Clasificadores para el reconocimiento automático de células blásticas en leucemias agudas linfoides y mieloides

Helena Molina Abril
Máster Universitario de Bioinformática *y* Bioestadística
Area
Caracterización y reconocimiento morfológico automático de células linfoides anormales y células blásticas para el soporte al diagnóstico de Linfomas y Leucemias agudas

Nombre Director (Consultor)
Edwin Santiago Alférez Baquero
Nombre Profesor/a responsable de la asignatura
Carles Ventura Royo y Jose Antonio Morán Moreno

Fecha Entrega
2-1-18

AGRADECIMIENTOS

Agradecimientos al laboratorio CORE del Hospital Clinic de Barcelona, y en particular a la Dra. Anna Merino, por el suministro de imágenes y apoyo clínico y diagnóstico. Agradecimientos también al Dr. Edwin Santiago Alférez Baquero, por el suministro de los datos y la dirección de este trabajo.

FICHA DEL TRABAJO FINAL

Título del trabajo:	*Clasificadores para el reconocimiento automático de células blásticas en leucemias agudas linfoides y mieloides*
Nombre del autor:	*Helena Molina Abril*
Nombre del consultor/a:	*Edwin Santiago Alférez Baquero*
Nombre del PRA:	*Carles Ventura Royo* *Jose Antonio Morán Moreno*
Fecha de entrega (mm/aaaa):	**01/2018**
Titulación:	*Máster* Universitario de *Bioinformática* y Bioestadística
Área del Trabajo Final:	*Caracterización y reconocimiento morfológico automático de células linfoides anormales y células blásticas para el soporte al diagnóstico de Linfomas y Leucemias agudas*
Idioma del trabajo:	*Castellano*
Palabras clave	*Leucemia aguda, aprendizaje automático, características morfológicas celulares*

Resumen del Trabajo (máximo 250 palabras): *Con la finalidad, contexto de aplicación, metodología, resultados y conclusiones del trabajo.*

La realización de pruebas morfológicas de análisis de sangre periférica sigue siendo un primer paso para el diagnóstico morfológico rápido, así como para la selección de técnicas adicionales y el seguimiento de los pacientes con enfermedades malignas de la sangre, incluyendo la leucemia aguda, en la que el tratamiento temprano es esencial para la mayor supervivencia de los pacientes. El análisis automático de imágenes de sangre periférica, se ha ido integrando en la rutina diaria de numerosos laboratorios clínicos. Sin embargo, por lo general estos sistemas, subestiman el número total de células blásticas, ya que éstas son fácilmente confundibles con linfocitos normales y reactivos. En este trabajo se pretende diferenciar de forma automática y objetiva entre distintos tipos de células, centrándonos especialmente en los linfocitos reactivos (infecciones) y las células blásticas (leucemias agudas). Se usará para ello una base de datos de características cuantitativas extraídas directamente de las imágenes digitales de muestras de frotis de sangre periférica, obtenidas en el laboratorio CORE del Hospital Clinic de Barcelona. Se aplicarán técnicas de aprendizaje automático para la generación de clasificadores, que serán evaluados a través del cálculo de distintas medidas de rendimiento.

Abstract (in English, 250 words or less):

Morphological tests of peripheral blood analysis is the first step in the process of fast morphological diagnosis of patients with malignant blood diseases, including acute leukemia, in which an early treatment is crucial for longer patient's survival. These tests are as well widely used for the selection of additional techniques and monitoring of such patients.

The automatic analysis of peripheral blood images has been integrated into the daily routine of numerous clinical laboratories. These automatic systems are greatly advantageous, since they allow to classify a large number of normal blood cells. However, these systems generally underestimate the total number of blast cells, since they are easily confused with normal and reactive lymphocytes.

In this work we aim to automatically and objectively differenciate between different types of cells, focussing our interest on reactive lymphocytes (infections) and blast cells (acute leukemias). A database of quantitative characteristics directly extracted from the digital images of peripheral blood smear samples, obtained in the CORE laboratory of the Hospital Clinic of Barcelona, will be used for this purpose. Machine learning techniques will be applied for the generation of classifiers, which will be evaluated through the calculation of different performance measures.

Índice

Lista de figuras

Lista de Tablas

1. Introducción

1.1 Contexto y justificación del Trabajo

Según la Organización Mundial de la Salud, se diagnostican entre 3 y 6 casos de leucemia aguda por cada 100.000 habitantes al año. En España, la leucemia infantil supone el 30% de todos los cánceres pediátricos, diagnosticándose cada año 300 nuevos casos, de los cuales más del 80% son leucemias agudas.

Las leucemias agudas son enfermedades de origen aún no aclarado que se caracterizan por la proliferación incontrolada de un determinado tipo de células inmaduras de la hematopoyesis. Estas células invaden la médula ósea, desplazando a las células normales y, progresivamente, al resto del organismo. El calificativo agudo define tanto la rápida velocidad de instauración de estas leucemias como la inmadurez de las células proliferantes, hechos que las diferencian de las leucemias crónicas en las que la instauración es más lenta y solapada, y las células proliferantes mucho más maduras. Sobre la base de sus características citológicas, poblaciones afectadas, tratamiento y pronóstico, clásicamente se distinguen dos grandes grupos de leucemias agudas:

- Leucemia mieloide aguda
- Leucemia aguda linfoblástica

En la Leucemia mieloide aguda las células de la línea mieloide (mieloblastos) proliferan de forma anormal invadiendo progresivamente la médula ósea, interfiriendo la producción de células normales de la sangre, lo que origina insuficiencia medular e infiltración de tejidos extra medulares.

En la Leucemia aguda linfoblástica, los linfoblastos (precursores de los linfocitos) se fabrican en cantidades excesivas y no maduran. Estos linfocitos inmaduros invaden la sangre, la médula ósea y los tejidos linfáticos, haciendo que se inflamen. También pueden invadir otros órganos, como los testículos o el sistema nervioso central.

Aunque en la actualidad el uso de test inmunológicos, citogenéticos y moleculares está cada vez más extendido, las realización de pruebas morfológicas de análisis de sangre periférica sigue siendo un primer paso para el diagnóstico morfológico rápido, así como para la selección de técnicas adicionales y el seguimiento de los pacientes con enfermedades malignas de la sangre, incluyendo la leucemia aguda, en la que el tratamiento temprano es esencial para la mayor supervivencia de los pacientes [1].

El análisis automático de imágenes de sangre periférica, se ha ido

integrando en la rutina diaria de numerosos laboratorios clínicos. Estos sistemas automáticos, han supuesto un gran avance, ya que permiten clasificar gran parte de células sanguíneas normales. Sin embargo, por lo general estos sistemas, subestiman el número total de células blásticas, ya que éstas son fácilmente confundibles con linfocitos normales y reactivos, siendo además incapaces de distinguir entre los linajes de células blásticas mieloides o linfoides. Por ejemplo, en [2] se cita la correcta clasificación de células blásticas en el sistema Cellavision DM96 al 76.6%.

En este trabajo, se pretende a través de técnicas de aprendizaje automático, diferenciar de forma automática y objetiva los distintos tipos de células, centrándonos especialmente en los linfocitos reactivos (infecciones) y las células blásticas (leucemias agudas). Se usará para ello una base de datos de características cuantitativas extraídas directamente de las imágenes digitales de muestras de frotis de sangre periférica, obtenidas en el laboratorio CORE del Hospital Clinic de Barcelona [3].

1.2 Objetivos del Trabajo

Objetivo general:

Proponer un sistema clasificador que permita, a partir de características cuantitativas extraídas de imágenes digitales de sangre periférica, diferenciar de forma automática y objetiva células blásticas frente a linfocitos reactivos.

Objetivos específicos:

- Implementar distintos métodos de preprocesamiento y selección de características que permitan, dentro de aquellas contempladas en el estudio, determinar las más apropiadas para ser empleadas en la clasificación.
- Implementar distintos métodos de clasificación supervisada que, haciendo uso de estas características, permitan determinar una mejor diferenciación entre células.
- Evaluar los distintos métodos de clasificación empleados, efectuando una comparativa entre ellos.

1.3 Enfoque y método sequido

Las técnicas de Machine Learning, o Aprendizaje Automático [4], constituyen una rama dentro del área de Inteligencia Artificial, que tiene como objetivo la creación de sistemas capaces de aprender por si sólos, sin ser programados de forma explicita, con la finalidad de realizar entre otras, tareas de clasificación o predicción de hechos futuros. Se trata por tanto de un conjunto de técnicas que tiene sentido aplicar al problema de estudio, dado que el objetivo final del trabajo será **clasificar correctamente**, dada una nueva muestra de sangre, el tipo de células detectadas (blástos o linfocitos reactivos).

Dentro de los métodos de Aprendizaje Automático que permiten resolver problemas de clasificación, nos encontramos con sitemas de clasificación **supervisados** y **no supervisados.** Los sistemas de **clasificación supervisados** son aquellos en los que, a partir de un conjunto de ejemplos de los que se conoce su clase (conjunto de entrenamiento), intentamos asignar una clasificación a un segundo conjunto de ejemplos de clase desconocida. En contra, los sistemas de **clasificación no supervisados** son aquellos en los que no se dispone de una batería de ejemplos previamente clasificados, sino que únicamente a partir de las propiedades de los ejemplos intentamos dar una **agrupación** (clasificación, clustering) de los ejemplos según su similaridad.

Dado que contamos con una base de datos inicial en la que se encuentran detalladas las características descriptivas de un conjunto de células, así como la clase a la que pertenecen, se optará por la utilización de algoritmos de clasificación supervisada que nos permitan clasificar como blastos o linfocitos reactivos futuras células de las cuales se desconozca a priori su tipo.

El diagrama a seguir para el desarrollo y utilización de esos métodos se muestra en la Figura 1.

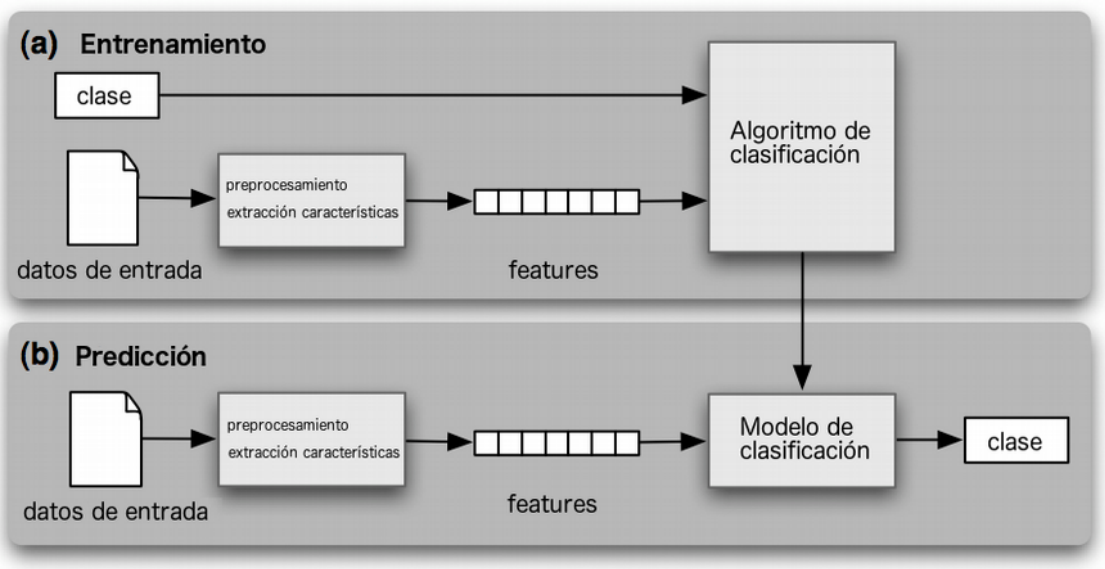

Figura 1: Entrenamiento y predicción en clasifiación supervisada

1.4 Planificación del Trabajo

A continuación se detalla el plan de trabajo, incluyendo las tareas a realizar, el calendario, los hitos a alcanzar y el análisis de riesgos.

1.4.1. Tareas

- Definición de los contenidos del trabajo

Estudio inicial del problema y definición inicial de los contenidos del trabajo a realizar.

- Definición del plan de trabajo

Se realizará una definición del plan de trabajo a llevar a cabo, hitos, y calendario de ejecución.

- Exploración inicial del conjunto de datos de estudio

Se analizará la base de datos inicial, mediante el uso de gráficos y descriptores que nos permitan hacernos una idea del conjunto de variables de estudio, y las relaciones entre ellas.

- Selección de los métodos de preprocesamiento

En esta tarea se sentarán las bases para seleccionar, en caso de ser necesarios, el conjunto de métodos de preprocesamiento que se aplicarán al conjunto de variables de estudio. Para ello, se realizará un análisis bibliográfico inicial de los distintos métodos existentes [5]. Se tendrán en cuenta tanto métodos de transformación de los datos iniciales, como métodos de reducción de dimensión (PCA, LDA, CCA, etc.).

- Selección de características

En esta tarea se sentarán las bases para determinar, en caso de ser necesarios, el conjunto de métodos de selección de características que se aplicarán al conjunto de variables de estudio. Para ello, se realizará un análisis bibliográfico inicial de los distintos métodos existentes [6]. Se explorarán tanto métodos de filtrado (correlación, chi-cuadrado, RELIEF, etc.) como métodos empaquetados.

- Selección de métodos de clasificación

En esta tarea se investigarán los distintos métodos de clasificación que puedan ser más eficientes para la correcta clasificación de los datos de entrada [7]. Se explorarán métodos basados en Máquinas de Soporte de Vectores, Redes Neuronales, Árboles Aleatorios, Vecinos cercanos, Naive Bayes, etc.

- Implementación de los distintos métodos seleccionados.

Se implementarán los métodos de clasificación, con su correspondiente preprocesamiento y selección de características previo. Se usará para ello el lenguaje de programación R a través de Rstudio [8].

- Comparación y evaluación de los distintos métodos

Se evaluarán los distintos métodos implementados, y se llevará a cabo una comparativa entre los mismos, utilizando para ello medidas de sensibilidad, especificidad y área bajo la curva.

- Elaboración del informe final

Se llevará a cabo la redacción del informe final incluyendo todos los resultados obtenidos.

- Elaboración de la presentación y defensa pública

Se llevará a cabo la elaboración de la presentación que servirá de soporte para la exposición pública del trabajo realizado.

- Difusión

Se estudiará la posibilidad de difundir los resultados obtenidos mediante la publicación en revistas o congresos científicos.

1.4.2. Calendario

El trabajo final de máster tiene una duración de aproximadamente 300 horas. Con objeto de conseguir los objetivos marcados anteriormente, se llevarán a cabo una serie de tareas, así como una planificación temporal (ver Figura 2). El diagrama de Gantt correspondiente puede verse en la Figura 3.

	Nombre	Duracion	Inicio	Terminado	Predecesores	Nombres del Recurso
1	Definición de los contenidos del trabajo	9 days	20/09/17 8:00	2/10/17 17:00		PEC0
2	Definición del plan de trabajo	10 days	3/10/17 8:00	16/10/17 17:00		PEC1
3	Exploración inicial del conjunto de datos de estudio	4 days	17/10/17 8:00	20/10/17 17:00		
4	Selección de los métodos de preprocesamiento	10 days	22/10/17 8:00	3/11/17 17:00		
5	Selección de características	10 days	22/10/17 17:00	3/11/17 17:00		
6	Selección de métodos de clasificación	12 days	3/11/17 8:00	20/11/17 17:00		PEC2
7	Implementación de los distintos métodos seleccionados.	15 days	21/11/17 8:00	11/12/17 17:00		
8	Comparación y evaluación de los distintos métodos	5 days	12/12/17 8:00	18/12/17 17:00		PEC3
9	Elaboración del informe final	11 days	19/12/17 8:00	2/01/18 17:00		PEC4
10	Elaboración de la presentación y defensa	14 days	3/01/18 8:00	22/01/18 17:00		PEC5a;PEC5b
11	Difusión	1 day	22/01/18 8:00	22/01/18 17:00		

Figura 2: Planificación de tareas

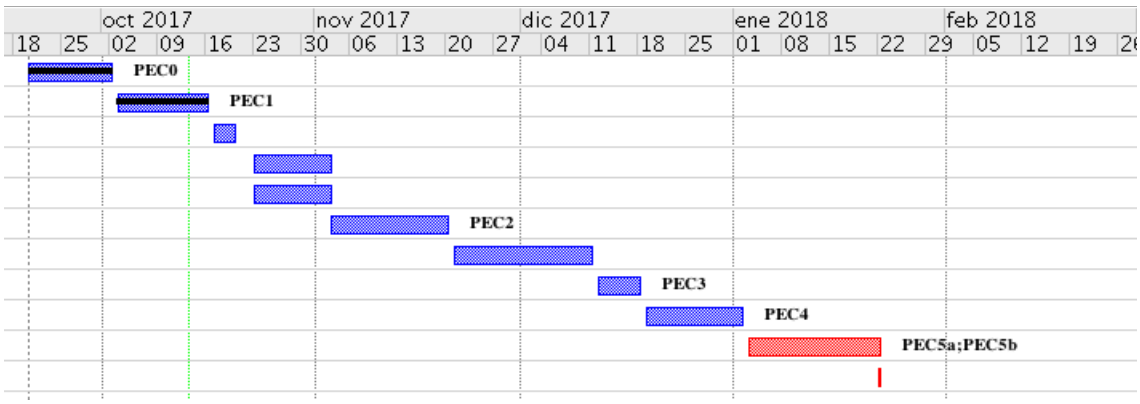

Figura 3: Diagrama de Gantt

A continuación se desglosa el número de horas a dedicar a cada tarea. Se han excluido los fines de semana (sábados y domingos), y se emplearán aproximadamente 3 horas diarias, de lunes a viernes en la ejecución del trabajo. No se han tenido en cuenta vacaciones o festivos ya que estos días, serán también empleados para la realización del trabajo.

- Definición del plan de trabajo: 27 horas distribuidas en 9 días.
- Definición del plan de trabajo: 30 horas distribuidas en 10 días.
- Exploración inicial de los datos: 15 horas distribuidas en 4 días.
- Selección de métodos de preprocesamiento: 30 horas distribuidas en 10 días.
- Selección de características: 30 horas distribuidas en 10 días.
- Selección de métodos de clasificación: 30 horas distribuidas en 10 días.
- Implementación: 45 horas distribuidas en 10 días.
- Comparativa: 15 horas distribuidas en 5 días.
- Elaboración informe: 35 horas distribuidas en 11 días.
- Elaboración de la presentación y defensa: 45 horas distribuidas en 14 días.

Se estiman un total de 302 horas de trabajo, más las empleadas en la posible difusión del mismo, que se considerarán ya fuera del Trabajo Fin de Máster.

1.4.3. Hitos

Los hitos se corresponden principalmente con las entregas programadas:

Hito	Entregable	Fecha
Inicio del trabajo		20 de septiembre de 2017
Entrega definición de contenidos	*PEC0*	2 de octubre de 2017
Entrega plan de trabajo	*PEC1*	16 de octubre de 2017
Revisión bibliográfica y selección de métodos finalizada	*PEC2*	20 de noviembre de 2017
Implementación de algoritmos de clasificación	*PEC3*	18 de diciembre de 2017

y comparativa finalizadas		
Entrega memoria	*PEC4*	2 de enero de 2018
Entrega presentación	*PEC5a*	10 de enero de 2018
Defensa y fin del trabajo	*PEC5b*	22 de enero de 2018

Tabla 1: Hitos del proyecto

1.4. Análisis de riesgos

A continuación se detallan las posibles incidencias y plan de contingencia:

Incidencia	Contingencia
Avería del puesto de trabajo	Se dispone de 1 puesto de trabajo adicional.
Corrupción o pérdida de datos	Todos los datos y scripts utilizados en el TFM se guardan en discos duros externos, así como en local. Igualmente, se mantendrá una copia actualizada en Github.
Enfermedad o accidente imprevisto	Se recuperarán las horas perdidas aumentando el número de horas de dedicación en fin de semana.
Desajuste entre planificación y carga real de trabajo	En la medida de lo posible se tratará de seguir la planificación. En caso de producirse un descuadre, se asignarán horas adicionales a las tareas con desfase.

Tabla 2: Incidencias y plan de contingencia

1.5 Breve sumario de productos obtenidos

Los productos obtenidos a la finalización del trabajo son:

- Plan de trabajo
- Memoria final
- Matriz de datos
- Script con el conjunto de clasificadores implementados, su evaluación y comparativa.
- Presentación virtual

1.6 Costes asociados a la realización del trabajo y producto final

La realización del presente trabajo tendrá lugar en el domicilio del autor, donde se dispone de todo el equipo necesario para la elaboración del mismo. Se usará para su desarrollo la versión 1.1.383 de Rstudio instalada en un equipo Ubuntu 16.04 LTS, siendo por tanto el coste de licencias nulo.

La implantación del producto final podrá llevarse a cabo sin coste asociado, siempre y cuando se disponga de un ordenador con capacidad suficiente para ejecutar el script de clasificación.

1.7 Breve descripción de los otros capítulos de la memoria

En el siguiente capítulo de esta memoria, Capítulo 2, se detalla la búsqueda bibliográfica realizada sobre tres de los aspectos fundamentales que compondrán el Trabajo Fin de Máster: Métodos de preprocesamiento de datos, métodos de selección de características, y métodos de clasificación dentro del aprendizaje automático supervisado. En el Capítulo 3 se detallan los resultados obtenidos tras la aplicación y evaluación de estos métodos, siendo seleccionados aquéllos que mejor rendimiento proporcionen para el conjunto de datos de estudio. El Capítulo 4 contiene las conclusiones finales del trabajo realizado. Los Capítulos 5 y 6 contienen el glosario de términos y la bibliografía respectivamente. Finalmente, el Capítulo 7 contiene un anexo con el código fuente implementado.

2. Materiales y métodos

2.1. Análisis exploratorio del conjunto de datos de estudio.

En este primer análisis exploratorio se pretende conocer las principales características de los datos de estudio, así como las relaciones entre ellos.

Para ello será necesario:

- Determinar las variables de entrada importantes para el análisis
- Conocer el tipo de dato de estas variables
- Conocer cómo se distribuyen los datos
- Identificar valores atípicos y faltantes
- Identificación de posibles patrones en los datos

Para llevar a cabo este análisis, se usará el software Rstudio instalado en una máquina Ubuntu Linux, con la versión de R 3.4.2. (2017-09-28).

A continuación se detallan las variables de interés dentro del conjunto de datos:

- *identidadnombre*:
  Variable categórica que indica el nombre de la entidad, es decir la enfermedad que padece el paciente. Puede tomar los valores:

  CLR: célula linfoide reactiva (o linfocitos reactivos), provienen de pacientes con algún tipo de infección vírica.
  LAL-B_BURKITT
  LAL-B_NO_BURKITT
  LAL-T
  LAM_MIELOIDE
  LAM_MIELOMONOCITICA
  LAM_MONOCITICA
  LAM_PROMIELOCITICA

  Donde LAL significa Leucemia aguda linfoide y LAM Leucemia aguda mieloide.

- *idtipocelulasbase:*
  Variable categórica que indica el tipo de célula. Puede tomar los valores:

  BLAST
  ATYPICAL_PROMYELOCYTE
  VARIANT_LYMPHOCYTE

  Donde BLAST indica que es una célula blástica o Blasto, ATYPICAL_PROMYELOCYTE es un promielocito atípico, precursor de los granulocitos, se puede considerar también como blasto (aunque está un nivel por debajo) y VARIANT_LYMPHOCYTE es el mismo linfocito reactivo.

- *idhistoria*:
  Variable numérica que indica el número de historia codificado, y sirve como identificador de paciente.

- *Fecha:*
  Variable categórica que indica la fecha en la cual el frotis de sangre periférica fue tomado y es un código a su vez para el frotis.

- *Hairiness, Area_cel, EquivDiameter_cel, Eccentricity_cel, Perimeter_cel, Solidity_cel, Extent_cel, circularity_cel, Elongation_cel, roundnessCH_cel, convexity_cel, circleVariance_cel, ellipVariance_cel, Area_nuc, EquivDiameter_nuc, Eccentricity_nuc, Perimeter_nuc, Solidity_nuc, Extent_nuc, circularity_nuc, Elongation_nuc, roundnessCH_nuc, convexity_nuc, circleVariance_nuc, ellipVariance_nuc, Ncytratio, Ncellratio, CentroidDist:*

  Variables numéricas que indican descriptores geométricos, que describen información acerca del tamaño y la forma de las regiones de interés: célula, núcleo, citoplasma y región externa a la célula.

- Resto de descriptores: descriptores de color y textura más complejos, entre ellos, descriptores de primer y segundo orden (Haralick features), descriptores Wavelet, descriptores Gabor, Local binary pattern (LBP) y descriptores granulométricos. Todos estos son aplicado a 6 espacios de color diferentes mediante 19 componentes de color diferentes: RGB, CMYK, XYZ, Lab, Luv, HSV. Los Gabor features son solo aplicados a RGB. Más información sobre la obtención de estos descriptores puede encontrarse en [9].

La Figura 4 muestra un ejemplo de células tipo blasto y linfocito reactivo, para las que han sido calculados todos estos parámetros:

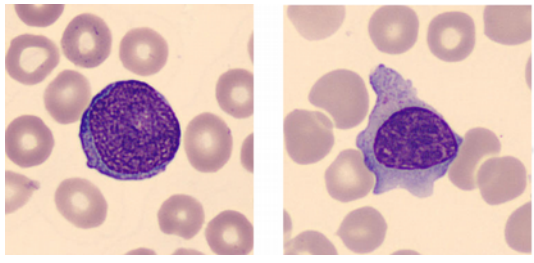

Figura 4: Blasto (izquierda) y Linfocito Reactivo(derecha)

El conjunto de datos inicial cuenta con 6918 observaciones de 2872 variables. Se puede observar que la muestra presenta 754 promielocitos atípicos, 7956 blastos, y 1208 linfocitos reactivos (ver Figura 5). Dado que la falta de balanceo entre clases puede causar problemas a la hora de intentar clasificar nuestros datos, habrá que tener en cuenta en el análisis posterior, que más del 70% de nuestras observaciones pertenecen a una misma clase (Blastos).

	frecuencia	porcentaje
ATYPICAL_PROMYELOCYTE	754	10.89910
BLAST	4956	71.63920
VARIANT_LYMPHOCYTE	1208	17.46169

Tabla 3: Frecuencia y porcentaje y de cada tipo de célula en el conjunto de datos

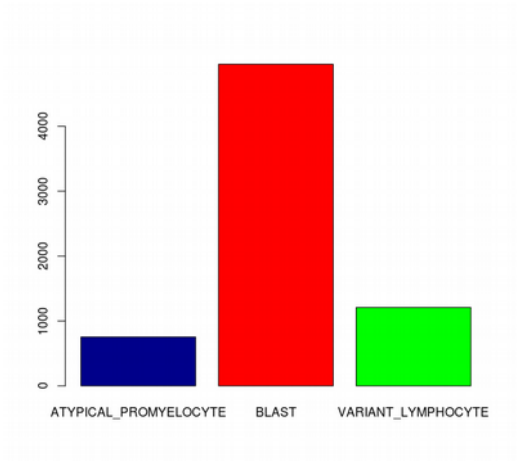

Figura 5: Diagrama de barras de frecuencias

En la Figura 6 observamos que no existen datos faltantes en nuestro conjunto de variables (mostrando sólo algunas de ellas a modo de ejemplo), por lo tanto, no será necesario por el momento que tratar con ellos.

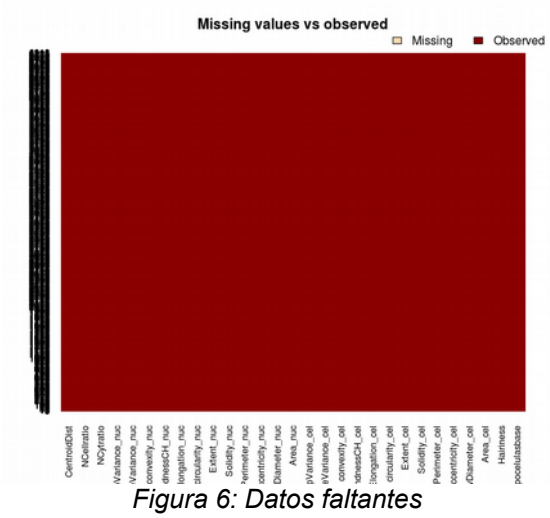

Figura 6: Datos faltantes

La Figura 7 muestra la matriz de correlación entre las distintas variables de estudio, teniendo sólo en cuenta los descriptores geométricos. El alto grado de correlación entre ciertos pares de variables (como *EquivDiameter_cel* y *Area_cel*, por ejemplo) tendrá que ser tenido en cuenta en el análisis, dado que ciertos clasificadores podrían verse afectados por la inclusión de información redundante entre las variables de estudio.

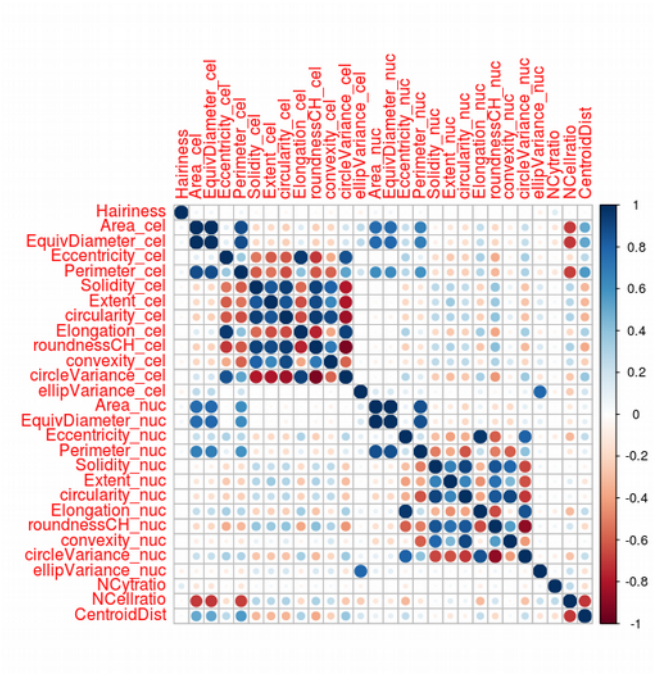

Figura 7: Matriz de correlación entre descriptores geométricos

En la Figura 8 se visualiza la distribución de algunas de las variables de estudio, mediante el uso de gráficos de densidad. La falta de normalidad de las distribuciones de algunas variables (como *hairiness, convexity_nuc*, etc.) tendrá que ser también tomada en cuenta a la hora de clasificar nuestros datos, ya que en ciertos tipos de métodos de clasificación, esta cuestión podría afectar. En la Figura 9 se muestran los diagramas de cajas de cada variable separados por clase. Algunas de las variables que se muestran muy diferentes para linfocitos reactivos que para el resto de tipos de células son por ejemplo *NcellRatio, CentroidDist* y *ellipvariance_cell.* Esto indica que estas variables podrían ser buenos predictores para clasificación.

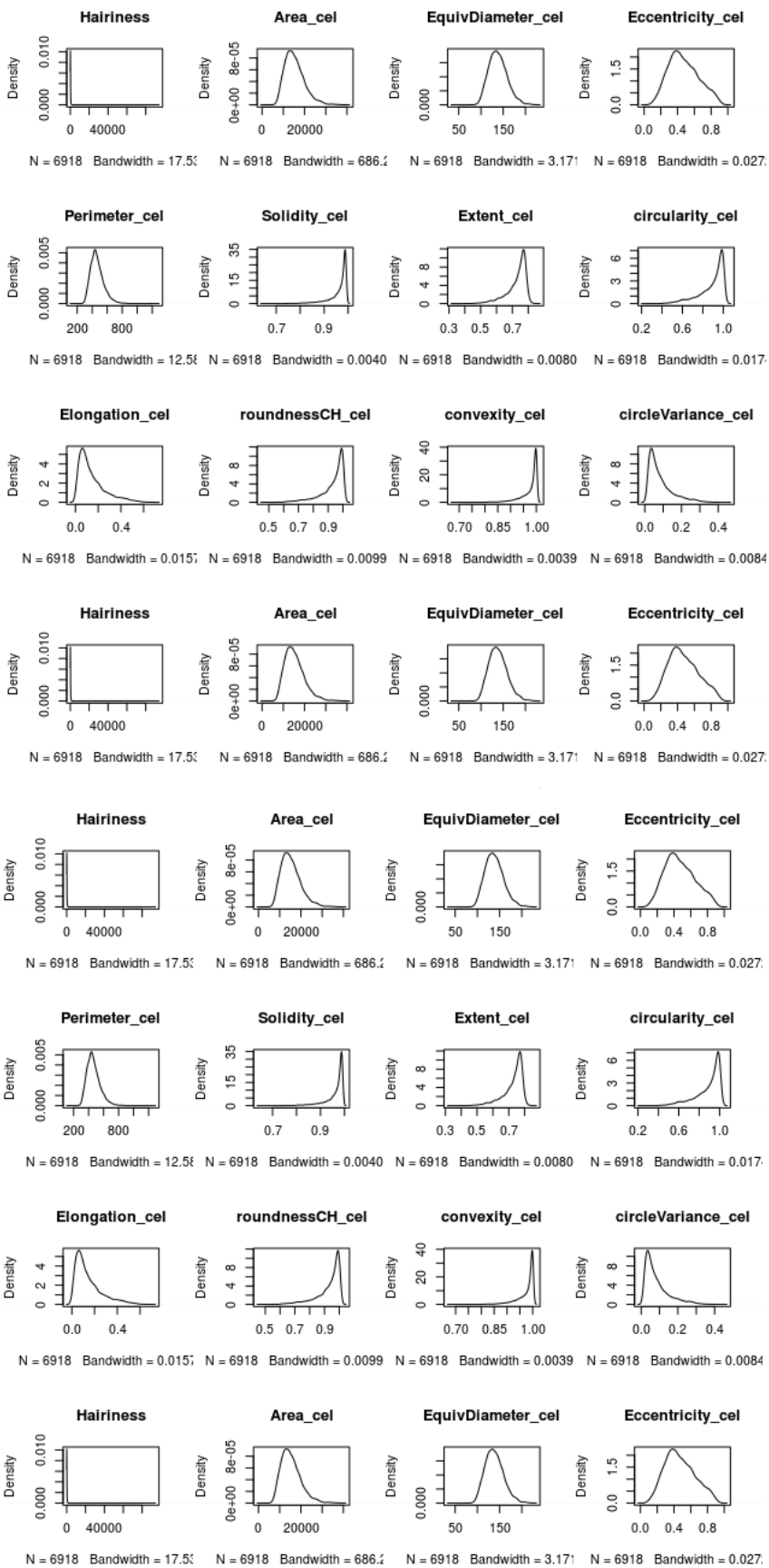

Figura 8: Diagramas de densidad de descriptores geométricos

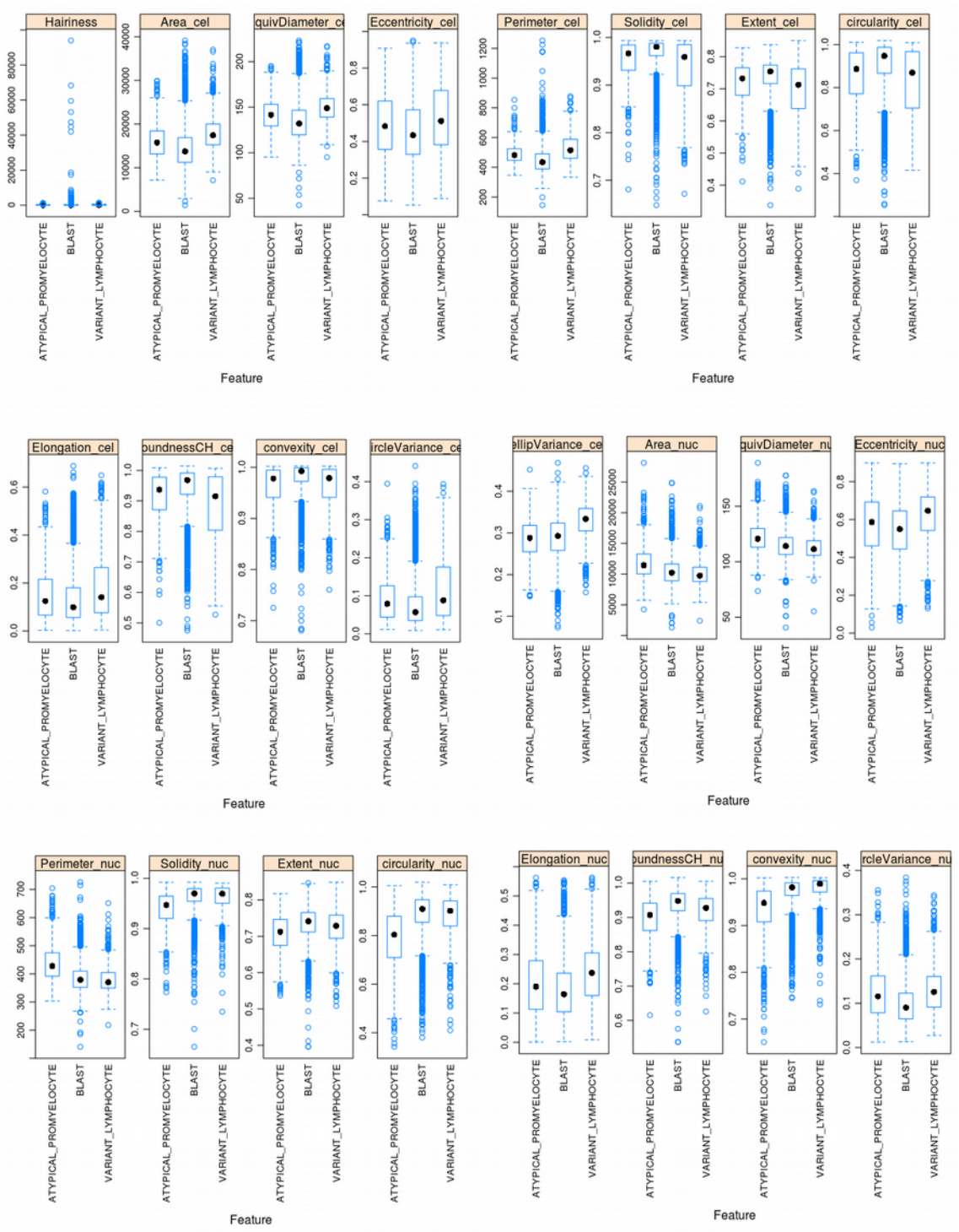

Figura 9: Diagramas de cajas por clases de los descriptores geométricos

2.1. Métodos de preprocesamiento y selección de características

Antes de comenzar con el análisis de datos y extracción de conocimiento, el conjunto de datos puede requerir cierto preprocesamiento dado que: los atributos pueden ser redundantes, pueden existir datos faltantes, pueden existir valores atípicos, los datos pueden no están en un formato adecuado, pueden aparecer inconsistencias, etc.

Por otro lado, en muchas ocasiones es también necesario seleccionar previamente los atributos o características que se utilizarán en el análisis, dado que: las características pueden estar correlacionadas y sesgar la predicción, pueden existir características irrelevantes que hagan aumentar el espacio de modelado innecesariamente, y puedan además hacer parecer ruido a las que sí lo son, etc.

En esta sección se presentan algunos de los métodos de preprocesamiento y selección de características más comúnmente utilizados, para su posterior uso en nuestro conjunto de datos de estudio.

Métodos de preprocesamiento

En ocasiones, es necesaria la preparación de los datos con el fin de obtener mejores resultados de los algoritmos de aprendizaje automático. Se llevará a cabo un repaso de los métodos disponibles, mediante el uso de la librería caret de R [10]. Se usará el software Rstudio instalado en una máquina Ubuntu Linux, con la versión de R 3.4.2. (2017-09-28).

A priori, es difícil saber qué métodos de preprocesamiento de datos usar, aunque existen determinadas “reglas” que suelen funcionar, como son por ejemplo; que los métodos basados en instancias (como el k-nearest neigbour) funcionan mejor si las variables de entrada tienen la misma escala; o que los métodos de regresión suelen funcionar mejor si las variables se encuentran estandarizadas, etc. Podríamos decir en rasgos generales que las transformaciones de datos más comunmente utilizadas tienen más probabilidades de ser útiles para algoritmos tales como algoritmos de regresión, métodos basados en instancias (como kNN), máquinas de soporte de vectores (SVM) y redes neuronales. Por el contrario, suelen resultar menos útiles para métodos basados en árboles y reglas.

En cualquier caso, este tipo de transformaciones ni siquiera en los métodos en los que suelen funcionar mejor, producen siempre mejores resultados que si utilizamos los datos brutos. Por eso, lo ideal es testear cada clase de transformaciones de datos, con cada clase de algoritmos de aprendizaje, y así poder determinar qué representación es la mejor para nuestros datos, y qué tipo de algoritmos explotan mejor la estructura de dichas representaciones.

A continuación se revisarán algunos de los métodos de transformación más comunes, y se aplicarán al conjunto de datos de estudio [11,12].

- **Escala**

La transformación de escala calcula la desviación estándar para un atributo y divide cada valor por esa desviación estándar.

$$x' = \frac{x}{\sigma} \tag{2.1}$$

- **Centrado**

La transformación de centrado calcula la media de un atributo y lo resta de cada valor.

$$x' = x - \bar{x} \tag{2.2}$$

- **Estandarización**

La combinación de las transformaciones de escalado y centrado estandarizará nuestros datos, dando lugar a variables con media igual a 0 y desviación estándar igual a 1.

$$x' = \frac{x - \bar{x}}{\sigma} \tag{2.3}$$

- **Normalización**

El escalado de los datos al rango [0, 1] se conoce como normalización.

$$x' = \frac{x - xmin}{xmax - xmin} \tag{2.4}$$

- **Transformación Box-Cox**

Cuando un atributo tiene una distribución similar a una Gaussiana pero desplazada, esto se denomina sesgo. La distribución de un atributo se puede desplazar para reducir el sesgo y hacerlo más gaussiano. La transformación de BoxCox puede realizar esta operación. En la Figura 10 se muestra el gráfico de densidad de la variable CentroidDist antes y después de esta transformación.

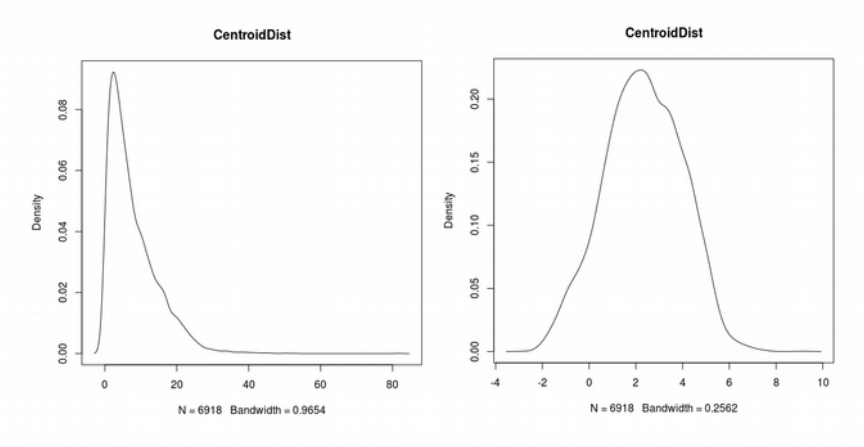

Figura 10: Variable antes y después de Box-Cox

- **Transformación Jeo-Johnson**

Esta transformación, similar a Box-Cox, admite valores brutos que son iguales a cero y negativos.

- **Análisis de componentes principales (PCA)**

Esta transformación mantiene los componentes por encima de un cierto umbral de varianza. El resultado son atributos que no están correlacionados entre sí, útiles para algoritmos como regresión lineal y generalizada. En el caso de nuestros datos, considerando únicamente los descriptores geométricos, son necesarias 10 componentes para capturar el 95% de la varianza. El dibujado de las dos primeras componentes, muestra una discreta separación entre las distintas clases de células (ver Figura 11).

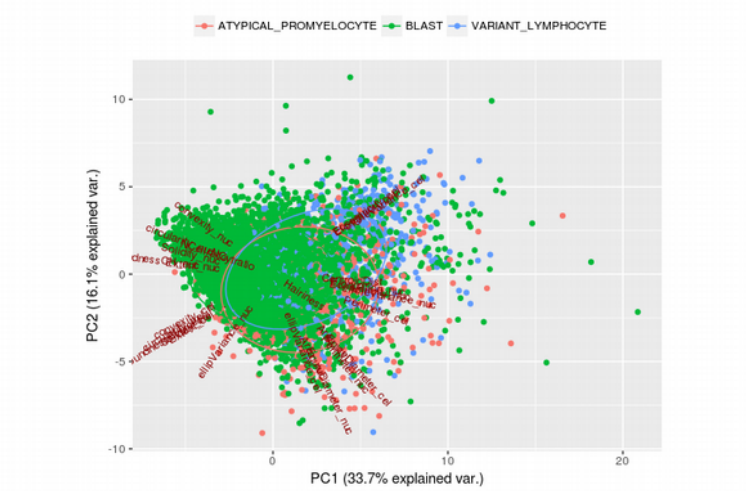

Figura 11: Dos primeros componentes principales de PCA

- **Análisis de componentes independientes (ICA)**

A diferencia de PCA, ICA conserva los componentes que son independientes. Se debe especificar la cantidad de componentes independientes deseados, y resulta útil para algoritmos de clasificación como naive bayes. En nuestros datos, considerando únicamente los descriptores geométricos y conservando por ejemplo 5 componentes, se obtiene cierta separación entre las clases (ver).

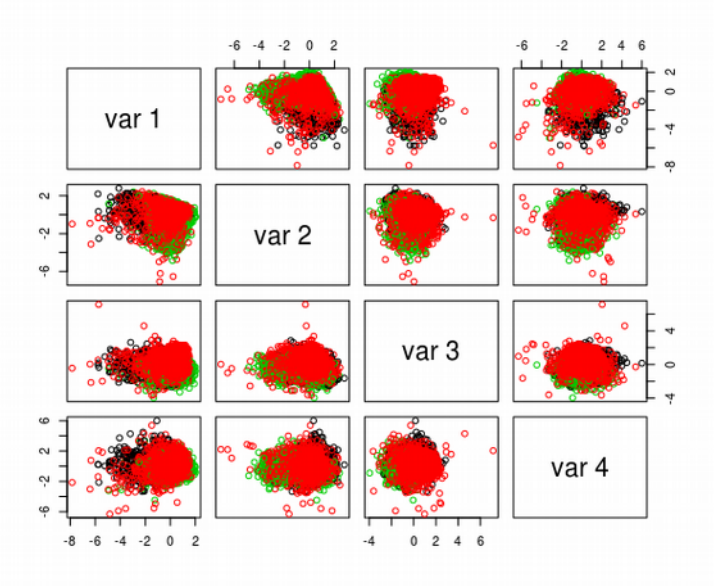

Figura 12: Resultado de ICA usando 5 componentes

Métodos de selección de características

Seleccionar las características correctas para los métodos de clasificación puede significar la diferencia entre un rendimiento pobre, frente a un rendimiento más eficiente. A continuación, presentamos los tipos más relevantes de métodos de selección de características [13]:

Métodos de filtrado

Los métodos de filtrado aplican un ranking sobre el conjunto de características de estudio. Este ranking indicará cómo de útiles podrán ser cada una de nuestras variables para la clasificación. A continuación se indican algunos de los más utilizados:

- **Coeficiente de correlación de Pearson**

  Uno de los problemas que pueden encontrar los métodos de clasificación al analizar nuestros datos, es que dentro del conjunto de variables, existan atributos que estén altamente correlacionados entre sí [14]. La eliminación de estos atributos, podría suponer un mejor rendimiento de nuestros clasificadores. La fórmula del cálculo de dicho coeficiente es:

$$\rho_{X,Y} = \frac{\sigma_{XY}}{\sigma_X \sigma_Y} = \frac{E[(X - \mu_X)(Y - \mu_Y)]}{\sigma_X \sigma_Y}, \tag{2.5}$$

Aplicando esta técnica a los descriptores geométricos de nuestro conjunto de datos, se obtiene que los atributos con correlación entre ellos mayor a 0.8 son:

```
Perimeter_cel
roundnessCH_nuc
roundnessCH_cel
circularity_cel
circleVariance_cel
circleVariance_nuc
Solidity_cel
EquivDiameter_cel
Perimeter_nuc
circularity_nuc
Elongation_cel
Elongation_nuc
EquivDiameter_nuc
```

- **Información mutua**

El método de información mutua permite determinar el grado de información compartida entre cada par de variables, así como medir en qué nivel el conocimiento de una de ellas permite reducir la incertidumbre sobre la otra [15]. Si hay mucha información compartida entre una de nuestras variables y la clase de nuestros datos, este es un buen indicador de que esta característica es importante y útil para distinguir a los miembros de una clase de otra. Se define la **información mutua** entre x_i e y_i como:

$$I(x_i; y_j) = \log \frac{P(x_i|y_j)}{P(x_i)} \tag{2.6}$$

Aplicada los descriptores geométricos de nuestro conjunto de datos obtenemos la lista ordenada de la Tabla 7: Distribución de clases obtenida tras la partición por paciente, donde los últimos candidatos son los que menos contribuyen a la clasificación según este criterio.

- **Relief**

Este método [16] se centra en las capacidades de separación de instancias, mediante la selección para cada una de ellas de las instancias más cercanas dentro de la misma clase, y fuera de ella. Esta selección se usa para generar una ponderación sobre cada característica, que se va actualizando iterativamente. Aplicado a los descriptores geométricos de nuestro conjunto de datos (para un número de instancias vecinas igual a 5) obtenemos los resultados que se muestran en la Tabla 8: Distribucion de clases en el test y training sets.

Variable	Valor	Variable	Valor
Eccentricity_cel	0.7852487	circularity_cel	0.7852487
circleVariance_cel	0.7852487	Eccentricity_nuc	0.7852487
circleVariance_nuc	0.7852487	NCytratio	0.7852487
circularity_nuc	0.7850483	Solidity_nuc	0.7844472
convexity_nuc	0.7816525	Extent_nuc	0.78357
Elongation_cel	0.7852487	convexity_cel	0.7821523
Elongation_nuc	0.7852487	Extent_cel	0.7821407
NCellratio	0.7852487	Perimeter_cel	0.7709454
roundnessCH_cel	0.7852487	Perimeter_nuc	0.7332379
roundnessCH_nuc	0.7852487	Area_cel	0.6739795
CentroidDist	0.7852487	EquivDiameter_cel	0.6739795
Solidity_cel	0.7852487	Area_nuc	0.5731643
ellipVariance_cel	0.7852487	EquivDiameter_nuc	0.5731643
ellipVariance_nuc	0.7852487	Hairiness	0.1251557

Tabla 4: Resultado del criterio de Información Mútua sobre descriptores geométricos

Variable	Importancia	Variable	Importancia
Hairiness	0.0004826807	EquivDiameter_nuc	0.0212945588
Area_cel	0.0219595384	Eccentricity_nuc	0.0173303913
EquivDiameter_cel	0.0234536543	Perimeter_nuc	0.0172953652
Eccentricity_cel	0.0104670338	Solidity_nuc	0.0027538445
Perimeter_cel	0.0101701037	Extent_nuc	0.0103263723
Solidity_cel	0.0007470577	circularity_nuc	0.0158732108
Extent_cel	0.0045360937	Elongation_nuc	0.0163022707
circularity_cel	-0.0019575518	roundnessCH_nuc	0.0060485500
Elongation_cel	0.0085319332	convexity_nuc	0.0167746279
roundnessCH_cel	-0.0009513237	circleVariance_nuc	0.0126435456
convexity_cel	-0.0056539769	ellipVariance_nuc	0.0389892785
circleVariance_cel	0.0041502482	NCytratio	0.0014593571
ellipVariance_cel	0.0483451073	NCellratio	0.0788467042
Area_nuc	0.0197877330	CentroidDist	0.0203873250

Tabla 5: Resultado del método Relief sobre descriptores geométricos

- **Métodos ensamblados**

  El concepto de técnicas de ensamblado [17] se puede aplicar a la selección de características mediante la creación de varios rankings, que se combinan para dar lugar a un ranking combinado mejorado.

Métodos empaquetados

Los métodos empaquetados (Wrapper) se conocen así porque envuelven un clasificador dentro de un algoritmo de selección de características [18]. De esta forma: Se elige un conjunto de características; se determina la eficacia de este conjunto; se realiza alguna perturbación sobre el conjunto original y se

vuelve a evaluar la eficacia del nuevo conjunto. El problema con este enfoque es que evaluar cada posible combinación conllevaría en la mayoría de los problemas reales, un muy elevado coste de ejecución. Es necesaria por tanto, la utilización de métodos heurísticos de búsqueda, que permitan determinar conjuntos óptimos de características. A continuación mostramos algunos de estos métodos.

- **Clasificación de características por su importancia**

  La importancia de las características se puede estimar a partir de los datos mediante la construcción de un modelo. Algunos métodos, como árboles de decisión, tienen un mecanismo incorporado para informar sobre la importancia de las variables. Para otros algoritmos, la importancia puede estimarse usando un análisis de curva ROC realizado para cada atributo. A modo de ejemplo, aplicamos el método Random Forest a los descriptores geométricos de nuestro conjunto de datos, y obtenemos la lista de importancia de nuestras variables (ver Tabla 6).

Variable	Importancia
NCytratio	434.19
NCellratio	414.45
ellipVariance_cel	179.60
Perimeter_nuc	164.53
CentroidDist	160.45
convexity_nuc	154.13
EquivDiameter_nuc	127.28
Area_nuc	123.39
circularity_nuc	117.83
Perimeter_cel	103.96
Solidity_cel	85.59
ellipVariance_nuc	81.08
Hairiness	79.15
roundnessCH_nuc	74.86
circleVariance_nuc	69.88
Solidity_nuc	68.97
Extent_cel	67.39
convexity_cel	63.99
roundnessCH_cel	62.01
circleVariance_cel	58.02

Tabla 6: Importancia según Random Forest sobre el conjunto de descriptores geométricos

- **Selección de características usando métodos de eliminación recursiva.**

  Los algoritmos de selección de características por eliminación recursiva funcionan de la siguiente forma: Primero, el algoritmo ajusta el modelo a todas las variables predictoras. Cada predictor se clasifica usando su importancia para el modelo. Sea S_i una secuencia de números

ordenados que son valores candidatos para el número de predictores a retener ($S_1 > S_2$, ...). En cada iteración de selección de características, se conservan los predictores S_i con mejor resultado de clasificación. El modelo se reajusta y se evalúa el rendimiento. Finalmente se determina el valor de S_i con el mejor rendimiento y los mejores predictores de S_i se usan para ajustarse al modelo final. La aplicación de este método a los descriptores geométricos de nuestro conjunto de datos nos devuelve la siguiente lista por orden de importancia: `NCellratio`, `NCytratio`, `convexity_nuc`, `CentroidDist`, `ellipVariance_cel`. En la Figura 13 podemos observar que los resultados de clasificación usando Random Forest se estabilizan aproximadamente a partir del uso de 10 precictores. Podemos además ver que los cinco primeros predictores seleccionados prácticamente coinciden con los seleccionados por el método aplicado anteriormente.

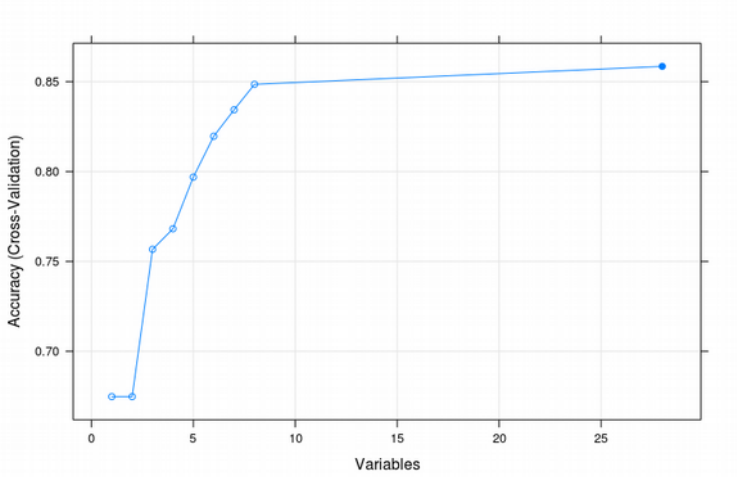

Figura 13: Resultados de clasificación RF y métodos de eliminación recursiva

- **Búsqueda genética**

Los algoritmos genéticos (GA) imitan a las fuerzas darwinianas de selección natural para encontrar valores óptimos de alguna función [19]. Se crea un conjunto inicial de soluciones candidatas y se calculan sus valores de aptitud correspondientes (donde los valores más altos son mejores). Este conjunto de soluciones se conoce como población y cada solución como individuo. Los individuos con los mejores valores de aptitud se combinan al azar para producir descendencias que constituyen la siguiente población. Para ello, los individuos son seleccionados y se cruzan entre sí (imitando la reproducción genética) además de estar sujetos a mutaciones aleatorias. Este proceso se repite una y otra vez en distintas iteraciones del procedimiento de búsqueda. Para la selección de características, los individuos son subconjuntos de predictores codificados como binarios; una característica está incluida o no en el subconjunto. Los valores de aptitud son una medida del rendimiento del modelo, como la exactitud de clasificación. El gran

inconveniente de estos métodos es la gran cantidad de recursos computacionales necesarios para su ejecución.

- **Recocido simulado**

  El recocido simulado (SA) es un método de búsqueda global que realiza pequeños cambios aleatorios (es decir, perturbaciones) en una solución candidata inicial [20]. Si el valor de rendimiento para el valor perturbado es mejor que la solución anterior, se acepta la nueva solución. De lo contrario, una probabilidad de aceptación se determina en función de la diferencia entre los dos valores de rendimiento y la iteración actual de búsqueda. Se puede aceptar una solución subóptima en función del cambio que eventualmente pueda producir una mejor solución en iteraciones posteriores. En el contexto de la selección de características, una solución es un vector binario que describe el subconjunto actual. El subconjunto se perturba al cambiar aleatoriamente un pequeño número de miembros en el subconjunto.

2.2. Métodos de clasificación.

El aprendizaje automático supervisado se centra en la búsqueda de algoritmos que a partir de datos suministrados externamente, permiten formular hipótesis generales que darán lugar a predicciones de clase sobre los datos futuros. De esta forma, los algoritmos de aprendizaje automático supervisado generan modelos concisos sobre la distribución de determinadas clases, que se usarán para predecir la clase a la que pertenecen nuevas instancias.

En nuestro caso de estudio, el objetivo de aplicar estos métodos se centrará en la creación de un modelo, que a partir de un conjunto de predictores (características geométricas, de textura, etc.) permita determinar el tipo de célula (clase) al que pertenecen. De esta forma, dada una muestra de sangre, se persigue clasificar de forma precisa si las células extraídas son de tipo blasto, o son linfocitos reactivos.

La elección de qué algoritmo de aprendizaje específico utilizar es un paso crítico en el análisis de datos. En la práctica habitual se utilizan distintas métricas que nos permiten evaluar el método que mejor resultados proporciona para nuestro conjunto de datos. Algunas de estas métricas son por ejemplo la exactitud (porcentaje de predicciones correctas entre el número total de predicciones), el área bajo la curva, etc. [21]. El uso de estas métricas, puede llevarse a cabo a través de numerosas técnicas. Una posible opción, consiste en dividir el conjunto de datos de entrada, entre un conjunto de entrenamiento y un conjunto de prueba. Una vez generado el modelo a partir de los datos de entrenamiento, calcular las métricas de evaluación en el subconjunto de prueba. Otra posible opción es la utilización de métodos como validación cruzada, en el que el conjunto de entrenamiento se divide en subconjuntos mutuamente excluyentes de igual tamaño, y para cada uno de ellos, el modelo

es construido en la unión del resto de subconjuntos. El promedio de la tasa de error de cada subconjunto es por lo tanto una estimación de la tasa de error del clasificador.

A continuación detallaremos algunos de los métodos de clasificación más comúnmente utilizados [22] que aplicaremos en una etapa futura del proyecto a nuestro conjunto de datos.

Algoritmos basados en lógica

En esta sección nos centraremos en dos grupos de métodos de aprendizaje lógico (simbólico): los árboles de decisión y los clasificadores basados en reglas:

- **Árboles de decisión**

  Los árboles de decisión son árboles que permiten clasificar instancias ordenándolas según el valor de sus predictores. Cada nodo en una árbol de decisión representa una característica en una instancia que ha de ser clasificada, y cada rama representa un valor que el nodo puede asumir. Las instancias comienzan clasificándose en el nodo raíz, y continúan a través del árbol según el valor de sus características. El problema de construir árboles de decisión binarios óptimos es un problema NP-completo. Se han desarrollado por tanto numerosos métodos en búsqueda de heurísticas eficientes que permitan la construcción de árboles de decisión. Uno de los algoritmos más conocidos para la construcción de árboles de decisión es el C4.5 [23]. Algunas de las ventajas de estos métodos son su robustez frente a valores a típicos, su escalabilidad, y su capacidad para modelar naturalmente límites de decisión no lineales gracias a su estructura jerárquica. Algunas de sus desventajas son que son propensos al sobre ajuste, aunque este puede ser aliviado mediante el uso de métodos “ensemble”, como bosques aleatorios.

- **Métodos basados en reglas**

  Los árboles de decisión se pueden traducir en un conjunto de reglas, de forma que se obtenga una regla para cada camino desde la raíz hasta una hoja del árbol. Sin embargo, las reglas también pueden ser inducidas directamente a partir de los datos de entrenamiento utilizando una variedad de algoritmos que se conocen como métodos basados en reglas [24]. Las reglas de clasificación representan cada clase en forma disyuntiva normal (DNF) de forma que durante la aplicación del método se construya el conjunto de reglas mínimo que es consistente con el conjunto de entrenamiento. Uno de los algoritmos más conocidos de este tipo es el algoritmo PART, que infiere reglas repetidamente mediante la generación de árboles de decisión parciales.

- **Bosques aleatorios**

Los algoritmos conocidos como bosques aleatorios (o Random forest) constituyen una extensión de los árboles de decisión que consiste en crear numerosos árboles de decisión independientes, construyéndolos a partir de datos de entrada ligeramente distintos. Para ello se seleccionan aleatoriamente con reemplazamiento un porcentaje de datos de la muestra total. Además, en cada nodo, al seleccionar la partición óptima, se tiene en cuenta únicamente una porción de los predictores, que son elegidos al azar en cada ocasión. Una vez construido el conjunto de árboles, en la fase de clasificación cada árbol se evalúa de forma independiente y la predicción del bosque se obtiene como la media del conjunto de árboles. La proporción de árboles que toman una misma respuesta se interpreta como la probabilidad de la misma [25]. Estos métodos extremadamente precisos, son para numerosos conjuntos de datos, los que mejor rendimiento proporcionan. Algunas de sus ventajas más importantes son: Su eficiencia, su capacidad para manejar un gran número de variables de entrada, proporcionan una estimación sobre la importancia de cada variable, generan una estimación interna no sesgada del error de generalización a medida que avanza la construcción del bosque, permiten estimar de forma robusta datos faltantes, mantienen una buena exactotid cuando faltan gran cantidad de datos, etc.

Algoritmos basados en Perceptrones

Otro tipo de algoritmos ampliamente utilizados en clasificación son los algoritmos basados en la noción de perceptrón [26].

- **Perceptrones de capa única**

  Se puede describir brevemente un perceptrón de una sola capa de la siguiente forma: Si $x_1 \ldots x_n$ son valores de características de entrada y $w_1 \ldots w_n$ un vector de pesos de conexión (típicamente números reales en el intervalo [-1, 1]), el perceptrón calcula la suma de las entradas ponderadas por los pesos, cuya salida pasa por un umbral ajustable: si la suma está por encima del umbral, la salida es 1; de lo contrario, es 0. La forma más común de que usar este método para clasificación consiste en ejecutar el algoritmo repetidamente a través del conjunto de entrenamiento hasta que se encuentre un vector de pesos que sea correcto en todo el conjunto de entrenamiento. Este vector, se usa posteriormente para predecir la clase del conjunto de prueba. El método WINNOW es un ejemplo de esta técnica [27].

- **Perceptrones multicapa**

  Los perceptrones de capa única, sólo permiten clasificar conjuntos linealmente separables de instancias. Si las instancias no son linealmente separables, el aprendizaje nunca llegará a un punto donde todas las instancias se clasifiquen correctamente. Los perceptrones multicapa, también conocidos como redes neuronales artificiales

(Artificial Neural Networks) se crearon para tratar de resolver este problema [28]. Las redes neuronales se componen de un elevado número de unidades (neuronas) que se conectan a través de un patrón de conexiones. Suelen existir tres tipos de neuronas: de entrada, de salida, e intermedias. En primer lugar, la red se entrena en un conjunto de datos emparejados que permiten determinar el mapeo entrada-salidas. Se calculan los pesos de las conexiones entre las neuronas y finalmente, se usa la red completa para determinar la clasificación de un nuevo conjunto de datos. Las ventajas de estos métodos son su facilidad de uso, la posibilidad de aproximar cualquier función, independientemente de su linealidad, y su buen rendimiento en problemas complejos. Por otro lado, requieren un gran número de instancias para su correcto funcionamiento, y un pequeño aumento de exactitud puede suponer un aumento de escala en varias magnitudes.

Métodos de aprendizaje estadístico

Los métodos estadísticos se caracterizan por tener un modelo de probabilidad explícito subyacente, que proporciona la probabilidad de que una determinada instancia pertenezca a cada clase. Algunos ejemplos de estos métodos son el Análisis Discriminante Lineal (LDA), el análisis Discriminante de Correspondencia [29], el método Naive Bayes y los clasificadores basados en redes bayesianas [30]. Una de las principales características de estos últimos, frente a los árboles de decisión o las redes neuronales, es la posibilidad de tener en cuenta información previa sobre un problema dado, en términos de relaciones estructurales entre sus características. Sin embargo, cuentan con el problema de que no son adecuados para conjuntos de datos con numerosas características [31]. En cuanto a Naive Bayes, sus ventajas son la simplicidad, facilidad de implementación y escalabilidad.

Máquinas de vectores de soporte

Las Máquinas de vectores de soporte (SVMs [32]) inducen separadores lineales o hiperplanos, ya sea en el espacio original de los ejemplos de entrada, si éstos son separables o cuasi-separables (ruido), o en un espacio transformado (espacio de características), si los ejemplos no son separables linealmente en el espacio original. La búsqueda del hiperplano de separación en estos espacios transformados, normalmente de muy alta dimensión, se hará de forma implícita utilizando las denominadas funciones kernel. Estas funciones resuelven el problema de clasificación trasladando los datos a un espacio donde el hiperplano solución es lineal y, por tanto, más sencillo de obtener. Una vez conseguido, la solución se transforma, de nuevo, al espacio original. Algunas ventajas de estos métodos son su capacidad para modelar límites no lineales, así como su robustez frente al sobre ajuste, especialmente en espacios de gran dimensión. Sus debilidades son la gran capacidad de memoria que necesitan, así como la necesidad de parametrización del kernel concreto a utilizar, además de su peor ajuste en conjuntos de gran tamaño de datos.

3. Análisis y resultados del conjunto de datos

A continuación se detallan los pasos seguidos en el análisis del conjunto de datos. Entre el conjunto total de pacientes, se ha creado una partición que contiene el 25% de pacientes para el conjunto de prueba, y el 75% para el conjunto de entrenamiento. La Tabla 7 muestra la distribución de clases obtenida tras esta partición es la siguiente (en número de pacientes):

	ATYPICAL_PROMYELOCYTE	BLAST	VARIANT_LYMPHOCYTE
Pacientes en training set	7	59	62
Pacientes en test set	2	19	20

Tabla 7: Distribución de clases obtenida tras la partición por paciente

Todas las células pertenecientes a Pacientes en el Taining set serán incluidas en el conjunto final de datos de entrenamiento, y todas aquéllas pertenecientes a pacientes incluidos en el test set, pasarán a formar parte del conjunto final de datos de prueba. De esta forma, la Tabla 8 y la Figura 14 muestran la distribución (en número de células) por clase:

	ATYPICAL_PROMYELOCYTE	BLAST	VARIANT_LYMPHOCYTE
Training set final	554	3697	848
Test set final	200	1259	360

Tabla 8: Distribucion de clases en el test y training sets

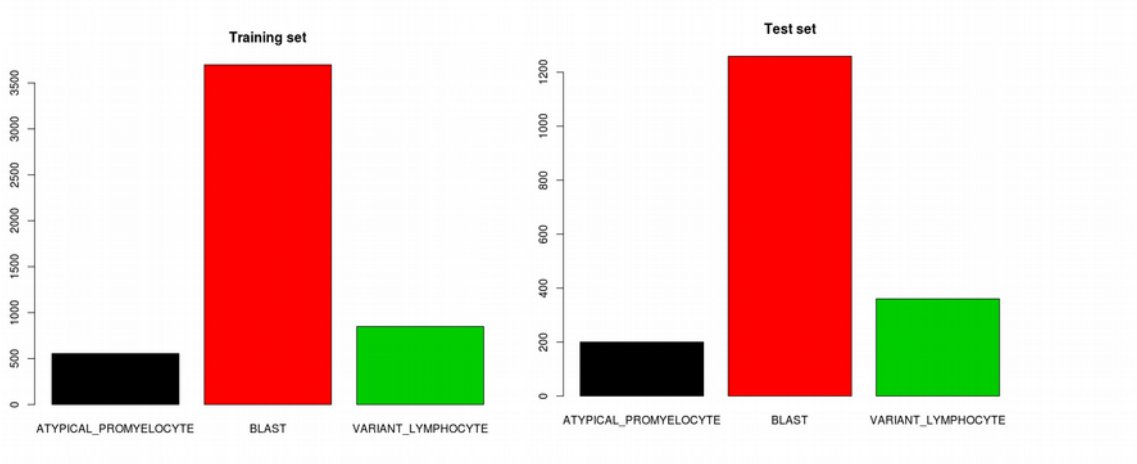

Figura 14: Diagrama de barras con la distribución de clases en training y test sets

La falta de balanceo entre clases podría provocar un mal rendimiento en ciertos clasificadores. Comprobamos el rendimiento de un clasificador por cada una de las clases principales que se detallaron en el capítulo anterior: Random Forest, Support Vector Machine, Linear Discriminant Analysis y Neural Networks. Para todos los métodos se han ejecutado dos tipos de preprocesamiento: Estandarización (CS) y Estandarización junto a PCA (CSPCA). El proceso de evaluación ha sido el siguiente en cada uno de los modelos generados: validación cruzada 5-fold para el modelado y evaluación sobre el conjunto de entrenamiento (training set) y evaluación sobre el conjunto de prueba (test set). Para la búsqueda de los parámetros de ajuste de cada uno de los métodos utilizados se ha utilizado una búsqueda aleatoria de longitud 5. Este método [33] resuelve el problema de optimización de hiperparámetros mediante una búsqueda aleatoria en el espacio de posibilidades, abordando el problema de forma distinta al método tradicional, que consiste en realizar la optimización de hiperparámetros mediante la búsqueda en cuadrículas. En la búsqueda en cuadrículas (grid search) se realiza una cerca exhaustiva a través de un subconjunto especificado manualmente del espacio de hiperparámetros.

Como medidas de evaluación se han usado:

- *La pérdida logarítmica* (logLoss), relacionada con la entropía cruzada, que mide el rendimiento de un modelo de clasificación donde la entrada de predicción es un valor de probabilidad entre 0 y 1. El objetivo de nuestros modelos de aprendizaje automático es minimizar este valor. Un modelo perfecto tendría una pérdida de 0.
- *Área bajo la curva* (AUC) mide el rendimiento de un clasificador mediante el área bajo la curva ROC, que se crea al trazar la fracción de verdaderos positivos frente a la fracción de falsos positivos. Un clasificador perfecto tendría un AUC de 1.
- *Exactitud* (Acc) es simplemente la proporción de instancias correctamente clasificadas. Esta medida puede llevar a confusión cuando el conjunto de clases no está balanceado. Un clasificador perfecto tendría una exactitud de 1.
- *Kappa* es una métrica que compara la exactitud observada con la exactitud esperada (posibilidad aleatoria), teniendo así en cuenta la probabilidad aleatoria de acierto (acuerdo con un clasificador aleatorio). Cuanto más cercana a 0, más parecido a un clasificador aleatorio será nuestro clasificador.
- *La sensibilidad* mide la proporción de positivos que se identifican correctamente como tales.
- *La especificidad* (también denominada tasa negativa verdadera) mide la proporción de negativos que se identifican correctamente como tales.

En los resultados de cada método empleado, aparecen además las siguientes medidas:

- El Valor Predictivo Positivo (VPP) es la probabilidad que un individuo realmente sea positivo cuando el resultado del modelo es positivo (considerando como positivo la pertenencia a una clase concreta).
- El Valor Predictivo Negativo (VPN) es la probabilidad que un individuo realmente sea negativo cuando el resultado del modelo es negativo (considerando como negativo la no pertenencia a una clase concreta).
- La prevalencia es la proporción de individuos que pertenecen a una clase concreta.
- El ratio de detección es a proporción de individuos de una clase concreta que son clasificados como tal.
- La prevalencia de detección es la prevalencia de las clases detectadas.
- La exactitud balanceada es la exactitud promedio obtenida en cada clase.

3.1. Resultados usando el conjunto completo de variables

3.1.1. Linear Discriminant Analysis

A continuación se exponen los resultados obtenidos por el método LDA incluido en la libería caret de R [10]. En primer lugar generamos el modelo mediante validación cruzada y evaluamos sobre el conjunto de entrenamiento. Los resultados obtenidos tras estandarizar los datos y tras además aplicar PCA se muestran en Tabla 9.

	logLoss	AUC	Acc	Kappa	Sensibilidad media	Especificidad media
LDA + CS	1.960645	0.9584263	0.9092024	0.7972869	0.8801	0.937924
LDA + CS + PCA	2.16614	0.9582711	0.9025294	0.7852353	0.8801089	0.9374419

Tabla 9: Resultados LDA + CS y LDA + CS + PCA

La Figura 15 muestra una comparativa sobre los dos tipos de preprocesamiento utilizados. Esta Figura (y el resto del mismo tipo que aparecen en la memoria) se ha realizado teniendo en cuenta todas las evaluaciones de la validación cruzada (5 folds), con un intervalo de confianza de 0.95, y mostrando un mini diagrama con los valores de las medidas del clasificador obtenidos. Como podemos observar, la estandarización ofrece ligeramente mejores resultados durante el entrenamiento que la estandarización junto con PCA.

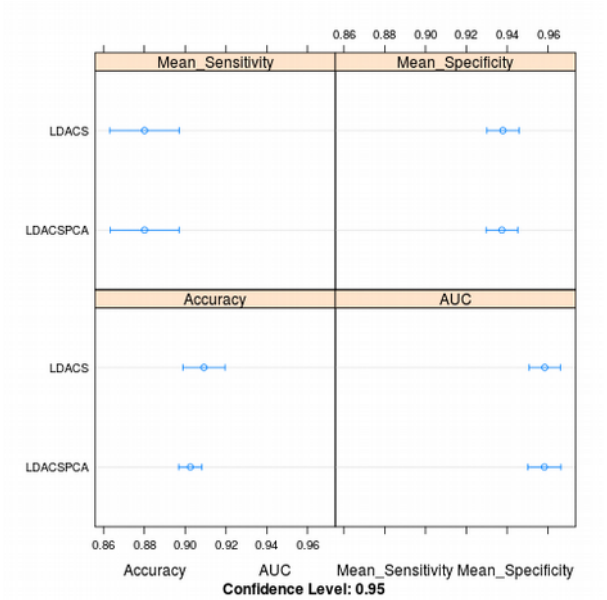

Figura 15: Comparativa LDA

Seleccionamos por tanto el método LDA con estandarización como preprocesamiento, y tras su evaluación sobre el conjunto de prueba, obtenemos unos valores de Exactitud de 0.9087 (0.8946-0.9216 95%CI) y Kappa de 0.8047. En la Tabla 10 podemos ver el número de instancias clasificadas correcta e incorrectamente. La Tabla 11 muestra los resultados de las medidas de evaluación por clase.

Referencia / Predicción	ATYPICAL_PROMYELOCYTE	BLAST	VARIANT_LYMPHOCYTE
ATYPICAL_PROMYELOCYTE	130	54	1
BLAST	67	1181	17
VARIANT_LYMPHOCYTE	3	24	342

Tabla 10: Resultados de clasificación LDA + CS en test set

	ATYPICAL_PROMYELOCYTE	BLAST	VARIANT_LYMPHOCYTE
Sensibilidad	0.65000	0.9380	0.9500
Especifididad	0.96603	0.8500	0.9815
Valor predictivo Positivo	0.70270	0.9336	0.9268
Valor predictivo Negativo	0.95716	0.8592	0.9876
Prevalencia	0.10995	0.6921	0.1979
Ratio de detección	0.07147	0.6493	0.1880
Prevalencia de detección	0.10170	0.6954	0.2029
Exactitud balanceada	0.80801	0.8940	0.9657

Tabla 11: Resultados de las medidas de evaluación LDA + CS en test set

La Figura 16 muestra las curvas ROC del método LDA + CS por cada clase.

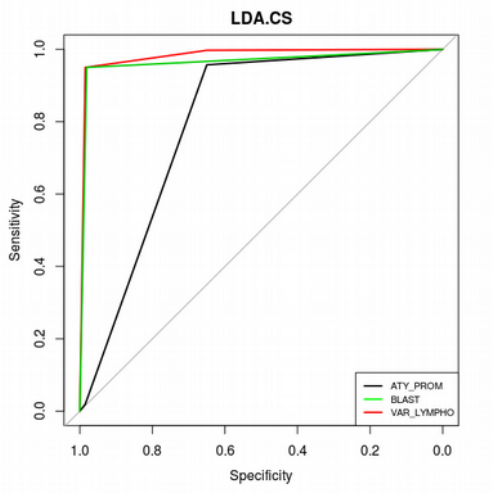

Figura 16: Curva ROC: LDA + CS

3.1.2. Árboles aleatorios (Random Forest)

A continuación se exponen los resultados obtenidos por el método ranger incluido en la libería caret de R. En primer lugar generamos el modelo mediante validación cruzada y evaluamos sobre el conjunto de entrenamiento. Los resultados obtenidos tras estandarizar los datos y tras además aplicar PCA se muestran en Tabla 12. Los parámetros seleccionados mediante búsqueda aleatoria para el algoritmo de árboles aleatorios son en ambos casos: `min.node.size = 20, mtry = 2811, splitrule = extratrees.`

	logLoss	AUC	Acc	Kappa	Sensibilidad media	Especificidad media
RF + CS	0.2171507	0.9870313	0.9458717	0.8676595	0.8712214	0.9388848
RF + CS + PCA	0.2171507	0.9870313	0.9458717	0.8676595	0.8712214	0.9388848

Tabla 12: Resultados RF + CS y RF + CS + PCA

La Figura 17 muestra una comparativa sobre los dos tipos de preprocesamiento utilizados. Como era de esperar, dada la robustez de los métodos de Árboles Aleatorios frente al estado de las variables de entrada, el proceso de estandarización no afecta a los resultados obtenidos por Random Forest.

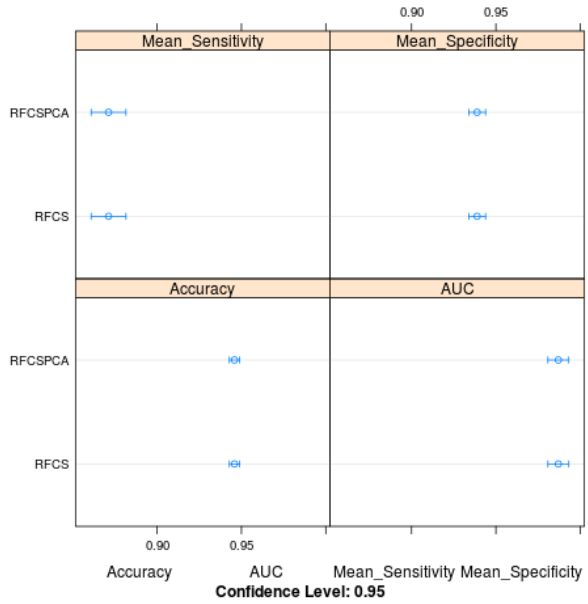

Figura 17: Comparativa RF

Seleccionamos el método RF con estandarización como preprocesamiento, y tras su evaluación sobre el conjunto de prueba, obtenemos unos valores de Exactitud de 0.89 (0.8748, 0.9041 95%CI) y Kappa de 0.7583. En la Tabla 13 podemos ver el número de instancias clasificadas correcta e incorrectamente. La Tabla 14 muestra los resultados de las medidas de evaluación por clase.

Referencia / Predicción	ATYPICAL_PROMYELOCYTE	BLAST	VARIANT_LYMPHOCYTE
ATYPICAL_PROMYELOCYTE	121	54	0
BLAST	79	1184	46
VARIANT_LYMPHOCYTE	0	21	314

Tabla 13: Resultados de clasificación RF + CS en test set

	ATYPICAL_PROMYELOCYTE	BLAST	VARIANT_LYMPHOCYTE
Sensibilidad	0.60500	0.9404	0.8722
Especifididad	0.96665	0.7768	0.9856
Valor predictivo Positivo	0.69143	0.9045	0.9373
Valor predictivo Negativo	0.95195	0.8529	0.9690
Prevalencia	0.10995	0.6921	0.1979
Ratio de detección	0.06652	0.6509	0.1726
Prevalencia de detección	0.09621	0.7196	0.1842
Exactitud balanceada	0.78582	0.8586	0.9289

Tabla 14: Resultados de las medidas de evaluación RF + CS en test set

La Figura 18 muestra las curvas ROC del método RF + CS por cada clase.

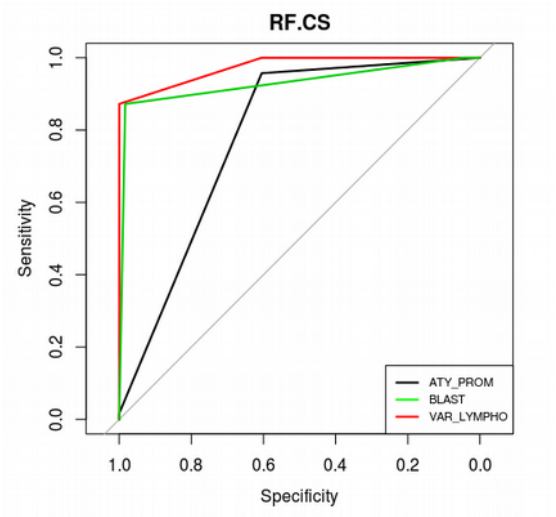

Figura 18: Curva ROC RF + CS

3.1.3. Máquinas de Soporte de Vectores (SVM)

A continuación se exponen los resultados obtenidos por el método *svmRadial* incluido en la libería caret de R. En primer lugar generamos el modelo mediante validación cruzada y evaluamos sobre el conjunto de entrenamiento. Los resultados obtenidos tras estandarizar los datos y tras además aplicar PCA se muestran en la Tabla 15 . Los parámetros obtenidos tras búsqueda aleatoria en el primero de los casos son: `sigma = 4.833605e-05` y `C = 190.5314`. Tras el segundo método de preprocesado: `sigma = 0.0001668998` y `C = 6.509542`.

	logLoss	AUC	Acc	Kappa	Sensibilidad media	Especificidad media
SVM + CS	0.09074702	0.9937252	0.9694075	0.9293009	0.9472806	0.9746038
SVM + CS + PCA	0.0771173	0.9957864	0.9735273	0.9385943	0.9499953	0.9767036

Tabla 15: Resultados SVM + CS y SVM + CS + PCA

La Figura 19 muestra una comparativa sobre los dos tipos de preprocesamiento utilizados. Como se puede observar, la estandarización junto con PCA ofrece ligeramente mejores resultados que la normalización.

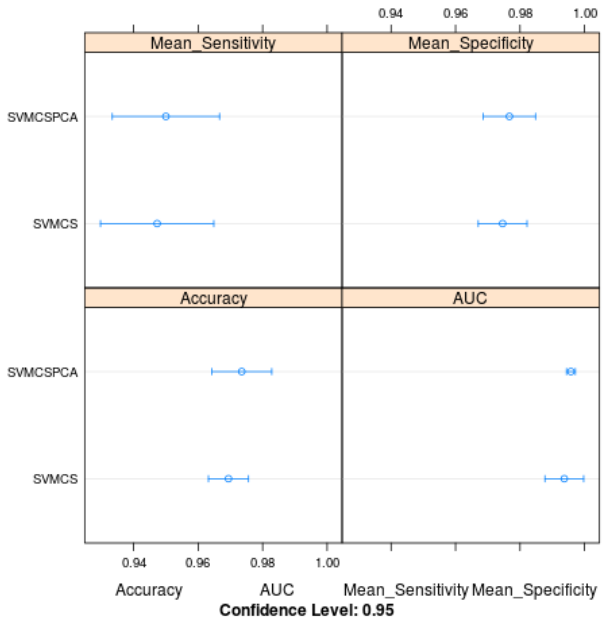

Figura 19: Comparativa SVM

Seleccionamos el método SVM con estandarización y PCA como preprocesamiento, y tras su evaluación sobre el conjunto de prueba, se obtienen unos valores de Exactitud de 0.9192 (0.9057, 0.9313 95%CI) y Kappa de 0.827. En la Tabla 16 se puede ver el número de instancias clasificadas correcta e incorrectamente. La Tabla 17 muestra los resultados de las medidas de evaluación por clase.

Referencia / Predicción	ATYPICAL_PROMYELOCYTE	BLAST	VARIANT_LYMPHOCYTE
ATYPICAL_PROMYELOCYTE	133	52	0
BLAST	67	1189	10
VARIANT_LYMPHOCYTE	0	18	350

Tabla 16: Resultados de clasificación SVM + CS + PCA en test set

	ATYPICAL_PROMYELOCYTE	BLAST	VARIANT_LYMPHOCYTE
Sensibilidad	0.66500	0.9444	0.9722
Especifididad	0.96788	0.8625	0.9877
Valor predictivo Positivo	0.71892	0.9392	0.9511
Valor predictivo Negativo	0.95900	0.8734	0.9931
Prevalencia	0.10995	0.6921	0.1979
Ratio de detección	0.07312	0.6537	0.1924
Prevalencia de detección	0.10170	0.6960	0.2023
Exactitud balanceada	0.81644	0.9035	0.9799

Tabla 17: Resultados de las medidas de evaluación SVM + CS + PCA en test set

La Figura 20 muestra las curvas ROC del método SVM + CS + PCA por cada clase.

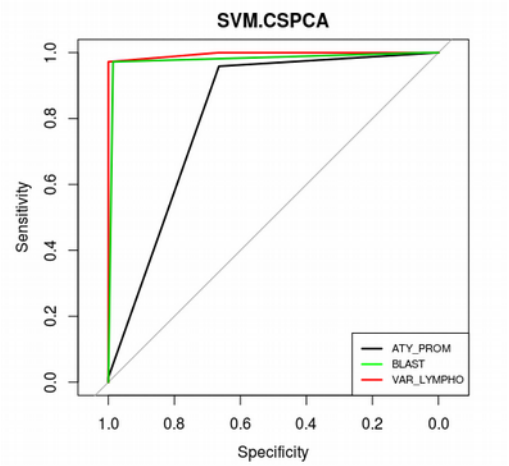

Figura 20: Curva ROC: SVM + CS + PCA

3.1.4. Redes Neuronales

A continuación se exponen los resultados obtenidos por el método nnet incluido en la libería caret de R. En primer lugar generamos el modelo mediante validación cruzada y evaluamos sobre el conjunto de entrenamiento. Los resultados obtenidos tras estandarizar los datos se muestran en la Tabla 18. Los parámetros obtenidos tras búsqueda aleatoria son: `size = 20` y decay = `7.608101`, donde *size* es el número de unidades en la capa oculta y *decay* es el parámetro de regularización que permite evitar el ajuste excesivo.

	logLoss	AUC	Acc	Kappa	Sensibilidad media	Especificidad media
NNET + CS	0.6613619	0.7076988	0.7499572	0.3032222	0.486353	0.7581267

Tabla 18: Resultados NNET + CS

El rendimiento del clasificador basado en redes neuronales se ve seriamente afectado por la falta de balanceo entre clases. Dado el buen rendimiento de los otros tres clasificadores testeados, descartamos este tipo de métodos para la evaluación final.

3.2. Resultados usando selección de características

En esta sección se muestran los resultados obtenidos con los métodos anteriores (RF, LDA y SVM), usando además el método de selección de características empaquetado conocido como Eliminación Recursiva (RFE). Este método consiste en utilizar algún clasificador para generar un modelo de datos y con éste generar un ranking de variables. Posteriormente se descartan un cierto número de las variables con menor puntuación en base a parámetros

de entrada del algoritmo y se repite la iteración con el nuevo subconjunto de variables.

	logLoss	AUC	Acc	Kappa	Sensibilidad media	Especificidad media
RFE + LDA	*2.1346*	2.1346	0.9037	0.7866	0.8771	0.9365
RFE + RF	*0.2426*	0.9852	0.9304	0.8261	0.8295	0.9208
RFE + SVM	*0.1006*	0.9932	0.9657	0.9207	0.9406	0.9717

Tabla 19: Resultados RFE

A pesar de la ligera mejoría en los resultados obtenidos frente a los resultados de la sección anterior, el algoritmo RFE devuelve en los tres casos que el mejor ajuste se consigue usando todas las variables del modelo, es decir, sin reducir el conjunto de características.

Sería necesario un análisis más profundo de los parámetros usados, así como la utilización de otros métodos de reducción, para obtener conclusiones más firmes a cerca de si la selección de características podría mejorar los resultados del clasificador. Sin embargo, en este trabajo hemos limitado esta experimentación, dado el elevado coste de computación de estos métodos.

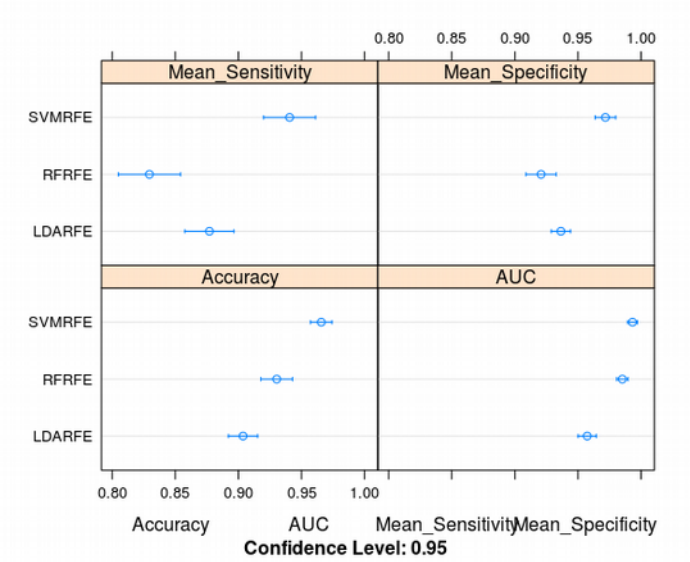

Figura 21: Resultados comparativa RFE

3.3.

3.3. Resultados usando métodos de balanceo

En esta sección se muestran los resultados obtenidos con el mejor método obtenido hasta el momento (SVM), usando además el método de balanceo de clases SMOTE. Este algoritmo conocido como Técnica de Muestreo de Minorías Sintéticas, muestrea instancias de la clase mayoritaria, y sintetiza nuevas instancias minoritarias mediante interpolación entre las existentes.

	logLoss	**AUC**	**Acc**	**Kappa**	**Sensibilidad media**	**Especificidad media**
SMOTE + SVM	*0.1113529*	0.9920512	0.9625415	0.9145552	0.9494832	0.9728178

Tabla 20: Resultados SMOTE

Podemos observar que usando este método, mejoran el número de instancias clasificadas en el conjunto de prueba, especialmente en la clase minoritaria (Promielocitos atípicos), así como las medidas de evaluación del clasificador en en esta clase.

Referencia / Predicción	ATYPICAL_PROMYELOCYTE	BLAST	VARIANT_LYMPHOCYTE
ATYPICAL_PROMYELOCYTE	153	77	0
BLAST	47	1161	9
VARIANT_LYMPHOCYTE	0	21	351

Tabla 21: Resultados de clasificación SVM + SMOTE en test set

	ATYPICAL_PROMYELOCYTE	BLAST	VARIANT_LYMPHOCYTE
Sensibilidad	0.76500	0.9222	0.9750
Especifididad	0.95244	0.9000	0.9856
Valor predictivo Positivo	0.66522	0.9540	0.9435
Valor predictivo Negativo	0.97042	0.8372	0.9938
Prevalencia	0.10995	0.6921	0.1979
Ratio de detección	0.08411	0.6383	0.1930
Prevalencia de detección	0.12644	0.6690	0.2045
Exactitud balanceada	0.85872	0.9111	0.9803

Tabla 22: Resultados de las medidas de evaluación SVM + SMOTe en test set

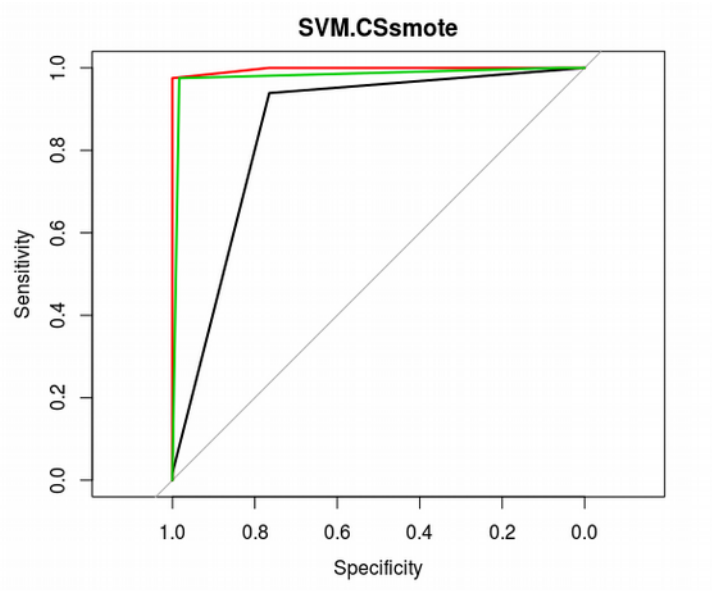

Figura 22: Curva ROC SVM + SMOTE

3.4. Comparativa final entre métodos

Dado que no existen grandes diferencias entre los métodos de normalización utilizados, y el rendimiento de los clasificadores es bueno para tres de los cuatro tipos de clasificadores empleados, se procederá a la comparativa final entre los distintos modelos. Seleccionamos para ello los modelos generados tras estandarización por Random Forest, SVM y LDA.

La Tabla 23 y la Figura 23 muestran que SVM (usando funciones de base radial en el Kernel) tiene un rendimiento superior al resto de clasificadores usados.

Accuracy							
	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	NA's
RF	0.9440628	0.9441724	0.9450980	0.9458717	0.9460255	0.9500000	0
SVM	0.9637610	0.9656863	0.9685967	0.9694075	0.9735034	0.9754902	0
LDA	0.9000979	0.9029412	0.9106968	0.9092024	0.9107843	0.9214917	0
AUC							
	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	NA's
RF	0.9784335	0.9874291	0.9887844	0.9870313	0.9894162	0.9910935	0
SVM	0.9853623	0.9943462	0.9949056	0.9937252	0.9964806	0.9975314	0
LDA	0.9506618	0.9548599	0.9589611	0.9584263	0.9606797	0.9669691	0
Kappa							
	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	NA's
RF	0.8619940	0.8627074	0.8672236	0.8676595	0.8678592	0.8785133	0
SVM	0.9153455	0.9209831	0.9278191	0.9293009	0.9381288	0.9442279	0
LDA	0.7767186	0.7847771	0.7981757	0.7972869	0.8028860	0.8238769	0

Tabla 23: Resultados de la comparativa entre RF, SVM y LDA

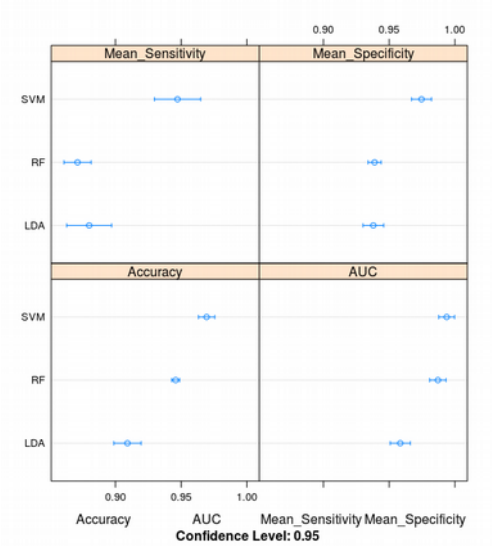

Figura 23: Comparativa SVM, RF y LDA

Como ha podido comprobarse en el apartado anterior, si consideramos además técnicas de balanceo de clases como SMOTE, el rendimiento mejora aún más, especialmente en la clasificación de las clases minoritarias.

4. Conclusiones

La principal motivación para la realización de este trabajo, ha sido profundizar en la aplicación de técnicas de aprendizaje automático para la resolución de problemas reales en el ámbito de la bioinformática, como lo es el problema de clasificación de células tratado en el presente trabajo. Tras el desarrollo, validación y comparación de los distintos clasificadores generados, se puede concluir que los objetivos propuestos en el plan de trabajo han sido alcanzados. Dado el gran número de variables que han sido analizadas, y el alto poder de computación necesario para generar los clasificadores, se produjo un pequeño retraso en una de las entregas, pero salvando éste contratiempo, todas las tareas planificadas han sido cumplimentadas con éxito.

Se han generado y evaluado cuatro tipos de clasificadores (SVM, RF, NNET y LDA) de los cuales tres han sido seleccionados para su posterior comparativa por su buen rendimiento. En cuanto a métodos de preprocesamiento y selección de atributos, se llevaron a cabo varias pruebas, concluyendo finalmente que una simple estandarización de los datos proporcionaba buenos resultados para los clasificadores empleados. Los tres modelos seleccionados para su posterior comparativa (SVM, RF, NNET) han sido testeados en un conjunto independiente de prueba, obteniendo en los tres casos una exactitud balanceada cercana al 0.8.

Se ha testeado además el método de selección de características por eliminación recursiva, no obteniendo mejoras significativas en la clasificación. Sería necesario un análisis más profundo de los parámetros usados, así como la utilización de otros métodos de reducción, para obtener conclusiones más firmes a cerca de si la selección de características podría mejorar los resultados del clasificador. Sin embargo, en este trabajo hemos limitado esta experimentación, dado el elevado coste de computación de estos métodos. Por último, se ha probado también el método de balanceo de clases SMOTE con SVM, que permite mejorar las medidas de evaluación del clasificador especialmente en las clases minoritarias.

Como trabajo futuro se planteará la posible publicación de los resultados obtenidos, así como su incorporación a una herramienta software que permita su fácil uso en la práctica real.

5. Glosario

Algoritmo
Una serie de pasos repetibles para llevar a cabo cierto tipo de tarea con datos.

Aprendizaje automático
El aprendizaje automático o aprendizaje de máquinas (del inglés, "Machine Learning") es el subcampo de las ciencias de la computación y una rama de la inteligencia artificial cuyo objetivo es desarrollar técnicas que permitan a las computadoras *aprender*. De forma más concreta, se trata de crear programas capaces de generalizar comportamientos a partir de una información suministrada en forma de ejemplos.

Aprendizaje supervisado
Aprendizaje supervisado es una técnica para deducir una función a partir de datos de entrenamiento. Los datos de entrenamiento consisten de pares de objetos (normalmente vectores): un componente del par son los datos de entrada y el otro, los resultados deseados, es decir, los resultados a los que debe llevarnos el modelo.

Aprendizaje no supervisado
El aprendizaje no supervisado es un método en el cuál modelo es ajustado a las observaciones. En este caso el algoritmo es entrenado usando un conjuntos de datos que no tiene ninguna etiqueta; nunca se le dice lo que representan los datos. La idea es que el algoritmo pueda encontrar por si solo patrones que ayuden a entender los datos.

Arboles de Decisión
Los Arboles de Decision son un algoritmo de Machine Learning que consisten en diagramas con construcciones lógicas, muy similares a los sistemas de predicción basados en reglas, que sirven para representar y categorizar una serie de condiciones que ocurren de forma sucesiva, para la resolución de un problema.

Atributos
Los Atributos son las propiedades individuales que se pueden medir de un fenómeno que se observa. La elección de atributos informativos, discriminatorios e independientes es un paso crucial para la eficacia de los algoritmos de Machine Learning.

Clasificación
En Machine Learning los problemas de **Clasificación** son aquellos en dónde el algoritmo de aprendizaje debe *clasificar* una serie de vectores en base a información de ejemplos previamente etiquetados. Es una caso típico del Aprendizaje supervisado

Conjunto de datos

Un Conjunto de datos o dataset es una colección de Datos que habitualmente están estructurados en forma tabular.

Datos
Un dato es una representación simbólica (numérica, alfabética, algorítmica, espacial, etc.) de un atributo o variable cuantitativa o cualitativa. Los datos describen hechos empíricos, sucesos y entidades. Es el elemento fundamental con el que trabaja la Ciencia de Datos.

Linear Discriminant Analysis (LDA)
·s una generalización del discriminante lineal de Fisher, un método utilizado en estadística, reconocimiento de patrones y aprendizaje automático para encontrar una combinación lineal de rasgos que caracterizan o separan dos o más clases de objetos o eventos. La combinación resultante puede ser utilizada como un clasificador lineal, o, más comúnmente, para la reducción de dimensiones antes de la posterior clasificación.

Modelo
En Machine Learning, un *modelo* es el objeto que va a representar la salida del algoritmo de aprendizaje. El *modelo* es lo que utilizamos para realizar las predicciones.

R
R es un lenguaje de programación interpretado diseñado específicamente para el análisis estadístico y la manipulación de datos. Junto con Python son los lenguajes más populares en Ciencia de Datos.

Random forest
También conocidos en castellano como «Bosques Aleatorios» es una combinación de árboles predictores tal que cada árbol depende de los valores de un vector aleatorio probado independientemente y con la misma distribución para cada uno de estos. Es una modificación sustancial de bagging que construye una larga colección de árboles no correlacionados y luego los promedia.

Red Neuronal
Las Redes Neuronales son un modelo computacional basado en un gran conjunto de unidades neuronales simples (neuronas artificiales), de forma aproximadamente análoga al comportamiento observado en los axones de las neuronas en los cerebros biológicos.

Support Vector Machine (SVM)
Las máquinas de vectores de soporte o SVM es un algoritmo de Machine Learning cuya idea central consiste en encontrar un plano que separe los grupos dentro de los datos de la mejor forma posible. Aquí, la separación significa que la elección del plano maximiza el margen entre los puntos más cercanos en el plano; éstos puntos se denominan vectores de soporte.

6. Bibliografía

[1] Bain, B.J. Leukaemia diagnosis. Oxford: John Wiley & Sons. 2010.

[2] Briggs, C., Longair, I., Slavik, M., et al. Can automated blood film analysis replace the manual differential? An evaluation of the CellaVision DM96 automated image analysis system. Int J Lab Hematol; 31:48–60, 2009.

[3] Bigorra,L., Merino, A., Alférez, S. and Rodellar, J. Feature Analysis and Automatic Identification of Leukemic Lineage Blast Cells and Reactive Lymphoid Cells from Peripheral Blood Cell Images. Journal of Clinical Laboratory Analysis 00: 1–9, 2016.

[4] Hastie, T., Tibshirani,R., Friedman, J.H. The Elements of Statistical Learning, Springer. ISBN 0-387-95284-5. 2001.

[5] Johnson, R.A. and Wichern, D.W. Applied Multivariate Statistical Analysis. Prentice Hall. 2007.

[6] Guyon, I., Elisseeff, A. An Introduction to Variable and Feature Selection. JMLR. 3. 2003.

[7] Mohri, M., Rostamizadeh, A., Talwalkar, A. Foundations of Machine Learning, The MIT Press ISBN 9780262018258. 2012.

[8] Ihaka, R. A Brief History R: Past and Future History. *http://cran.r-project.org/doc/html/interface98-paper/paper_2.html*.

[9] Puigví, L., Merino, A., Alférez S., Acevedo A., Rodellar J. New quantitative features for the morphological differentiation of abnormal lymphoid cell images from peripheral blood. J Clin Pathol. 70(12):1038-1048. 2017.

[10] https://topepo.github.io/caret/feature-selection-using-simulated-annealing.html

[11] Pyle, D,. *Data Preparation for Data Mining*. Morgan Kaufmann Publishers, Los Altos, California. 1999.

[12] Kotsiantis, S., Kanellopoulos, D., Pintelas, P,. *Data Preprocessing for Supervised Learning*. International Journal of Computer Science, 1 (2), 111–117. 2006.

[13] Guyon, I., Elisseeff, A.. *An introduction to variable and feature selection. Journal of Machine Learning Research*, 3 , 1157-1182, 2003.

[14] Pearson product-moment correlation coefficient. http://en.wikipedia.org/wiki/Pearson_ product_moment_correlation_coefficient

[15] Mutual information. http://en.wikipedia.org/ wiki/Mutual_information

[16] Kira, K, Rendell, L.A.. *A practical approach to feature selection*. Proceedings of the ninth international workshop on Machine learning, Derek Sleeman and Peter Edwards (Eds.). Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 249-256. 1992.

[17] Saeys, Y., Abeel, T., de Peer, Y. V.. *Robust feature selection using ensemble feature selection techniques*. Machine Learning And Knowledge Discovery In Databases, 2008.

[18] Kohavi, R., John, G.. *Wrappers for feature subset selection*. Artificial Intelligence, 1997.

[19] Mitchell, M.. *An Introduction to Genetic Algorithms*. MIT Press, Cambridge, MA, USA, 1998.

[20] Rutenbar, R.A.. *Simulated annealing algorithms: an overview.* IEEE Circuits and Devices Magazine, 5 (1), 19-26, 1989.

[21] Hand, D. J., *Measuring classifier performance: a coherent alternative to the area under the ROC curve*. Machine Learning. 77(1) 103–123, 2009.

[22] Kotsiantis, S. B., *Supervised Machine Learning: A Review of Classification Techniques.* Informatica. 31. 249-268, 2007.

[23] Quinlan, J. R., *C4.5: Programs for machine learning*. Morgan Kaufmann, San Francisco. 1993.

[24] Furnkranz, J., *Separate-and-Conquer Rule Learning*. Artificial Intelligence Review 13: 3-54. 1999.

[25] Breiman, L., *Random Forests.* Machine Learning **45** (1): 5–32. 2001

[26] Rosenblatt, F., *Principles of Neurodynamics*. Spartan, New York. 1962.

[27] Littlestone, N., Warmuth, M., The weighted majority algorithm. Information and Computation108 (2): 212–261, 1994

[28] Zhang, G., *Neural networks for classification: a survey.* IEEE Transactions on Systems, Man, and Cybernetics, Part C 30(4): 451-462, 2001

[29] Mika, S., Rätsch, G., Weston, J., Schölkopf, B., Müller, K.-R., *Fisher discriminant analysis with kernels.* In Y.-H. Hu, J. Larsen, E. Wilson, and S. Douglas, editors, Neural Networks for Signal Processing IX, pages 41-48. IEEE, 1999.

[30] Cestnik, B., Estimating probabilities: *A crucial task in machine learning*. In Proceedings of the European Conference on Artificial Intelligence, pages 147-149. 1990

[31] Cheng, J., Greiner, R., Kelly, J., Bell, D., & Liu, W., *Learning Bayesian networks from data: An information-theory based approach*. Artificial Intelligence 137: 43–90. 2002

[32] Cortes, C., Vapnik, V., *Support-vector networks.* Machine Learning. 20 (3): 273–297. 1995.

[33] Bergstra, J., Bengio, J.. *Random Search for Hyper-Parameter Optimization.* Journal of Machine Learning Research 13 (2012) 281-305.

7. Anexos

7.1. Modelo de script

(en este caso se ha usado LDA, pero sería similar para el resto de métodos)

```
#Añadimos las librerías necesarias
.libPaths( c( .libPaths(), "/home/habril/Rlibs") )
using<-function(...) {
    libs<-unlist(list(...))
    req<-unlist(lapply(libs,require,character.only=TRUE))
    need<-libs[req==FALSE]
    if(length(need)>0){
        install.packages(need,"/home/habril/Rlibs", repos='http://cran.us.r-project.org')
        lapply(need,require,character.only=TRUE)
    }
}

using("caret","devtools","mlbench","entropy","FSelector","doMC", "doParallel","ranger","MLmetrics","pROC")

#Definimos la función que contiene las métricas de resultados
fiveStats <- function(...) c(multiClassSummary(...), defaultSummary(...))

#Definimos el conjunto de semillas para la validación cruzada
set.seed(1234)
seedsRF <- vector(mode = "list", length = 6)
for(i in 1:6) seedsRF[[i]] <- sample.int(3000, 18)
seedsRF[[6]] <- sample.int(3000, 1)

#Leemos el conjunto de datos
filename="dataset_Helena.csv"
datasetRaw = read.csv(filename)
sum(is.na(datasetRaw))
str(datasetRaw)

#Datos que contienen una única célula por paciente
dataReduced<-datasetRaw[!duplicated(datasetRaw$idhistoria), ]
inTraining <- createDataPartition(dataReduced$idtipocelulasbase, p = .75, list = FALSE)

table(dataReduced[ inTraining,]$idtipocelulasbase)
table(dataReduced[-inTraining,]$idtipocelulasbase)

trainPatients <- dataReduced[ inTraining,]$idhistoria
testPatients <- dataReduced[ -inTraining,]$idhistoria

#Conjunto de datos final de prueba y entrenamiento
dataFullTrain<-subset(datasetRaw,datasetRaw$idhistoria %in% trainPatients)
dataTrain<-dataFullTrain[,                              !colnames(dataFullTrain)                          %in%
c("identidadnombre","idtipocelulasbase","idhistoria","fecha","archivos")]

dataFullTest<-subset(datasetRaw,!(datasetRaw$idhistoria %in% trainPatients))
dataTest<-dataFullTest[,                              !colnames(dataFullTest)                          %in%
c("identidadnombre","idtipocelulasbase","idhistoria","fecha","archivos")]

dataTrainClass<-dataFullTrain$idtipocelulasbase
dataTestClass<-dataFullTest$idtipocelulasbase

table(dataTrainClass)
table(dataTestClass)
plot(dataTrainClass,col=1:3,main="Training set")
plot(dataTestClass,col=1:3,main="Test set")

#LDA + CS
ctrlLDACS <- trainControl(method = "cv", number=5,
                          preProc = c("center","scale"),
                          savePredictions = TRUE,
                          verboseIter = TRUE,
                          summaryFunction = fiveStats,
                          classProb=TRUE,
                          allowParallel = TRUE,
                                              search="random",
                          seeds=seedsRF)

LDA.CS.model<- train(x=dataTrain,y=dataTrainClass,
                     method="lda",
                     trControl = ctrlLDACS,
                     tuneLength=5,
                     importance = "permutation",MaxNWts=60000)

save(LDA.CS.model, file="LDACSmodel.R")
```

```
#Resultados sobre el conjunto de entrenamiento
print(LDA.CS.model)
LDA.CS.model$results[rownames(LDA.CS.model$bestTune)[1],]
print(varImp(LDA.CS.model))

#Resultados sobre el conjunto de prueba
testPrediction.LDA.CS <- predict(LDA.CS.model,dataTest)
confusionMatrix(testPrediction.LDA.CS,dataTestClass)

#Curva ROC
testPrediction.LDA.CS.n <- as.numeric( predict(LDA.CS.model,dataTest,type="raw"))
roc.multi <- multiclass.roc(dataTestClass, testPrediction.LDA.CS.n)
auc(roc.multi)
rs <- roc.multi[['rocs']]
plot.roc(rs[[1]],main="LDA.CS")
sapply(2:length(rs),function(i) lines.roc(rs[[i]],col=i))
legend("bottomright", legend=c("ATY_PROM", "BLAST", "VAR_LYMPHO"),
       col=c("black", "green","red"), lwd=2,cex= 0.7)

#LDA + CS +  PCA
ctrlLDACSPCA <- trainControl(method = "cv", number=5,
                             preProc = c("center","scale","pca"),
                             savePredictions = TRUE,
                             verboseIter = TRUE,
                             summaryFunction = fiveStats,
                             classProb=TRUE,
                             allowParallel = TRUE,
                                                search="random",
                             seeds=seedsRF)

LDA.CSPCA.model<- train(x=dataTrain,y=dataTrainClass,
                        method="lda",
                        trControl = ctrlLDACSPCA,
                        tuneLength=5,
                        importance = "permutation",MaxNWts=60000)

save(LDA.CSPCA.model, file="LDACSPCAmodel.R")

#Resultados sobre el conjunto de entrenamiento
print(LDA.CSPCA.model)
LDA.CSPCA.model$results[rownames(LDA.CSPCA.model$bestTune)[1],]
print(varImp(LDA.CSPCA.model))

#Resultados sobre el conjunto de prueba
testPrediction.LDA.CSPCA <- predict(LDA.CSPCA.model,dataTest)
confusionMatrix(testPrediction.LDA.CSPCA,dataTestClass)

#Curva ROC
testPrediction.LDA.CSPCA.n <- as.numeric( predict(LDA.CSPCA.model,dataTest,type="raw"))
roc.multi <- multiclass.roc(dataTestClass, testPrediction.LDA.CSPCA.n)
auc(roc.multi)
rs <- roc.multi[['rocs']]
plot.roc(rs[[1]],main="LDA.CS.PCA")
sapply(2:length(rs),function(i) lines.roc(rs[[i]],col=i))
legend("bottomright", legend=c("ATY_PROM", "BLAST", "VAR_LYMPHO"),
       col=c("black", "green","red"), lwd=2,cex= 0.7)


#Comparativa
results <- resamples(list(LDACS=LDA.CS.model, LDACSPCA=LDA.CSPCA.model))
summary(results)
# boxplots
png("LDA-bw.png")
bwplot(results,metric=c("Accuracy","AUC","Mean_Sensitivity","Mean_Specificity"))
dev.off()
# dot plots
png("LDA-dt.png")
dotplot(results,metric=c("Accuracy","AUC","Mean_Sensitivity","Mean_Specificity"))
dev.off()
```