:M	
18	2:N	2:N	
19	1:B	2:N	+
20	2:N	2:N	
Correctly Classified 12 < -- 60 %			
Incorrectly Classified 8 < -- 40 %			

Con el clasificador *FT* se obtuvo 60 % de instancias correctamente clasificadas, es decir, 12 de 20 instancias.

J48

inst #	actual	predicted	error
1	1:B	3:M	+
2	1:B	2:N	+
3	2:N	2:N	
4	2:N	3:M	+
5	3:M	3:M	
6	2:N	3:M	+
7	2:N	2:N	
8	1:B	2:N	+
9	2:N	2:N	
10	3:M	3:M	
11	2:N	2:N	
12	3:M	3:M	
13	2:N	3:M	+
14	1:B	2:N	+
15	2:N	2:N	
16	3:M	3:M	
17	3:M	3:M	
18	2:N	3:M	+
19	1:B	2:N	+
20	2:N	2:N	
Correctly Classified 11 < -- 55 %			
Incorrectly Classified 9 < -- 45 %			

Con el clasificador *J48* se obtuvo 55 % de instancias correctamente clasificadas, es decir, 11 de 20 instancias.

J48graft

inst #	actual	predicted	error
1	1:B	3:M	+
2	1:B	2:N	+
3	2:N	2:N	
4	2:N	3:M	+
5	3:M	3:M	
6	2:N	3:M	+
7	2:N	2:N	
8	1:B	2:N	+
9	2:N	2:N	
10	3:M	3:M	
11	2:N	2:N	
12	3:M	3:M	
13	2:N	3:M	+
14	1:B	2:N	+
15	2:N	2:N	
16	3:M	3:M	
17	3:M	3:M	
18	2:N	3:M	+
19	1:B	2:N	+
20	2:N	2:N	
Correctly Classified 12 < -- 55 %			
Incorrectly Classified 8 < -- 45 %			

Con el clasificador *J48graft* se obtuvo 60 % de instancias correctamente clasificadas, es decir, 12 de 20 instancias.

LADTree

inst #	actual	predicted	error
1	1:B	3:M	+
2	1:B	2:N	+
3	2:N	3:M	+
4	2:N	2:N	
5	3:M	1:B	+
6	2:N	3:M	+
7	2:N	3:M	+
8	1:B	2:N	+
9	2:N	2:N	
10	3:M	3:M	
11	2:N	2:N	
12	3:M	3:M	
13	2:N	2:N	
14	1:B	2:N	+
15	2:N	2:N	
16	3:M	3:M	
17	3:M	3:M	
18	2:N	3:M	
19	1:B	2:N	+
20	2:N	2:N	+
Correctly Classified 10 < -- 50 %			
Incorrectly Classified 10 < -- 50 %			

Con el clasificador *LADTree* se obtuvo 50 % de instancias correctamente clasificadas, es decir, 10 de 20 instancias.

LMT

inst #	actual	predicted	error
1	1:B	3:M	+
2	1:B	2:N	+
3	2:N	2:N	
4	2:N	3:M	+
5	3:M	3:M	
6	2:N	2:N	
7	2:N	2:N	
8	1:B	2:N	+
9	2:N	2:N	
10	3:M	3:M	
11	2:N	2:N	
12	3:M	3:M	
13	2:N	2:N	
14	1:B	2:N	+
15	2:N	2:N	
16	3:M	3:M	
17	3:M	3:M	
18	2:N	2:N	
19	1:B	2:N	+
20	2:N	2:N	
Correctly Classified 14 < -- 70 %			
Incorrectly Classified 6 < -- 30 %			

Con el clasificador *LMT* se obtuvo 70 % de instancias correctamente clasificadas, es decir, 14 de 20 instancias.

NBTree

inst #	actual	predicted	error
1	1:B	2:N	+
2	1:B	2:N	+
3	2:N	3:M	+
4	2:N	2:N	
5	3:M	3:M	
6	2:N	3:M	+
7	2:N	2:N	
8	1:B	2:N	+
9	2:N	2:N	
10	3:M	3:M	
11	2:N	2:N	
12	3:M	3:M	
13	2:N	2:N	
14	1:B	2:N	+
15	2:N	2:N	
16	3:M	3:M	
17	3:M	3:M	
18	2:N	3:M	+
19	1:B	2:N	+
20	2:N	3:M	+
Correctly Classified 11 < -- 55 %			
Incorrectly Classified 9 < -- 45 %			

Con el clasificador *NBTree* se obtuvo 55 % de instancias correctamente clasificadas, es decir, 11 de 20 instancias.

RandomTree

inst #	actual	predicted	error
1	1:B	2:N	+
2	1:B	2:N	+
3	2:N	3:M	+
4	2:N	2:N	
5	3:M	3:M	
6	2:N	2:N	
7	2:N	1:B	+
8	1:B	2:N	+
9	2:N	2:N	
10	3:M	3:M	
11	2:N	2:N	
12	3:M	3:M	
13	2:N	3:M	+
14	1:B	2:N	+
15	2:N	2:N	
16	3:M	3:M	
17	3:M	3:M	
18	2:N	2:N	
19	1:B	2:N	+
20	2:N	3:M	+
Correctly Classified 11 < -- 55 %			
Incorrectly Classified 9 < -- 45 %			

Con el clasificador *RandomTree* se obtuvo 55 % de instancias correctamente clasificadas, es decir, 11 de 20 instancias.

REPTree

inst #	actual	predicted	error
1	1:B	3:M	+
2	1:B	3:M	+
3	2:N	3:M	+
4	2:N	3:M	+
5	3:M	3:M	
6	2:N	3:M	+
7	2:N	3:M	+
8	1:B	3:M	+
9	2:N	3:M	+
10	3:M	3:M	
11	2:N	3:M	+
12	3:M	3:M	
13	2:N	3:M	+
14	1:B	3:M	+
15	2:N	3:M	+
16	3:M	3:M	
17	3:M	3:M	
18	2:N	3:M	+
19	1:B	3:M	+
20	2:N	3:M	+
Correctly Classified 5 < -- 25 %			
Incorrectly Classified 15 < -- 75 %			

Con el clasificador *REPTree* se obtuvo 25 % de instancias correctamente clasificadas, es decir, 5 de 20 instancias.

SimpleCart

inst#	actual	predicted	error
1	1:B	3:M	+
2	1:B	3:M	+
3	2:N	3:M	+
4	2:N	3:M	+
5	3:M	3:M	
6	2:N	3:M	+
7	2:N	3:M	+
8	1:B	3:M	+
9	2:N	3:M	+
10	3:M	3:M	
11	2:N	3:M	+
12	3:M	3:M	
13	2:N	3:M	+
14	1:B	3:M	+
15	2:N	3:M	+
16	3:M	3:M	
17	3:M	3:M	
18	2:N	3:M	+
19	1:B	3:M	+
20	2:N	3:M	+
Correctly Classified 5 < -- 25 %			
Incorrectly Classified 15 < -- 75 %			

Con el clasificador *SimpleCart* se obtuvo 25 % de instancias correctamente clasificadas, es decir, 5 de 20 instancias.

UserClassifier

inst#	actual	predicted	error
1	1:B	3:M	+
2	1:B	3:M	+
3	2:N	3:M	+
4	2:N	3:M	+
5	3:M	3:M	
6	2:N	3:M	+
7	2:N	3:M	+
8	1:B	3:M	+
9	2:N	3:M	+
10	3:M	3:M	
11	2:N	3:M	+
12	3:M	3:M	
13	2:N	3:M	+
14	1:B	3:M	+
15	2:N	3:M	+
16	3:M	3:M	
17	3:M	3:M	
18	2:N	3:M	+
19	1:B	3:M	+
20	2:N	3:M	+
Correctly Classified 5 < -- 25 %			
Incorrectly Classified 15 < -- 75 %			

Con el clasificador *UserClassifier* se obtuvo 25 % de instancias correctamente clasificadas, es decir, 5 de 20 instancias.

Anexo 2

Código: Módulo Pantalla de inicio

```
% -----
function segmentacion_Callback(hObject, eventdata, handles)
% hObject    handle to segmentacion (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
CorteNormalizado

% -----
function filtrado_Callback(hObject, eventdata, handles)
% hObject    handle to filtrado (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
ProcesaM

% -----
function clasificacion_Callback(hObject, eventdata, handles)
% hObject    handle to clasificacion (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
Clasificacion

% -----
function salir_Callback(hObject, eventdata, handles)
% hObject    handle to salir (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% -----
function manual_Callback(hObject, eventdata, handles)
% hObject    handle to manual (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
winopen('manual.docx')

% -----
function acercaD_Callback(hObject, eventdata, handles)
% hObject    handle to acercaD (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
```

```
% handles      structure with handles and user data (see GUIDATA)
msgbox('Detección de anomalías cancerígenas.
Por: Eddy Sánchez UJAT-DAIS', 'Acerca de...');
```

Código: Módulo Filtrar

```
% PARAMETROS DE LA TRANSFORMADA:
nlevels = [1, 1, 3, 2] ;           % Decomposition level
pfilter = 'maxflat' ;             % Pyramidal filter
dfilter = 'dmaxflat7' ;          % Directional filter

set(handles.Texto, 'String', 'Transformando..');
drawnow
% TRANSFORMAR: Nonsampled Contourlet decomposition
coeffs = nsctdec( double(handles.im), nlevels, dfilter, pfilter );
uiwaitbar(H, 0.37);
set(handles.Texto, 'String', 'Obtener edges..');
drawnow
%DEFINE EL NUMERO DE ORIENTACIONES POR NIVEL
nDirecciones=[0 2 2 8 4];
%Tmax=0; %MAYOR THRESHOLD EN TODOS LOS NIVELES
Tmax=0;
NumNiveles=4;
disp(['tiempo transcurido:', num2str(etime(clock, t)), 'Seg']);
disp('Obtener edges..'); t = clock;

%LISTAR TODOS LOS NIVELES
for Nivel=[2,3,4,5]%2:(NumNiveles+1)
    %PROCESAR NIVEL
    for i=1:nDirecciones(Nivel)
        %figure
        %ENCONTRAR THRESHOLD LOCAL
        xx=coeffs{Nivel}{i};
        Var= std2(xx); %Standard deviation of matrix elements
        T{Nivel}{i}=(3*Var);

        %APLICAR FUNCION PARA ENCONTRAR BORDES
        Bordes{Nivel}{i}=edge(coeffs{Nivel}{i}, 'prewitt', T{Nivel}{i});
        %imshow(bordes{i}, [])
        %title( ['Bordes nivel 4, ', num2str(i)] ) ;

        %VER SI ES EL MAYOR THRESHOLD
        if( T{Nivel}{i}>Tmax )
            Tmax=T{Nivel}{i};
        end
    end
end
```

```

end
end
end

uiwaitbar(H,0.5);
set(handles.Texto,'String','Modificando Subbandas..');
drawnow
cof=coeffs;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% MODIFICAR LAS SUBBANDAS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
disp('Modificando');
MiConta=1;

for Nivel=[2,3,4,5]%2:(NumNiveles+1)

    %PROCESAR NIVEL
    for i=1:nDirecciones(Nivel)

        Var= std2(coeffs{Nivel}{i});

        %OBTENER LA GANANCIA GLOBAL
        %g=(Tmax*Var*4)/T{Nivel}{i};
        g=(1/Var*4);

        [X,Y]=size( coeffs{Nivel}{i} );

        for x=1:X
            for y=1:Y
                tempo=Bordes{Nivel}{i};
                if( tempo(x,y)==1)
                    cof{Nivel}{i}(x,y)=coeffs{Nivel}{i}(x,y)*g;
                else
                    %3
                    cof{Nivel}{i}(x,y)=(1/Var*3)*coeffs{Nivel}{i}(x,y);
                end
            end
        end
    end
end
uiwaitbar(H,0.5+MiConta*0.0325);MiConta=MiConta+1;
drawnow
end

uiwaitbar(H,0.63);
set(handles.Texto,'String','Reconstruyendo..');

```



```

'Parent',h,...
'EdgeColor','none',...
'EraseMode','none');

% --- Executes on button press in cargarImagen.
function cargarImagen_Callback(hObject, eventdata, handles)
try
    [filename,pathname] = uigetfile({'*.bmp';'*.pgm';'*.png';'*.jpg'});
    7
    if isequal(filename,0)
        %Do nothing yet
    else
        handles.myImage = imread(fullfile(pathname, filename));

        [X, myMap] = gray2ind(handles.myImage);
        mImagen2 = ind2rgb(X,myMap);

        image(mImagen2, 'Parent', handles.axes1);
        colormap(gray);
    end
    guidata(hObject, handles);
catch
    msgbox('Error al cargar la imagen')
end

% --- Executes on button press in segmentar.
function segmentar_Callback(hObject, eventdata, handles)

%-----
% Se segmenta manualmente la zona de interés
%-----
segment = imcrop(handles.myImage);
imwrite (segment, 'zonaSegmentada.pgm');

%-----
% Se normaliza la zona de interés a 1024x1024
%-----
norm = imresize (segment, [512 512]);
imshow(norm);

%-----
% Se obtiene el histograma de la zona de interés
%-----
figure

```

```

hold on
title ('Histograma de frecuencia de la zona de interés')
ylabel('Amplitud')
imhist (norm)
hold off

%-----
% Se exporta el histograma obtenido a excel (conversión a tabla)
%-----

% Primero: Se almacena el histograma en una variable -----
xlsx = imhist (norm)

% Segundo: Se exporta la variable(histograma) a .xlsx(excel) -----
xlswrite('Histograma.xlsx',xlsx,'Hoja1','A5:IV5, A5:IV5')

% Tercero: Se pregunta -----
%%%%%%%% MODULO "¿ANORMALIDAD? B/N/M"

function anorm_Callback(hObject, eventdata, handles)
% Declaracion de variables
anormB = ['B']; % indica que la anormalidad es "benigna"
anormN = ['N']; % indica que la anormalidad es "normal"
anormM = ['M']; % indica que la anormalidad es "maligna"

val = get(hObject,'String'); % Se obtiene valor de la caja de texto

if (val == anormB) % Se compara el valor obtenido con el tipo de
anormalidad
    xlswrite('Histograma.xlsx', anormB, 'Hoja1', 'IW5');
    % Se inserta "B" en la selda [256,6] de la tabla "Histogramas3"
else if (val == anormN) % Se compara el valor introducido con el rango
    xlswrite('Histograma.xlsx', anormN, 'Hoja1', 'IW5');
    % Se inserta "SI" en la selda [256,6] de la tabla "Histogramas3"
else if (val == anormM) % Se compara el valor introducido con el rango
    xlswrite('Histograma.xlsx', anormM, 'Hoja1', 'IW5');
    % Se inserta "SI" en la selda [256,6] de la tabla "Histogramas3"
end
end
end

```

Código: Módulo Segmentar

```
% guardo el archivo de a ser clasificado en la variable "test"
test = get(handles.archivoARFF);

% guardo la ruta en la variable "c"
c = [pathname,filename]
% muestro la ruta del archivo a ser clasificado
set(handles.archivoARFF,'String',c);
% Update handles structure
guidata(hObject, handles);

function archivoARFF_Callback(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in deteccion.
function deteccion_Callback(hObject, eventdata, handles)

% Se hace la clasificación y se almacena el resultado en un
archivo .txt
!java -classpath weka.jar weka.classifiers.meta.Decorate
-W weka.classifiers.trees.LADTree -t Htraining.csv -T Histograma.csv
-p 5 > detect.txt
winopen ('detect.txt')
```

Detección de anomalías en mastografías digitales usando el proceso KDD

ORIGINALITY REPORT

4%

SIMILARITY INDEX

PRIMARY SOURCES

1	ceur-ws.org Internet	116 words — 1%
2	www.mathworks.com Internet	63 words — 1%
3	archivos.ujat.mx Internet	44 words — < 1%
4	www.archivos.ujat.mx Internet	33 words — < 1%
5	nozdr.ru Internet	27 words — < 1%
6	www.cic.ipn.mx Internet	25 words — < 1%
7	www.enelinternetdelascosas.org Internet	23 words — < 1%
8	msc.itsm.edu.mx Internet	20 words — < 1%
9	rdu.unc.edu.ar Internet	14 words — < 1%

10	erecursos.uacj.mx Internet	13 words — < 1%
11	tesis.ipn.mx Internet	13 words — < 1%
12	www.sepi.esimez.ipn.mx Internet	12 words — < 1%

EXCLUDE QUOTES ON
EXCLUDE BIBLIOGRAPHY ON

EXCLUDE SOURCES OFF
EXCLUDE MATCHES < 10 WORDS