

Proyecto Fin de Carrera

Ingeniería de Telecomunicación

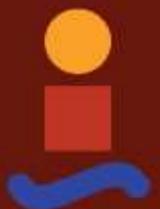
Aprendizaje Máquina para el diagnóstico de hemangiomas

Autor: José María Moreno Santa María Cantos

Tutoras: Begoña Acha Piñero y María del Carmen Serrano Gotarredona

Dpto. Teoría de la Señal y Comunicaciones
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2024



Proyecto Fin de Carrera
Ingeniería de Telecomunicación

Aprendizaje Máquina para el diagnóstico de hemangiomas

Autor:

José María Moreno Santa María Cantos

Tutoras:

Begoña Acha Piñero

Catedrática de Universidad

María del Carmen Serrano Gotarredona

Catedrática de Universidad

Dpto. de Teoría de la Señal y Comunicaciones

Escuela Técnica Superior de Ingeniería

Universidad de Sevilla

Sevilla, 2024

Proyecto Fin de Carrera: Aprendizaje Máquina para el diagnóstico de hemangiomas

Autor: José María Moreno Santa María Cantos

Tutoras: Begoña Acha Piñero
María del Carmen Serrano Gotarredona

El tribunal nombrado para juzgar el Proyecto arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2024

El Secretario del Tribunal

A mi familia

A mis maestros

A mis amigos

“Para algunos, la vida es

galopar un camino

empedrado de horas,

minutos y segundos.

Yo más humilde soy y

solo quiero que la ola

que surge del último suspiro

de un segundo me

transporte mecido

hasta el siguiente.”

-Robe Iniesta,

1998

Agradecimientos

En primer lugar, quiero expresar mi profundo agradecimiento por el apoyo incondicional de mi familia durante todos estos años de carrera, y por darme consejos para seguir adelante a pesar de las dificultades. Mi familia merece un lugar especial en esta dedicatoria, pues son el pilar fundamental de mi vida que me sustentaba cuando todo parecía venirse abajo. Gracias a vuestro calor, amor y apoyo incondicional, hoy puedo decir que soy Ingeniero. Os debo todo.

A mis amigos de toda la vida por las charlas infinitas quejándome de lo dura que era la carrera y de todas las veces que me han apoyado a seguir luchando. A mis amigos de la carrera, quien me iba a decir a mí que iba a llegar a una carrera en Sevilla y mis mejores amigos iban a ser 4 chavales de Huelva: Nono, Samu, Julio y Tino, vuestro apoyo en los primeros años en los que estábamos muy perdidos por la ETSI fueron esenciales al final hemos ido salido hacia delante y aquí estamos. A mis amigos de la rama de Imagen y Sonido: Román, Salado, Pedro, de la cual no conocía nada hasta mi tercer año y me ayudó muchísimo a engancharme a la carrera, a ellos les debo el hacerme más ameno estos últimos momentos de la carrera y siempre me tendrán para lo que les haga falta en la vida.

Y por último a mis profesoras de este TFG a Carmen y Begoña, si tenía que hacer este trabajo era con ellas, me han ayudado mucho a lo largo de la carrera y me han brindado todo tipo de facilidades para la realización de este TFG, gracias de verdad.

José María Moreno Santa María Cantos

Sevilla, 2024

Resumen

Un hemangioma infantil (IH, por su sigla en inglés) es una marca de nacimiento que se produce cuando un grupo de vasos sanguíneos y otras células no crecen normalmente. Al principio, la marca comienza con un pequeño bulto o parche de color en la piel y puede crecer rápidamente. Su color depende de qué tan profundo esté el IH dentro de la piel. A menudo, el IH no es visible al nacimiento. La marca empieza a aparecer en las primeras semanas de vida.

En el presente trabajo de fin de grado (TFG), se propone un método para la clasificación de la profundidad de hemangiomas infantiles mediante la extracción de características de color y textura de imágenes. La correcta clasificación de la profundidad es crucial para determinar el tratamiento adecuado y el pronóstico del IH.

El proceso de clasificación se basa en el análisis de imágenes utilizando técnicas avanzadas de procesamiento digital. Las características de color se extraen utilizando modelos de color como RGB, HSV y Lab*, que permiten captar las variaciones cromáticas asociadas a la profundidad de los hemangiomas. Por otro lado, las características de textura se obtienen mediante el uso de métodos como la matriz de co-ocurrencia de niveles de gris (GLCM), que analiza patrones de repetición y contraste en la imagen.

Estas características se integran en un modelo de aprendizaje automático que clasifica las imágenes de los hemangiomas en diferentes categorías de profundidad. Para entrenar y evaluar el modelo, se utilizan conjuntos de datos de imágenes anotadas por expertos médicos.

El estudio también incluye una revisión de la literatura sobre métodos previos de clasificación de IH y una comparación de los resultados obtenidos con nuestro enfoque. Los resultados preliminares muestran que la combinación de características de color y textura mejora la precisión de la clasificación en comparación con métodos que utilizan solo una de estas características.

Finalmente, se discuten las implicaciones clínicas de la implementación de este sistema, así como las posibles mejoras y futuras líneas de investigación. Este enfoque no solo proporciona una herramienta útil para los dermatólogos y pediatras, sino que también sienta las bases para el desarrollo de sistemas de diagnóstico asistido por computadora más avanzados en el campo de la dermatología pediátrica.

Abstract

An infantile hemangioma (IH) is a birthmark that occurs when a group of blood vessels and other cells do not grow normally. Initially, the mark begins as a small bump or colored patch on the skin and can grow rapidly. Its color depends on how deep the IH is within the skin. Often, the IH is not visible at birth. The mark starts to appear in the first few weeks of life.

In this final degree project (TFG), a method is proposed for classifying the depth of infantile hemangiomas through the extraction of color and texture features from images. Correct classification of the depth is crucial for determining the appropriate treatment and prognosis of the IH.

The classification process is based on image analysis using advanced digital processing techniques. Color features are extracted using color models such as RGB, HSV, and Lab*, which capture the chromatic variations associated with the depth of the hemangiomas. On the other hand, texture features are obtained using methods such as the gray-level co-occurrence matrix (GLCM), which analyzes patterns of repetition and contrast in the image.

These features are integrated into a machine learning model that classifies images of hemangiomas into different depth categories. To train and evaluate the model, datasets of images annotated by medical experts are used.

The study also includes a review of the literature on previous methods of IH classification and a comparison of the results obtained with our approach. Preliminary results show that combining color and texture features improves classification accuracy compared to methods that use only one of these features.

Finally, the clinical implications of implementing this system are discussed, as well as potential improvements and future lines of research. This approach not only provides a useful tool for dermatologists and pediatricians but also lays the foundation for the development of more advanced computer-aided diagnosis systems in the field of pediatric dermatology.

Agradecimientos	ixx
Resumen	xi
Abstract	xiii
Índice	xiv
Índice de Tablas	xvi
Índice de Figuras	xviii
1 Introducción	1
1.1. Contexto y justificación del estudio	1
1.2. Objetivos de la investigación	2
1.3. Relevancia clínica de la clasificación	2
2 Fundamentos Teóricos	4
2.1. Hemangiomas infantiles	4
2.1.1. Definición	4
2.1.2. Características	4
2.2. Clasificación de los hemangiomas infantiles por profundidad e importancia de la extracción de las características	5
2.3. Métodos de análisis de imágenes médicas en dermatología pediátrica	6
3 Extracción de Características de Color	9
3.1. Teoría del color en imágenes	9
3.2. Técnicas de extracción de características de color	10
3.2.1. Histogramas de color	10
3.2.2. Modelos de color(RGB,HSV,LAB)	11
3.3. Estudio clínico que muestra la importancia del color y la textura en el diagnóstico de hemangiomas infantiles	12
4 Extracción de Características de Textura	13
4.1. Conceptos básicos de textura en imágenes	13
4.2. Métodos de extracción de características de textura	14
4.3. Implementación de métodos de extracción de características de textura en imágenes de hemangiomas infantiles	15
5 Métodos de Clasificación	17
5.1. Algoritmos de clasificación comunes	17
5.1.1. K-Vecinos más cercanos (K-NN)	17
5.1.2. Máquinas de Soporte Vectorial (SVM)	18
5.1.3. Redes Neuronales Artificiales (ANN)	19
5.2. Entrenamiento y validación de modelos de clasificación	19
5.3. Evaluación de la precisión de la clasificación	20
6 Técnicas de selección de características	21
6.1. Algoritmos de selección comunes	21
6.2. Algoritmos de selección en este proyecto	23
6.2.1. Información Mutua (MI)	22

6.2.2.	Sequential Forward Selection (SFS)	23
6.2.3	Sequential Backward Selection (SBS)	24
6.2.4	T-test	25
6.3.	<i>Técnicas de reducción de dimensionalidad</i>	27
6.3.1.	Principal Component Analysis (PCA)	27
6.3.2.	T-distributed Stochastic Neighbor Embedding (t-SNE)	28
7	Estudios y Resultados Relevantes	30
7.1.	<i>Materiales</i>	30
7.2.	<i>Resultados obtenidos utilizando técnicas de extracción de características de color y textura</i>	31
7.2.1.	Métodos de selección características	31
7.2.2.	Resultados de la clasificación	34
8	Discusión y Conclusiones	40
	Referencias	42
	Glosario	44
	Anexo	46

ÍNDICE DE TABLAS

Tabla 7-1. Tabla de las características seleccionadas por cada método	33
Tabla 7-2. Tabla con todos los resultados de clasificación	38

ÍNDICE DE FIGURAS

Figura 2-1. Imagen de un hemangioma infantil.	4
Figura 2-2. Imagen de una dermatoscopia	6
Figura 2-3. Imágenes de una tomografía computarizada y su resultado.	7
Figura 2-4. Imagen ultrasónica: equipo y toma	7
Figura 2-5. Equipo de microscopia confocal	7
Figura 3-1. Gráfica comparativa de Skewness vs Kurtosis	10
Figura 3-2. Fórmula de Kurtosis y Skewness	11
Figura 3-3. Resumen de los modelos de color	11
Figura 5-1. Grafismo representativo de una K-NN	17
Figura 5-2. Grafismo representativo de una SVM	18
Figura 5-3. Grafismo representativo de una ANN	19
Figura 6-1. Fórmula de Información Mutua	23
Figura 6-2. Fórmula del t-Test	26
Figura 6-3. Ejemplo de cómo se vería una representación con PCA	28
Figura 6-4. Ejemplo de cómo se vería una representación con t-SNE	29
Figura 7-1. Visualización de la imagen original vs la recortada	31
Figura 7-2. Gráfica representativa de las características seleccionadas	34
Figura 7-3. Gráfica representativa del método IM sin entrenar vs entrenado	35
Figura 7-4. Gráfica representativa del método t-Test sin entrenar vs entrenado	36
Figura 7-5. Gráfica representativa del método SFS sin entrenar vs entrenado	37
Figura 7-6. Gráfica representativa del método SBS sin entrenar vs entrenado	38

1 INTRODUCCIÓN

El único modo de hacer un gran trabajo es amar lo que haces.

- Steve Jobs -

Hasta la fecha se han realizado varios estudios sobre el tema de los hemangiomas infantiles, el objetivo técnico de este escrito va a ser dar un paso más en esta investigación e intentar ayudar a través de este proyecto tanto a futuros investigadores como a futuros médicos que se especialicen sobre el tema para elaborar un mejor diagnóstico y así tener más probabilidad de acertar a la hora de realizar una primera exploración al paciente.

1.1. Contexto y Justificación del estudio

Los hemangiomas infantiles son lesiones vasculares benignas que afectan principalmente a los niños durante los primeros años de vida. Aunque la mayoría de los hemangiomas infantiles no requieren tratamiento, algunos pueden presentar complicaciones médicas, como úlceras, sangrado o, en casos raros, problemas de visión, respiración o función de órganos.

Para tomar decisiones clínicas adecuadas y planificar tratamientos específicos, es esencial una evaluación precisa de la profundidad del hemangioma, ya que esta característica influye en la gestión y el pronóstico.

Mejorar la precisión en el diagnóstico: La clasificación de la profundidad de los hemangiomas infantiles es un aspecto crítico para determinar el enfoque de tratamiento adecuado. La extracción de características de color y textura de imágenes médicas puede proporcionar una herramienta más precisa y objetiva para esta clasificación en comparación con la evaluación visual subjetiva.

Reducción de errores de diagnóstico: La interpretación visual de imágenes de hemangiomas infantiles puede ser subjetiva y depender de la experiencia del médico. La investigación en extracción de características puede ayudar a reducir los errores de diagnóstico y garantizar que los pacientes reciban la atención adecuada desde el principio.

Una de las motivaciones para llevar a cabo este estudio sería la teledermatología, es decir realizado de manera telemática, para hablar de este tema me baso en un estudio realizado durante 3 años analizando sus puntos fuertes y sus limitaciones. Se desarrolló un entorno completamente basado en la web para el soporte de teledermatología como resultado de un proceso de evaluación dinámica con personal clínico. Se realizaron un total de 120 teleconsultas (82 pediátricas y 28 para adultos) durante el estudio de concordancia clínica. Se realizó un análisis de concordancia para cada grupo de enfermedades dermatológicas.

Se encontraron altas tasas de concordancia en pediatría para dermatosis inflamatorias (76%) y también para adultos (75%) con infecciones e infestaciones. Los médicos estaban satisfechos con el sistema de teledermatología, pero el tiempo dedicado a la consulta en atención primaria fue un factor limitante (19 minutos para cada teleconsulta).

Una extensa discusión sobre los aspectos exitosos y limitantes de la experiencia de teledermatología reveló las

razones detrás de la decisión final de no proceder con su implementación. Se consideró que no estaba alineado con la estrategia de la Organización de Atención Médica (HCO, por sus siglas en inglés) y, por lo tanto, no obtuvo un alto nivel de apoyo para su implementación a largo plazo. [1]

1.2. Objetivo de la investigación

El objetivo principal de la investigación es el ayudar a los médicos y hacerle la vida más fácil a la hora de tratar con un paciente y ofrecerle un diagnóstico lo más preciso posible, esto lo podemos definir en los siguientes puntos:

Optimización de los recursos médicos: Al identificar con precisión la profundidad de los hemangiomas, se pueden evitar tratamientos innecesarios, como biopsias o intervenciones quirúrgicas, reduciendo así la carga tanto para los pacientes como para el sistema de atención médica.

Aplicación de tecnologías avanzadas: La tecnología de procesamiento de imágenes y análisis de datos ha avanzado significativamente en las últimas décadas. La aplicación de estas tecnologías en el campo de la dermatología pediátrica puede llevar a soluciones más eficaces y menos invasivas para el manejo de hemangiomas infantiles.

Contribución a la investigación médica: La investigación en la extracción de características de color y textura para la clasificación de hemangiomas infantiles puede abrir nuevas perspectivas en la comprensión de estas lesiones vasculares y su comportamiento clínico.

1.3. Relevancia clínica de la clasificación

Las anomalías vasculares representan un espectro de trastornos que van desde una simple "marca de nacimiento" hasta entidades que amenazan la vida. Los pacientes con estas anomalías suelen experimentar nomenclaturas incorrectas y diagnósticos erróneos.

Un diagnóstico preciso es crucial para una evaluación y manejo adecuados, y a menudo requiere de especialistas multidisciplinarios. Los esquemas de clasificación proporcionan una terminología coherente y sirven como guía para patólogos, clínicos e investigadores.

Uno de los objetivos de la Sociedad Internacional para el Estudio de Anomalías Vasculares (ISSVA) es lograr una clasificación uniforme. La última clasificación (1997) dividió las lesiones vasculares en malformaciones vasculares y lesiones vasculares proliferativas (tumores). Sin embargo, desde entonces se han identificado entidades adicionales que son complejas y menos fáciles de clasificar bajo encabezados genéricos, como malformación capilar, malformación venosa, malformación linfática, etc. [2]

En resumen, la clasificación es fundamental en el ámbito clínico porque:

- Proporciona a los médicos información crítica para tomar decisiones clínicas informadas. Ayuda a determinar si un hemangioma infantil en particular requiere intervención médica, cuál es el enfoque de tratamiento más adecuado y cuándo iniciar o detener el tratamiento.
- La clasificación ayuda a estratificar a los pacientes según el riesgo, lo que permite una atención más personalizada. Los hemangiomas superficiales pueden tener menos complicaciones que los profundos, y esto afecta la gestión clínica.
- Permite la optimización de los enfoques de tratamiento, lo que evita la administración innecesaria de medicamentos o procedimientos invasivos.
- Esto puede reducir la ansiedad y mejorar la comprensión de la situación clínica.
- Facilita la comunicación entre profesionales de la salud y permite un seguimiento coherente a lo largo del tiempo.
- Los datos de clasificación recopilados a lo largo de la investigación pueden contribuir a la comprensión más profunda de la epidemiología, el tratamiento y los resultados de los hemangiomas infantiles. Esto

puede conducir a avances en la investigación médica y a una mejora continua de la atención médica.

2 FUNDAMENTOS TEÓRICOS

La percepción, sin comprobación ni fundamento, no es garantía suficiente de verdad.

- Bertrand Russell

Aunque se han realizado investigaciones significativas en este campo, todavía existen preguntas sobre su origen exacto y su comportamiento. El tratamiento y la gestión de los hemangiomas infantiles dependen de múltiples factores y requieren un enfoque individualizado.

2.1. Hemangiomas infantiles

2.1.1. Definición

Los hemangiomas infantiles (HI) son tumores benignos producidos por las expansiones clonales de las células endoteliales. Se presentan en el 3% al 10% de los bebés, apareciendo al nacer o dentro de las primeras 4 o 6 semanas después del nacimiento. La evaluación de la extensión y volumen de los HI no es sencilla, ya que pueden presentar patrones irregulares o estar cubiertos por cabello. [3]



Figura 2-1. Imagen de un hemangioma infantil

2.1.2. Características

Los hemangiomas infantiles son tumores benignos que se desarrollan en los vasos sanguíneos y presentan las siguientes características:

Crecimiento rápido: Los hemangiomas infantiles tienden a crecer rápidamente durante los primeros meses de vida del bebé.

Aparición en la piel: Son comúnmente visibles en la piel y pueden tener una apariencia de color rojo o morado. Pueden variar en tamaño desde pequeñas manchas hasta crecimientos más grandes y prominentes.

Localización variable: Aunque suelen aparecer en la piel, los hemangiomas infantiles también pueden desarrollarse en otros lugares del cuerpo, como el hígado o el cerebro.

Fases de crecimiento e involución: Muchos hemangiomas infantiles pasan por fases de crecimiento rápido seguidas de una involución gradual. Algunos pueden desaparecer completamente con el tiempo, mientras que otros pueden dejar cicatrices o cambios en la piel.

Posibles complicaciones: Dependiendo de su ubicación y tamaño, los hemangiomas infantiles pueden causar complicaciones como ulceración (formación de heridas abiertas en la piel), sangrado, disfunción de órganos o deformidades físicas.

Cambio en el color: Con el tiempo, algunos hemangiomas infantiles pueden cambiar de color, volviéndose más pálidos o adquiriendo un tono azulado.

Más comunes en las niñas: Los hemangiomas infantiles son más comunes en las niñas que en los niños. [4]

Tratamiento: El tratamiento de los hemangiomas infantiles puede ser necesario en casos de complicaciones o cuando el crecimiento del tumor afecta la calidad de vida del niño. El propranolol es uno de los medicamentos utilizados para tratarlos, y la cirugía también puede ser una opción en algunos casos.

Es importante destacar que no todos los hemangiomas infantiles requieren tratamiento y que su evolución puede ser variable en cada caso. Por lo tanto, es fundamental que sean evaluados y supervisados por un profesional de la salud, como un pediatra o un dermatólogo, para determinar el enfoque adecuado.

2.2. Clasificación de los hemangiomas infantiles por profundidad e importancia de la extracción de las características

La profundidad a su vez se va a determinar por la percepción del color (en el espacio $L^*a^*b^*$), también nos afecta el matiz (hue) y el croma. Por eso vamos a necesitar las coordenadas de los $L^*a^*b^*$, del matiz y del croma. Luego necesitaremos una serie de descriptores que serán: luminosidad, la media del matiz, la media del croma, etc.

En la selección de características utilizamos dos métodos de selección, la selección secuencial hacia delante (SFS) y la selección secuencial hacia atrás (SBS). [5]

SFS es un procedimiento que va de abajo hacia arriba donde una característica es añadida a la vez, esto marca a su vez el rendimiento mínimo del error de clasificación comparándolo con la adición de una nueva característica. El algoritmo para cuando añadir una nueva característica incrementa el error de clasificación.

SBS es el procedimiento inverso, es decir de arriba hacia abajo empieza con el set completo de características y va descartando las características más discriminatorias. El algoritmo se para una vez se empieza a incrementar el error de clasificación.

En función de la profundidad, los hemangiomas pueden ser superficiales (en la dermis papilar) o profundos (en dermis reticular o tejido celular subcutáneo, en cuyo caso suelen presentarse como nódulos azulados mal delimitados, bajo una piel aparentemente normal). Las formas nodulares congénitas de localización subcutánea plantean problemas de diagnóstico diferencial y a menudo presentan características histológicas peculiares (“hemangiomas inmaduros”). [6]

La selección de rendimiento es evaluada por el validador XVAL (fivefold cross validation), así aprovechamos la desventaja en la sensibilidad que presenta SFS y SBS es disminuida, el procedimiento se repite cambiando los subconjuntos de validación de cada subconjunto de entrenamiento, así sucesivamente hasta que los 5 subconjuntos han sido usados como conjuntos de validación.

La extracción de características desempeña un papel esencial en el análisis médico al simplificar los datos, revelar patrones ocultos y mejorar la eficacia de los modelos y herramientas utilizadas en la atención médica, la investigación y el diagnóstico. Ayuda a los profesionales de la salud a tomar decisiones más informadas y precisas, lo que puede tener un impacto significativo en la atención al paciente y en la investigación médica.

La extracción de características nos es útil porque:

- Puede contribuir a la detección temprana de enfermedades y al diagnóstico preciso al resaltar características que son indicativas de una condición médica en sus primeras etapas.
- En la investigación médica puede ser crucial para identificar biomarcadores, que son características específicas que pueden utilizarse para diagnosticar o predecir enfermedades de manera precisa.
- Al identificar características específicas de un paciente, es posible adaptar las intervenciones médicas para obtener mejores resultados.
- Puede revelar patrones y correlaciones ocultas en los datos médicos. Esto es especialmente valioso para identificar relaciones entre variables que pueden estar relacionadas con enfermedades o condiciones médicas.

La selección adecuada de características puede simplificar la construcción y entrenamiento de modelos de machine learning y estadísticos. Los modelos con características relevantes tienden a ser más efectivos y generalizables.

La reducción de la dimensionalidad de los datos al seleccionar las características más relevantes o al crear nuevas características que resuman la información importante. Esto facilita la visualización y el análisis de los datos:

- Al reducir la cantidad de características en los datos, se puede mejorar la eficiencia computacional en tareas de análisis y modelado, lo que permite un procesamiento más rápido y menos demandante en recursos.
- Al centrarse en las características más significativas, los médicos y los investigadores pueden comprender mejor los factores clave que influyen en una condición o enfermedad.

2.3. Métodos de análisis de imágenes médicas en dermatología pediátrica

En dermatología pediátrica, los métodos de análisis de imágenes médicas son herramientas clave para el diagnóstico y la gestión de diversas afecciones cutáneas en niños. A continuación, se describen algunos de los métodos comunes de análisis de imágenes médicas utilizados en este campo:

Dermatoscopia: La dermatoscopia implica el uso de un dispositivo manual llamado dermatoscopio para examinar lesiones cutáneas a una magnificación mayor que a simple vista. La dermatoscopia asistida por computadora implica algoritmos automatizados para analizar imágenes dermatoscópicas, incluyendo reconocimiento de patrones y extracción de características para ayudar en el diagnóstico de lesiones cutáneas.



Figura 2-2. Imagen de una dermatoscopia

Fotografía Digital: Se capturan imágenes digitales de alta resolución para la documentación y análisis de lesiones cutáneas. Técnicas de procesamiento de imágenes, como la segmentación y extracción de características, pueden aplicarse para identificar y cuantificar características específicas de las lesiones cutáneas.

Tomografía Computarizada (TC) e Imágenes por Resonancia Magnética (IRM): Las exploraciones de TC e IRM proporcionan imágenes detalladas en secciones transversales de la piel y los tejidos subyacentes. Algoritmos automatizados de segmentación pueden aplicarse para delinear capas de la piel o identificar anomalías. La registración de imágenes también puede utilizarse para realizar un seguimiento de cambios a lo largo del tiempo.

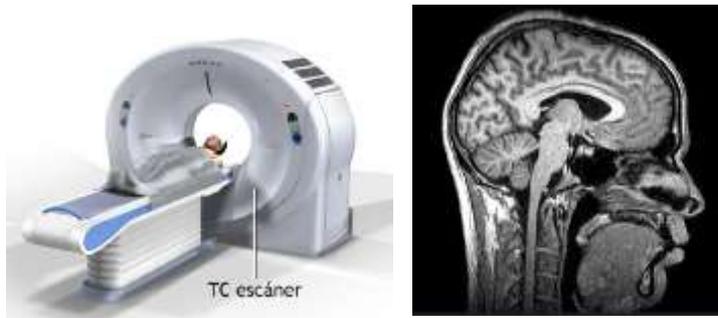


Figura 2-3. Imágenes de una tomografía computarizada y su resultado

Imagen Ultrasonica: La ecografía se utiliza para visualizar la estructura y el grosor de las capas de la piel. El análisis de texturas y la extracción de características pueden aplicarse a imágenes ecográficas para caracterizar lesiones cutáneas o anomalías.



Figura 2-4. Imagen ultrasónica: equipo y toma

Microscopía Confocal de Reflectancia: La microscopía confocal proporciona imágenes en tiempo real de alta resolución a nivel celular. Las técnicas de análisis de imágenes se centran en la morfología celular y pueden ayudar en el diagnóstico de trastornos cutáneos pediátricos identificando patrones celulares específicos.



Figura 2-5. Equipo de microscopía confocal

3 EXTRACCIÓN DE CARACTERÍSTICAS DE COLOR

El color debe ser pensado, soñado, imaginado.

- Henri Matisse -

La extracción de características de color es un proceso fundamental en el análisis de imágenes que busca identificar y cuantificar las propiedades cromáticas de los elementos presentes en una imagen. Esta técnica se utiliza en diversas disciplinas, desde el procesamiento de imágenes médicas hasta la visión por computadora y la industria multimedia. La información de color proporciona detalles valiosos sobre la composición y la distribución de objetos en una imagen, permitiendo la identificación de patrones, la segmentación de regiones de interés y la mejora de la comprensión visual de los datos.

3.1. Teoría del color en imágenes

La teoría del color en imágenes es un campo fundamental que se basa en los principios y modelos que describen cómo percibimos y representamos el color. Estos conceptos son esenciales en diversas disciplinas, como el diseño gráfico, la fotografía, la informática gráfica, la visión por computadora y la medicina. Aquí hay algunas ideas clave relacionadas con la teoría del color en imágenes:

Modelos de Color:

- Los modelos de color son representaciones matemáticas que describen cómo se forman y combinan los colores. Ejemplos comunes incluyen el modelo RGB (rojo, verde, azul), CMYK (cian, magenta, amarillo, negro) utilizado en impresión, y el modelo HSL/HSV (tono, saturación, luminosidad/valor) que describe el color en términos perceptuales.

Percepción del Color:

- La teoría del color también aborda cómo percibimos los colores. Factores como la iluminación, el contexto y las propiedades individuales del observador pueden influir en la percepción del color.

Espacios de Color:

- Un espacio de color es un sistema tridimensional que representa todos los colores posibles. El modelo RGB, por ejemplo, tiene tres dimensiones para los componentes de rojo, verde y azul. La conversión entre diferentes espacios de color es esencial en el procesamiento de imágenes.

Contraste y Combinación de Colores:

- La teoría del color también se aplica al diseño visual, donde el contraste y la combinación de colores desempeñan un papel crucial. Comprender cómo interactúan diferentes colores puede mejorar la legibilidad, la estética y la claridad en las representaciones visuales.

Corrección y Mejora de Color:

- En el procesamiento de imágenes, la corrección y mejora de color se utilizan para ajustar y optimizar la apariencia de una imagen. Esto puede incluir la corrección del balance de blancos, la compensación de colores sobreexpuestos o subexpuestos, y la mejora general de la calidad visual.

Segmentación de Imágenes por Color:

- En la visión por computadora, la segmentación por color se basa en la identificación y separación de regiones de interés en una imagen según sus propiedades cromáticas. Esto es útil en aplicaciones como la detección de objetos o la extracción de características específicas.

Psicología del Color:

- La psicología del color estudia cómo los colores afectan las emociones, el comportamiento y la percepción. En el diseño de interfaces de usuario y presentaciones visuales, se utiliza para transmitir mensajes específicos y generar respuestas emocionales.

En resumen, la teoría del color en imágenes es un campo interdisciplinario que combina principios matemáticos, perceptuales y psicológicos para comprender y manipular eficazmente la información cromática en representaciones visuales. Su aplicación se extiende desde la creación artística hasta la ingeniería de software y la medicina.

3.2. Técnicas de extracción de características de color

Las técnicas de extracción de características de color son métodos utilizados para analizar y extraer información útil de las imágenes basadas en sus componentes de color. Estas técnicas son fundamentales en campos como la visión por ordenador, el procesamiento de imágenes y el reconocimiento de patrones.

3.2.1. Histogramas de color

Los histogramas de color representan la distribución de los valores de color en una imagen. Pueden ser histogramas unidimensionales (por ejemplo, histogramas de intensidad para imágenes en escala de grises) o tridimensionales (para imágenes en color), donde se representan las distribuciones de los componentes de color (rojo, verde, azul o cualquier otro espacio de color utilizado).

Debido a su intuición en comparación con otras cualidades y más información significativa, la simplicidad de la extracción de imágenes y los histogramas de distribución de color al utilizar un conjunto de cuadros, el color es la propiedad más frecuente y comúnmente utilizada.

Para este estudio hemos seleccionado las siguientes características: Media de L, Media de a, Media de b, Varianza de L, Varianza de a, Varianza de b, Skewness (medida de simetría) de L, Skewness de a, Skewness de b, Kurtosis (característica de forma de su distribución de frecuencias/probabilidad) de L, Kurtosis de a, Kurtosis de b.

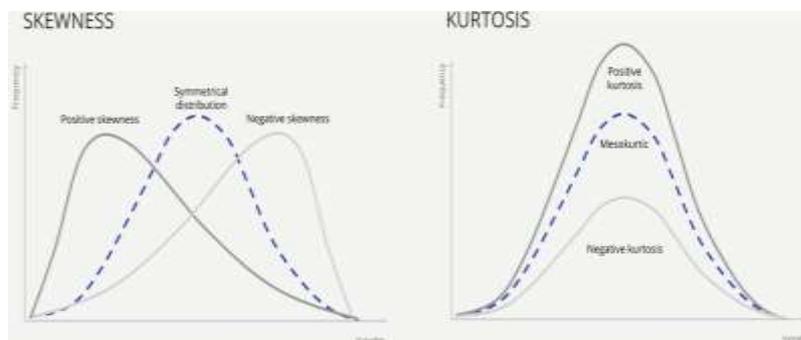


Figura 3-1. Gráfica comparativa de Skewness vs Kurtosis

<p>Fórmula</p> $\text{Kurt} = \frac{\mu_4}{\sigma^4}$ <p>Kurt = curtosis μ_4 = cuarto momento central σ^4 = desviación típica</p>	<p>Formula</p> $\tilde{\mu}_3 = \frac{\sum_i^N (X_i - \bar{X})^3}{(N - 1) * \sigma^3}$ <p>$\tilde{\mu}_3$ = skewness N = number of variables in the distribution X_i = random variable \bar{X} = mean of the distribution σ = standard deviation</p>
--	--

Figura 3-2. Fórmula de Kurtosis y Skewness

3.2.2. Modelos de color (RGB, HSV, LAB)

Los modelos de color como RGB, HSV (Matiz, Saturación, Valor), Lab (Luminosidad, a*, b*), entre otros, permiten representar los colores de una imagen en diferentes sistemas de coordenadas. Cada espacio de color tiene sus propias características que pueden ser útiles para tareas específicas. Por ejemplo, HSV es útil para separar información de tono, saturación y brillo.

Las técnicas basadas en el análisis de color permiten clasificar de manera precisa y eficiente las regiones de hemangiomas en las imágenes. Se han propuesto muchos espacios de color, entre ellos: RGB, rgb, XYZ, Lab*, Luv*, HSV, HLS, YCrCb, YUV, I1I2I3 y TSL. Diferentes posibilidades para modelar la distribución de probabilidad de una clase de color dada aparecen para cada espacio; uno de los métodos no paramétricos más extendidos es el modelado mediante histogramas (ya comentado anteriormente). Esto presenta diversas alternativas para representar una clase de color: el número de canales, qué canales utilizar y el tamaño de los histogramas. Utilizando un conjunto amplio y variado de imágenes de hemangiomas se ha realizado un análisis comparativo exhaustivo de los modelos de color propuestos para el problema de clasificación de las imágenes. Los resultados experimentales demuestran la superioridad de los modelos que separan la luminancia de la crominancia. En particular, Lab* proporciona los mejores resultados con el canal a*, produciendo un 99.2% de clasificación correcta, por eso lo hemos seleccionado para este proyecto, esto se verá en el apartado 8. [7]

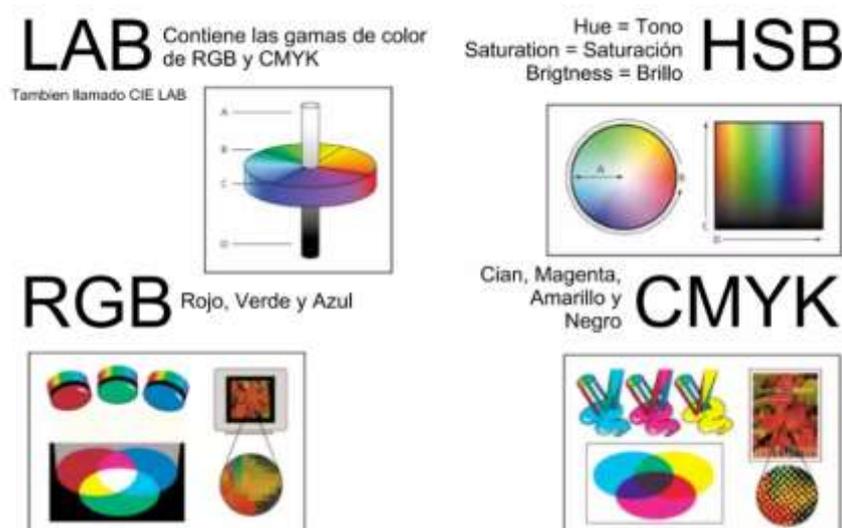


Figura 3-3. Resumen de los modelos de color

3.3 Estudio clínico que muestra la importancia del color y la textura en el diagnóstico de hemangiomas infantiles

Basándose en un estudio realizado en el que tratan a los pacientes con aplicación secuencial de laser de colorante pulsado [8], podemos decir que en la evaluación se utilizó una escala visual basada en fotografías efectuadas antes de realizar la primera sesión de tratamiento y al menos un mes después de la última. Cuatro dermatólogos evaluaron en una escala del 10 al 0 la efectividad del tratamiento teniendo en cuenta el color, la textura cutánea y la elevación de la lesión; considerando el valor de «10» como el original previo al tratamiento y «0» como una piel completamente normal. La media obtenida de las valoraciones de los 4 evaluadores se consideró como el valor de mejoría de cada paciente.

Igualmente se estudiaron las fotografías y la historia clínica en busca de efectos adversos transitorios como edema, infección o ulceración, y otros de mayor duración como cambios pigmentarios o cicatrización anómala. Remarcar que se estudiaron todos los pacientes con hemangiomas en distintas fases evolutivas (proliferativa, en involución y residual) tanto en conjunto, como cada subgrupo de manera independiente. En el análisis estadístico se realizó la prueba de bondad de ajuste a la distribución normal de Kolmogorov-Smirnov, y no se encontraron valores de normalidad en ninguna de las variables que miden los valores de cada uno de los evaluadores, ni en la media de los mismos. Por lo tanto, se utilizaron como valores descriptivos la mediana y el rango intercuartílico.

El cambio entre los valores iniciales y finales fue evaluado mediante la prueba de Wilcoxon, donde se consideró un valor de $p < 0,05$ como estadísticamente significativo.

La concordancia entre los observadores se valoró mediante el coeficiente de correlación intraclase (ICC).

Como conclusión de este estudio y así enlazándolo con el nuestro podemos decir que el tratamiento de los hemangiomas infantiles debe individualizarse en cada paciente según una serie de factores como localización de la lesión, compromiso funcional asociado a la misma, fase evolutiva del hemangioma o demanda de tratamiento activo por parte de los familiares del paciente.

4 EXTRACCIÓN DE CARACTERÍSTICAS DE TEXTURA

La belleza altera la textura de la realidad.

- Donna Tartt -

La extracción de características de textura es un campo crucial en el procesamiento de imágenes y análisis de datos, con una amplia gama de aplicaciones en diversas disciplinas como la medicina, la agricultura, la industria, la seguridad, entre otras. La textura, que se refiere a la distribución espacial de los niveles de intensidad en una imagen, contiene información importante sobre las propiedades estructurales y superficiales de los objetos representados. Por lo tanto, la capacidad para caracterizar y cuantificar la textura de una imagen es fundamental para una variedad de tareas de análisis y reconocimiento automático.

Se explorará en profundidad el proceso de extracción de características de textura, abordando sus fundamentos teóricos, técnicas de extracción más comunes y aplicaciones prácticas. Se examinarán tanto los métodos clásicos como las últimas tendencias y avances en el campo, con el objetivo de comprender su funcionamiento, sus ventajas y limitaciones, así como su idoneidad para diferentes contextos y problemas específicos.

Además, se destacará la importancia de la extracción de características de textura en la era actual de la inteligencia artificial y el aprendizaje automático, donde el análisis de imágenes desempeña un papel crucial en numerosas aplicaciones de vanguardia, como la detección de enfermedades, el reconocimiento de objetos, la monitorización de cultivos, entre otros.

4.1. Conceptos básicos de textura en imágenes

La textura en el contexto de las imágenes se refiere a las propiedades visuales de una superficie representada en la imagen, que determinan su apariencia en términos de suavidad, rugosidad, regularidad, uniformidad, entre otros aspectos. Es la característica que describe la distribución espacial de los niveles de intensidad en una región de la imagen, así como la variación de estos niveles dentro de esa región. La textura puede ser percibida visualmente como patrones repetitivos, variaciones de contraste, o detalles finos que contribuyen a la sensación táctil o visual de la superficie representada. En el análisis de imágenes, la extracción y caracterización de la textura es fundamental para tareas como la segmentación, la clasificación y la detección de objetos.

A la hora de hablar de la textura no nos podemos olvidar de hablar y de definir los siguientes términos:

Distribución de niveles de intensidad: La forma en que los niveles de intensidad están distribuidos en una región de la imagen. Esta distribución puede ser uniforme, regular, aleatoria, entre otros.

Patrones repetitivos: La presencia de patrones repetitivos en una imagen, que pueden ser regulares o irregulares y pueden ser un indicador de textura.

Contraste: La diferencia entre los niveles de intensidad adyacentes en una imagen. Un alto contraste puede indicar una textura más pronunciada.

Co-ocurrencia de píxeles: La probabilidad de que dos píxeles adyacentes tengan ciertos niveles de intensidad juntos. Esta medida se utiliza para capturar la relación espacial entre los píxeles y puede revelar información

sobre la textura.

Histograma de textura: Un histograma que muestra la distribución de ciertas características texturales en una imagen, como la energía, la homogeneidad, la correlación, etc.

Filtros espaciales: Filtros que se aplican a la imagen para resaltar ciertas características texturales, como bordes, líneas o patrones.

Transformada Wavelet: Una técnica que descompone una imagen en diferentes escalas y direcciones, lo que permite analizar la textura en diferentes niveles de detalle.

Descriptores de textura: Características numéricas o vectores que representan la textura de una imagen de manera cuantitativa, como la entropía, la energía, la homogeneidad, etc.

Clasificación de textura: La tarea de asignar una etiqueta a una región de la imagen según su apariencia textural. Se utilizan algoritmos de aprendizaje automático para esta tarea, basados en los descriptores de textura extraídos.

4.2. Métodos de extracción de características de textura

Los métodos de extracción de características de textura son técnicas utilizadas para capturar información cuantitativa sobre la textura presente en una imagen. Estas técnicas se emplean en diversas aplicaciones, como el análisis de imágenes médicas, la inspección industrial, la detección de objetos y la clasificación de imágenes, entre otros. Aquí hay una descripción general de algunos métodos comunes de extracción de características de textura [9]:

Matrices de Co-ocurrencia de Niveles de Gris (GLCM):

- Este método es el que vamos a utilizar y calcula la distribución conjunta de los niveles de gris de los píxeles adyacentes en una imagen. La GLCM es una matriz que describe la frecuencia de aparición de pares de píxeles con valores de gris específicos y una relación espacial determinada dentro de una imagen.

A partir de la GLCM se pueden extraer diferentes medidas estadísticas:

Como la energía (mide la uniformidad de la textura), la homogeneidad (mide la proximidad de la distribución de elementos en la GLCM a la diagonal de la matriz), el contraste (mide la intensidad del contraste entre un píxel y sus vecinos sobre toda la imagen) y la correlación (mide cuán correlacionado está un píxel con sus vecinos a lo largo de la imagen), que caracterizan la textura de la imagen.

Transformada Wavelet:

- La transformada wavelet descompone una imagen en diferentes escalas y orientaciones, permitiendo capturar detalles de textura en diferentes niveles de resolución. Las características de textura se pueden extraer de los coeficientes de detalle de la transformada wavelet en cada escala y orientación.

Filtros de Gabor:

- Los filtros de Gabor son un conjunto de filtros lineales que están optimizados para capturar características de textura en diferentes frecuencias y orientaciones. Estos filtros son sensibles a patrones de textura específicos y pueden utilizarse para extraer características discriminativas de textura.

Histograma de Gradientes Orientados (HOG):

- Este método calcula la distribución de gradientes orientados en una imagen, lo que proporciona información sobre la distribución espacial de los bordes y texturas. El HOG es especialmente útil en la detección de objetos y reconocimiento de patrones.

Transformada de Fourier:

- La transformada de Fourier se utiliza para analizar la frecuencia de los patrones presentes en una imagen. Las características de textura se pueden extraer a partir del espectro de frecuencia de la imagen, lo que permite capturar información sobre la textura periódica y la textura global.

Momentos estadísticos:

- Los momentos estadísticos se utilizan para describir la distribución de intensidades de los píxeles en una imagen. Estos momentos pueden ser invariantes a la escala, la rotación y la traslación, lo que los hace útiles para la extracción de características de textura robustas.

Estos son solo algunos ejemplos de métodos de extracción de características de textura, y existen muchas otras técnicas que se utilizan en función de la aplicación específica y las características de las imágenes. La elección del método adecuado depende de factores como la naturaleza de la textura, la resolución de la imagen y los requisitos de la aplicación. Utilizamos matrices de co-ocurrencia de niveles de gris (GLCM) para evaluar cuantitativamente parámetros y representaciones texturales, y para determinar cuáles son los valores de parámetros y representaciones más adecuados para mapear la textura. Realizamos experimentos sobre los niveles de cuantización de la imagen y los valores de desplazamiento y orientación de la GLCM, examinando los efectos que los descriptores texturales, como la entropía, tienen en la representación de diferentes texturas. [10]

4.3. Implementación de métodos de extracción de características de textura en imágenes de hemangiomas infantiles

La implementación de métodos de extracción de características de textura en imágenes de hemangiomas infantiles utilizando la matriz de co-ocurrencia de niveles de gris (GLCM) implica el análisis cuantitativo de la textura presente en estas imágenes para caracterizar los patrones y estructuras presentes en los hemangiomas. En el apartado anterior ya explicamos los parámetros que extraemos de ella y ahora aquí hay una descripción general de cómo se podría llevar a cabo esta implementación:

Preprocesamiento de las imágenes:

- Antes de aplicar el método GLCM, es importante realizar un preprocesamiento de las imágenes de hemangiomas infantiles para mejorar la calidad y facilitar la extracción de características. Esto puede incluir la corrección de la iluminación, la normalización de la intensidad, el filtrado de ruido, entre otros pasos.

Extracción de la matriz GLCM:

- Una vez que las imágenes han sido preprocesadas, se calcula la matriz de co-ocurrencia de niveles de gris (GLCM) para cada imagen. La GLCM captura la frecuencia con la que ocurren ciertas combinaciones de niveles de gris a una distancia y en una dirección específica en la imagen.

Selección de características:

- A partir de la matriz GLCM, se pueden extraer diversas medidas estadísticas que describen la textura de la imagen. Algunas de estas medidas incluyen la energía, la homogeneidad, el contraste, la correlación y la entropía. Es importante seleccionar cuidadosamente las características más relevantes y discriminativas para la tarea específica de análisis de hemangiomas infantiles.

Análisis de características:

- Una vez que se han extraído las características de textura de las imágenes utilizando la GLCM, se realiza un análisis cuantitativo para identificar patrones y diferencias significativas entre diferentes tipos de hemangiomas o entre hemangiomas y tejido circundante. Esto puede implicar el uso de técnicas estadísticas o de aprendizaje automático para la clasificación y el diagnóstico de los hemangiomas.

Validación y evaluación:

- Finalmente, es crucial validar y evaluar el rendimiento del método de extracción de características de textura implementado. Esto puede implicar la comparación con métodos alternativos, la evaluación de la precisión en la clasificación de hemangiomas y la evaluación del impacto clínico de las características extraídas en el diagnóstico y tratamiento de los pacientes.

En resumen, la implementación de métodos de extracción de características de textura en imágenes de hemangiomas infantiles utilizando la GLCM puede proporcionar información valiosa para la caracterización y el análisis de estas lesiones. Esto puede ayudar a los médicos en el diagnóstico, seguimiento y tratamiento de los

pacientes afectados por hemangiomas infantiles.

5 MÉTODOS DE CLASIFICACIÓN

La realidad es infinitamente diversa, escapa a las deducciones ingeniosas del pensamiento abstracto, no soporta la clasificación estrecha y exacta, la realidad tiende al fraccionamiento perpetuo, a la variedad infinita.

- Fiódor Dostoyevski -

Los métodos de clasificación de imágenes son técnicas utilizadas para asignar etiquetas o categorías a imágenes basadas en características visuales. Estas técnicas son fundamentales en campos como la visión por computadora, el procesamiento de imágenes y el aprendizaje automático. Los métodos de clasificación de imágenes pueden variar desde enfoques simples basados en reglas hasta algoritmos de aprendizaje automático más avanzados. Algunos de los métodos comunes incluyen el K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Redes Neuronales Artificiales (ANN), Árboles de Decisión y Random Forests, entre otros. Cada método tiene sus propias ventajas y desventajas, y la elección del método adecuado depende de la naturaleza de los datos y el problema específico que se esté abordando.

5.1. Algoritmos de clasificación comunes

5.1.1. K-Vecinos más cercanos (K-NN)

Es un algoritmo de aprendizaje supervisado utilizado para clasificación y regresión. Es un método simple pero poderoso en el campo de la clasificación de imágenes.

El principio detrás de KNN es bastante intuitivo. Cuando se le presenta una nueva imagen para clasificar, el algoritmo busca en el conjunto de datos de entrenamiento las K imágenes más cercanas en términos de distancia a la nueva imagen, donde K es un número entero definido por el usuario. Luego, asigna a la nueva imagen la etiqueta que más frecuentemente aparece entre las K imágenes vecinas.

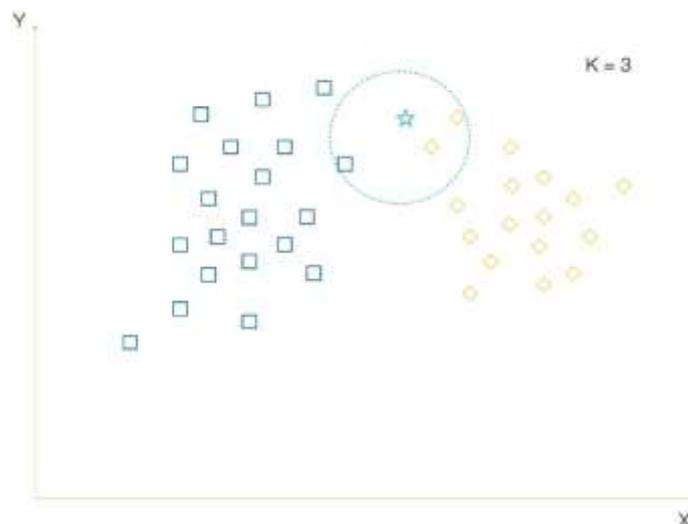


Figura 5-1. Grafismo representativo de una K-NN

La distancia entre las imágenes se puede calcular utilizando diversas métricas, siendo la distancia euclidiana la opción más común. Sin embargo, dependiendo del dominio del problema y de las características de las imágenes, se pueden utilizar otras métricas de distancia.

Una de las ventajas del algoritmo KNN es su simplicidad y su capacidad para manejar datos no lineales y no paramétricos. Además, no requiere de una fase de entrenamiento costosa, ya que todo el proceso de aprendizaje se realiza durante la etapa de clasificación. Sin embargo, KNN puede volverse computacionalmente costoso cuando se trabaja con grandes conjuntos de datos, ya que necesita calcular la distancia entre la nueva imagen y todas las imágenes de entrenamiento.

En resumen, KNN es una herramienta útil y fácil de entender para la clasificación de imágenes, especialmente para conjuntos de datos pequeños o medianos con características bien definidas. Sin embargo, su rendimiento puede verse afectado por la elección de K y la calidad de los datos de entrenamiento. [11]

5.1.2. Máquinas de Soporte Vectorial (SVM)

Es otro algoritmo de aprendizaje supervisado, **es el que vamos a utilizar en este estudio** tanto para clasificación como para regresión. En el contexto de la clasificación de imágenes, SVM es especialmente poderoso debido a su capacidad para manejar conjuntos de datos de alta dimensionalidad, como es común en el caso de las imágenes.

La idea principal detrás de SVM es encontrar el hiperplano óptimo que maximice el margen de separación entre las clases en el espacio de características. En el caso de la clasificación de imágenes, cada imagen se representa como un vector de características de alta dimensión, donde cada dimensión representa una característica específica de la imagen.

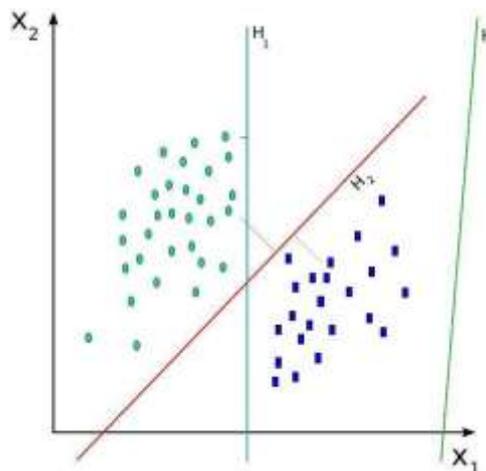


Figura 5-2. Grafismo representativo de una SVM

El hiperplano óptimo encontrado por SVM es aquel que separa las clases de manera más efectiva, minimizando el riesgo de error de clasificación en el conjunto de datos de entrenamiento y generalizando bien a nuevos datos de prueba. En el caso de conjuntos de datos no linealmente separables, SVM puede utilizar una técnica llamada "kernel trick" para proyectar los datos en un espacio de características de mayor dimensión donde las clases sean linealmente separables.

Una de las ventajas clave de SVM es su capacidad para manejar conjuntos de datos de alta dimensionalidad de manera eficiente. Además, SVM es resistente al sobreajuste, especialmente en conjuntos de datos de alta dimensionalidad. Sin embargo, la elección del kernel y los parámetros asociados puede ser crucial para el rendimiento del modelo, y SVM puede volverse computacionalmente costoso en conjuntos de datos muy grandes.

En resumen, SVM es un algoritmo potente y versátil para la clasificación de imágenes, especialmente adecuado

para conjuntos de datos de alta dimensionalidad y que requieren una clara separación entre las clases. Su capacidad para manejar conjuntos de datos no linealmente separables y su resistencia al sobreajuste lo convierten en una opción popular en el campo de la visión por computadora y el procesamiento de imágenes. [12]

5.1.3. Redes Neuronales Artificiales (ANN)

Las Redes Neuronales Artificiales (ANN, por sus siglas en inglés Artificial Neural Networks) son modelos de aprendizaje profundo inspirados en el funcionamiento del cerebro humano. Están compuestas por capas de neuronas interconectadas que procesan la información de entrada y generan una salida.

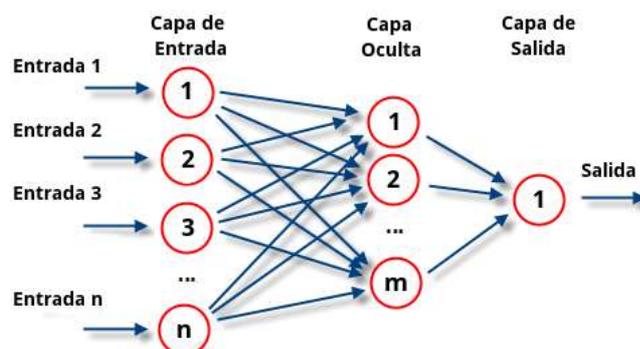


Figura 5-3. Grafismo representativo de una ANN

En el contexto de la clasificación de imágenes, las ANN son ampliamente utilizadas debido a su capacidad para aprender representaciones complejas de los datos. La arquitectura básica de una ANN consiste en una capa de entrada, una o más capas ocultas y una capa de salida. Cada neurona en una capa está conectada a todas las neuronas de la capa anterior y siguiente, formando una red densamente interconectada.

Durante el proceso de entrenamiento, las ANN ajustan los pesos de las conexiones entre las neuronas para minimizar una función de pérdida, que mide la discrepancia entre las salidas predichas y las salidas reales. Este ajuste de pesos se realiza mediante algoritmos de optimización como el descenso de gradiente estocástico.

Una de las ventajas clave de las ANN es su capacidad para aprender automáticamente características relevantes de los datos de entrada, lo que las hace muy adecuadas para la clasificación de imágenes. A medida que las imágenes se propagan a través de las capas de la red, se extraen y combinan características de diferentes niveles de abstracción, permitiendo que la red capture patrones complejos y variaciones en los datos.

Sin embargo, las ANN pueden ser susceptibles al sobreajuste, especialmente en conjuntos de datos pequeños o ruidosos. Además, el entrenamiento de ANN puede ser computacionalmente intensivo, especialmente en arquitecturas profundas con muchas capas y neuronas.

En resumen, las Redes Neuronales Artificiales son herramientas poderosas y flexibles para la clasificación de imágenes, capaces de aprender representaciones complejas de los datos de entrada. Su capacidad para modelar relaciones no lineales y su adaptabilidad a una amplia variedad de problemas las convierten en una opción popular en el campo de la visión por computadora y el aprendizaje automático. [13]

5.2. Entrenamiento y Validación de modelos de clasificación

El entrenamiento y la validación de modelos de clasificación son procesos fundamentales para desarrollar modelos de aprendizaje automático eficaces y generalizables.

Entrenamiento del modelo:

Preprocesamiento de datos: Este paso implica la preparación de los datos de entrenamiento para el modelo. Hemos separado la base de datos de 82 imágenes en dos, una con 72 imágenes que es la que usaremos de entrenamiento y otra con 10 imágenes para la realización del test.

Selección del modelo: Se elige un algoritmo de clasificación adecuado, en este caso es SVM.

Entrenamiento del modelo: Se utiliza el conjunto de datos de entrenamiento para ajustar los parámetros del modelo SVM. Durante este proceso, el modelo aprende de los datos y busca minimizar una función de pérdida o maximizar una función de precisión.

- Primero selecciona las características relevantes, es decir coge las características extraídas y las entrena con las etiquetas de las imágenes de entrenamiento.
- Segundo se emplea la validación cruzada durante el entrenamiento del modelo para evaluar su rendimiento de manera más robusta. Esto nos da la pérdida de clasificación que dependiendo de cada método de extracción cambiará.

Validación del modelo:

Evaluación del rendimiento: Una vez que el modelo ha sido entrenado, se calculan métricas de evaluación como precisión, recall, para comprender cómo el modelo generaliza a datos no vistos.

Interpretación del modelo: Se analizan las predicciones del modelo y se interpreta su comportamiento para comprender cómo toma decisiones y qué características son más importantes para la clasificación. Para esta parte hemos usado la curva de la **ROC** en la que tomamos como eje X, es el ratio de los falsos positivos y en el eje Y, es el ratio de verdaderos positivos.

En resumen, el entrenamiento y la validación de modelos de clasificación son procesos iterativos que requieren una cuidadosa planificación, preprocesamiento de datos, selección y ajuste de modelos, y evaluación rigurosa del rendimiento del modelo para desarrollar sistemas de clasificación efectivos y generalizables.

5.3. Evaluación de la precisión de la clasificación

Los pasos que hemos seguido para la evaluación de la precisión de la clasificación (es fundamental para entender qué tan bien está funcionando un modelo de clasificación en la tarea para la que fue entrenada), son las siguientes:

Primero generamos una matriz de confusión que es una tabla que describe el rendimiento del modelo en términos de verdaderos positivos, falsos positivos, verdaderos negativos y falsos negativos. Es una herramienta útil para entender dónde está cometiendo errores el modelo, en nuestro caso le damos de parámetros las etiquetas de las 10 imágenes de test y las predicciones con el modelo entrenado.

Segundo calculamos la precisión que es una medida de la proporción de predicciones correctas realizadas por el modelo sobre el total de predicciones. Se calcula como el número de predicciones correctas dividido por el número total de predicciones. Es una métrica útil cuando todas las clases tienen la misma importancia.

En tercer lugar, calculamos la sensibilidad o recall que es una medida de la capacidad del modelo para encontrar todas las instancias de una clase determinada. Se calcula como el número de predicciones correctas para una clase dividido por el número total de instancias que pertenecen a esa clase.

Por último, calculamos el porcentaje de fallo del test que en ninguno de los casos supera el 40% como se muestra más adelante en la sección 7 de resultados.

6 TÉCNICAS DE SELECCIÓN DE CARACTERÍSTICAS

Tienes más cualidades de lo que tú mismo crees; pero para saber si son de oro bueno las monedas, hay que hacerlas rodar, hacerlas circular.

- Gregorio Marañón-

La selección de características es un proceso crucial en la preparación de datos para modelos de aprendizaje automático. Este proceso implica identificar y retener las características más relevantes de un conjunto de datos, eliminando aquellas que son irrelevantes o redundantes. La selección adecuada de características no solo mejora la eficiencia computacional al reducir la dimensionalidad del problema, sino que también puede aumentar la precisión y la interpretabilidad del modelo.

Existen diversas técnicas de selección de características que se pueden clasificar en varias categorías principales: métodos de filtro, métodos de envoltura y métodos integrados. Cada categoría ofrece un enfoque diferente para evaluar la importancia de las características, desde la independencia estadística hasta el rendimiento del modelo predictivo. A continuación, exploraremos algunas de las técnicas más utilizadas en cada una de estas categorías.

6.1. Algoritmos de selección comunes

Técnicas de Filtro:

Estas técnicas se basan en las propiedades estadísticas de las características. No consideran el modelo de aprendizaje en sí, sino que se basan en la relación entre cada característica y la variable de salida.

Chi-Cuadrado: Evalúa la independencia de una característica respecto a la variable de salida utilizando la prueba Chi-cuadrado.

Correlación: Calcula la correlación entre cada característica y la variable de salida. Las características con baja correlación pueden ser eliminadas.

Información Mutua: Mide la cantidad de información que una característica proporciona sobre la variable de salida.

Varianza: Elimina características con baja varianza ya que no aportan información significativa.

Técnicas de Wrapper (o envoltentes):

Estas técnicas evalúan la calidad de un subconjunto de características entrenando un modelo de aprendizaje y utilizando su rendimiento como métrica.

Sequential Forward Selection (SFS): Empieza con un conjunto vacío y añade iterativamente características que mejoran el rendimiento del modelo.

Sequential Backward Selection (SBS): Empieza con todas las características y elimina iterativamente aquellas cuya remoción mejora o no afecta negativamente el rendimiento.

Recursive Feature Elimination (RFE): Entrena el modelo y elimina las características menos importantes iterativamente hasta alcanzar el número deseado de características.

Técnicas de Embedding:

Estas técnicas seleccionan características como parte del proceso de entrenamiento del modelo. Los algoritmos incorporan la selección de características durante su propio entrenamiento.

Regresión Lasso (Least Absolute Shrinkage and Selection Operator): Utiliza una penalización L1 que puede hacer que los coeficientes de algunas características se reduzcan a cero, eliminándolas efectivamente.

Árboles de Decisión y Bosques Aleatorios: Utilizan la importancia de las características basada en la reducción de la impureza de los nodos. Las características menos importantes pueden ser descartadas.

Redes Neuronales con Regularización: Algunas arquitecturas de redes neuronales utilizan técnicas de regularización como Dropout para seleccionar características relevantes.

Métodos Basados en la Importancia del Modelo:

Muchas técnicas de aprendizaje automático pueden calcular la importancia de cada característica como parte de su proceso de entrenamiento. Estas importancias se pueden utilizar para seleccionar las características más relevantes.

Gradient Boosting Machines (GBM): Como XGBoost o LightGBM, que proporcionan medidas de importancia de las características.

Regresión Logística y Modelos Lineales: La importancia de las características puede derivarse de los coeficientes ponderados del modelo.

Métodos Univariados y Multivariados:

Métodos Univariados: Evalúan cada característica de manera individual sin considerar las interacciones entre ellas. Ejemplos incluyen pruebas de t, análisis de varianza (ANOVA) y pruebas de Chi-cuadrado.

Métodos Multivariados: Tienen en cuenta las interacciones entre características, como Análisis de Componentes Principales (PCA) y Análisis Discriminante Lineal (LDA).

Métodos Basados en Heurísticas y Metaheurísticas:

Algoritmos Genéticos: Utilizan mecanismos de selección natural para encontrar subconjuntos de características que optimizan el rendimiento del modelo.

Optimización por Enjambre de Partículas (PSO): Emula el comportamiento de enjambres para explorar el espacio de características y encontrar las más relevantes.

La selección de características es una técnica esencial para mejorar la eficiencia y la precisión de los modelos de aprendizaje automático. La elección del método adecuado depende del tipo de datos, del problema específico y de los recursos computacionales disponibles. Es común combinar varias técnicas para obtener los mejores resultados posibles.

6.2. Algoritmos de selección en este proyecto

6.2.1. Información Mutua (MI)

La Información Mutua es una técnica de filtro utilizada en la selección de características que mide la cantidad de información compartida entre dos variables. En el contexto de aprendizaje automático, se utiliza para evaluar la relevancia de una característica respecto a la variable objetivo. Es particularmente útil porque captura tanto relaciones lineales como no lineales entre las características y la variable de salida.[14]

Concepto de Información Mutua:

La información mutua entre dos variables aleatorias X e Y se define como la reducción en la incertidumbre de una variable dada la otra. Matemáticamente, la información mutua $I(X;Y)$ se puede expresar como:

$$I(X;Y) = H(X) + H(Y) - H(X,Y)$$

donde $H(X)$ y $H(Y)$ son las entropías marginales de X e Y , y $H(X,Y)$ es la entropía conjunta.

$$\begin{aligned} I(X; Y) &\equiv H(X) - H(X | Y) \\ &\equiv H(Y) - H(Y | X) \\ &\equiv H(X) + H(Y) - H(X, Y) \end{aligned}$$

Figura 6-1. Fórmula del Información Mutua

Pasos para calcular la Información Mutua:

1. Discretización (si es necesario): Si las características son continuas, a menudo se discretizan en intervalos para aplicar fórmulas de información mutua.
2. Calcular Entropías: Calcular las entropías marginales $H(X)$, $H(Y)$ y la entropía conjunta $H(X,Y)$.
3. Calcular Información Mutua: Usar las entropías calculadas para determinar $I(X;Y)$.

Aplicación en Selección de Características:

La información mutua se utiliza para seleccionar las características que tienen la mayor dependencia con la variable objetivo. Las características con alta información mutua son consideradas relevantes.

Algoritmo Básico:

Calcular Información Mutua: Para cada característica X_i , calcular la información mutua $I(X_i;Y)$ respecto a la variable objetivo Y .

Ordenar Características: Ordenar las características en función de su información mutua con la variable objetivo.

Seleccionar Características: Elegir las características con los valores más altos de información mutua.

Ventajas:

Captura de Relaciones No Lineales: A diferencia de la correlación, la información mutua puede capturar relaciones no lineales entre las características y la variable objetivo.

Simplicidad y Eficiencia: Es un método relativamente simple y eficiente de calcular, especialmente útil para conjuntos de datos grandes.

Limitaciones:

Discretización: La necesidad de discretizar datos continuos puede llevar a pérdida de información y resultados sensibles a cómo se realiza esta discretización.

No Considera Redundancia: La información mutua evalúa cada característica de manera independiente respecto a la variable objetivo, sin considerar la redundancia entre características.

Conclusión

La información mutua es una técnica poderosa y flexible para la selección de características que puede manejar tanto relaciones lineales como no lineales entre las características y la variable objetivo. A pesar de sus limitaciones, es una herramienta valiosa en el arsenal de cualquier científico de datos para mejorar la eficiencia y precisión de los modelos de aprendizaje automático.

6.2.2. Sequential Forward Selection (SFS)

La Selección Secuencial hacia Adelante (SFS) es una técnica de envoltura (wrapper) utilizada en la selección de características que construye un subconjunto de características de manera iterativa, añadiendo en cada paso la característica que más mejora el rendimiento del modelo. Es un método greedy que busca optimizar la selección de características basándose en el rendimiento del modelo de aprendizaje automático.

Concepto de SFS:

La idea básica detrás de SFS es comenzar con un conjunto vacío de características y añadir características una por una. En cada iteración, la característica añadida es aquella que, cuando combinada con las ya seleccionadas, proporciona la mayor mejora en el rendimiento del modelo.

Pasos de SFS

1. Inicialización: Comenzar con un conjunto vacío de características seleccionadas.

2. Iteración:

- Evaluar cada característica que aún no ha sido seleccionada, combinándola con el conjunto de características seleccionadas.
- Entrenar el modelo utilizando este subconjunto de características.
- Medir el rendimiento del modelo (por ejemplo, usando precisión, recall, F1-score, AUC, etc.).
- Seleccionar la característica que más mejora el rendimiento del modelo.

3. Actualización: Añadir la característica seleccionada al conjunto de características seleccionadas.

4. Repetición: Repetir los pasos 2 y 3 hasta alcanzar un criterio de parada, como un número predefinido de características seleccionadas o una mejora mínima en el rendimiento del modelo.

Ventajas:

Simplicidad: El algoritmo es fácil de entender e implementar.

Efectividad: Puede resultar en una buena combinación de características para el modelo dado, mejorando su rendimiento.

Control: Permite un control iterativo sobre la selección de características, lo que puede ser útil para entender la importancia de cada característica.

Limitaciones:

Computacionalmente Costoso: Para cada iteración, se entrena un modelo nuevo para cada característica candidata, lo que puede ser computacionalmente intensivo.

Greedy: Es una técnica voraz que no garantiza encontrar el subconjunto globalmente óptimo de características, ya que toma decisiones locales en cada paso.

Sobreajuste: Puede ser propenso al sobreajuste, especialmente con conjuntos de datos pequeños, ya que selecciona características basadas en el rendimiento del modelo en el conjunto de entrenamiento.

Conclusión:

La Selección Secuencial hacia Adelante (SFS) es una técnica poderosa y fácil de implementar para la selección de características. A pesar de sus limitaciones en términos de costos computacionales y riesgo de sobreajuste, ofrece un enfoque sistemático para mejorar el rendimiento del modelo mediante la selección iterativa de características relevantes. Es una herramienta valiosa para los científicos de datos que buscan optimizar sus modelos de aprendizaje automático.

6.2.3. Sequential Backward Selection (SBS)

La Selección Secuencial hacia Atrás (SBS) es una técnica de envoltura utilizada en la selección de características que elimina de manera iterativa las características menos importantes, comenzando con el conjunto completo de características. Este método busca optimizar el rendimiento del modelo de aprendizaje automático eliminando características redundantes o irrelevantes.

Concepto de SBS:

SBS es lo opuesto a la Selección Secuencial hacia Adelante (SFS). En lugar de comenzar con un conjunto vacío

y añadir características, SBS comienza con todas las características disponibles y elimina una por una las que menos contribuyen al rendimiento del modelo.

Pasos de SBS:

1. Inicialización: Comenzar con el conjunto completo de características.
2. Iteración:
 - Entrenar el modelo utilizando el conjunto actual de características.
 - Evaluar el rendimiento del modelo.
 - Eliminar cada característica una a una, evaluando el modelo cada vez que se elimina una característica.
 - Seleccionar la característica cuya eliminación tenga el menor impacto negativo (o el mayor impacto positivo) en el rendimiento del modelo.
3. Actualización: Eliminar la característica seleccionada del conjunto de características.
4. Repetición: Repetir los pasos 2 y 3 hasta alcanzar un criterio de parada, como un número predefinido de características restantes o una mejora mínima en el rendimiento del modelo.

Ventajas:

Simplificación del Modelo: Al eliminar características irrelevantes, se puede simplificar el modelo y mejorar su interpretabilidad.

Reducción de Overfitting: Puede ayudar a reducir el sobreajuste al eliminar características ruidosas.

Rendimiento: Potencialmente mejora el rendimiento del modelo en términos de precisión y generalización.

Limitaciones:

Computacionalmente Costoso: Similar a SFS, SBS puede ser computacionalmente intensivo, ya que implica entrenar el modelo múltiples veces para cada iteración.

Greedy: Es un algoritmo voraz, lo que significa que toma decisiones locales y no garantiza encontrar el subconjunto globalmente óptimo de características.

Dependencia del Modelo: El rendimiento y la selección de características dependen del modelo específico utilizado en la evaluación.

Conclusión:

La Selección Secuencial hacia Atrás (SBS) es una técnica efectiva para la selección de características que puede mejorar la simplicidad y el rendimiento del modelo eliminando características innecesarias. A pesar de ser computacionalmente costosa y voraz, ofrece una forma sistemática de optimizar la selección de características basada en el rendimiento del modelo. SBS es una herramienta útil para los científicos de datos que buscan refinar sus modelos y mejorar su capacidad de generalización.

6.2.4. T-test

El t-test es una técnica estadística que se utiliza para determinar si existe una diferencia significativa entre las medias de dos grupos. En el contexto de la selección de características, el t-test se utiliza para evaluar si una característica es relevante para diferenciar entre dos clases de la variable objetivo en problemas de clasificación.

Concepto de t-test:

El t-test compara las medias de una característica X entre dos grupos (por ejemplo, clases 0 y 1) para ver si las diferencias observadas son estadísticamente significativas. Esto ayuda a identificar las características que contribuyen significativamente a la distinción entre las clases.

Tipos de t-test:

t-test para una muestra: Compara la media de una sola muestra con una media conocida o hipotética.

t-test para dos muestras independientes: Compara las medias de dos grupos independientes.

t-test pareado: Compara las medias de dos muestras relacionadas o emparejadas.

En la selección de características, generalmente se utiliza el t-test para dos muestras independientes.

Pasos para Realizar el t-test para Selección de Características:

1. Formulación de Hipótesis:

Hipótesis Nula (H_0): No hay diferencia significativa en las medias de la característica X entre los dos grupos.

Hipótesis Alternativa (H_1): Hay una diferencia significativa en las medias de la característica X entre los dos grupos.

2. Cálculo de la Estadística t:

La estadística t se calcula usando la diferencia de medias de los dos grupos, la varianza dentro de los grupos y el tamaño de las muestras.

$$t = \frac{(\bar{X}_1 - \bar{X}_2)}{\sqrt{\frac{S_1^2}{n_1} + \frac{S_2^2}{n_2}}}$$

Figura 6-2. Fórmula del t-Test

3. Determinación del p-valor:

El p-valor indica la probabilidad de observar una diferencia al menos tan extrema como la observada, bajo la hipótesis nula. Un p-valor bajo (generalmente < 0.05) indica que la diferencia es significativa.

4. Selección de Características:

Las características con un p-valor bajo se consideran relevantes y se seleccionan para el modelo.

Aplicación en Selección de Características:

En el contexto de la selección de características, el t-test se puede aplicar a cada característica individualmente para evaluar su relevancia en la discriminación de clases. Las características con p-valores bajos se seleccionan como relevantes.

Ventajas:

Simplicidad: Fácil de entender e implementar.

Rapidez: Computacionalmente eficiente para conjuntos de datos pequeños y medianos.

Relevancia Estadística: Proporciona una base estadística sólida para la selección de características.

Limitaciones:

Asunción de Normalidad: El t-test asume que las muestras siguen una distribución normal, lo que puede no ser el caso en la práctica.

Sólo para Comparaciones Binarias: El t-test es adecuado para problemas de clasificación binaria. Para más de dos clases, se requieren técnicas más avanzadas como ANOVA.

Independencia de Características: Evalúa cada característica de manera independiente, sin considerar la posible correlación entre características.

Conclusión:

El t-test es una técnica estadística útil para la selección de características en problemas de clasificación binaria. Su simplicidad y eficiencia lo hacen adecuado para conjuntos de datos pequeños y medianos, proporcionando

una forma sólida de identificar características relevantes basadas en diferencias significativas entre las clases. Sin embargo, es importante tener en cuenta sus limitaciones y complementar esta técnica con otras cuando sea necesario.

6.3. Técnicas de reducción de dimensionalidad

Las técnicas de reducción de dimensionalidad son métodos utilizados para transformar datos de alta dimensión en un espacio de menor dimensión, conservando la mayor cantidad de información relevante posible. Estas técnicas son fundamentales en el preprocesamiento de datos para mejorar la eficiencia de los algoritmos de aprendizaje automático, facilitar la visualización y reducir el ruido.

Las técnicas de reducción de dimensionalidad son esenciales para manejar datos de alta dimensión, mejorando tanto el rendimiento computacional como la capacidad de interpretación. La elección de la técnica adecuada depende de la naturaleza de los datos y del objetivo específico del análisis, ya sea para preprocesamiento, visualización o mejora del rendimiento de los modelos de aprendizaje automático.

PCA y t-SNE son técnicas poderosas para la reducción de dimensionalidad y visualización de datos. Mientras que PCA puede ser considerado una técnica indirecta de selección de características al identificar las dimensiones más importantes, t-SNE es principalmente una herramienta de visualización. Ambos métodos pueden complementar otras técnicas de selección de características proporcionando una mejor comprensión de los datos y ayudando a identificar las dimensiones más relevantes.

6.3.1. Principal Component Analysis (PCA)

PCA es una técnica de reducción de dimensionalidad que transforma los datos originales en un nuevo conjunto de variables no correlacionadas llamadas componentes principales. Estos componentes están ordenados de tal manera que el primer componente tiene la mayor varianza posible, el segundo componente tiene la segunda mayor varianza, y así sucesivamente.

Pasos de PCA:

Estandarización: Los datos son estandarizados para tener una media de 0 y una varianza de 1.

Matriz de Covarianza: Se calcula la matriz de covarianza de los datos estandarizados.

Vectores y Valores Propios: Se calculan los vectores y valores propios de la matriz de covarianza.

Componentes Principales: Se seleccionan los componentes principales, que son combinaciones lineales de las características originales, ordenados por la cantidad de varianza explicada.

Uso de PCA:

Reducción de Dimensionalidad: PCA reduce la dimensionalidad del conjunto de datos manteniendo la mayor parte de la varianza.

Visualización: Facilita la visualización de datos de alta dimensión en 2D o 3D.

Preprocesamiento: Mejora la eficiencia de los algoritmos de aprendizaje al reducir la dimensionalidad.

En nuestro estudio observamos estas gráficas con respecto a PCA.

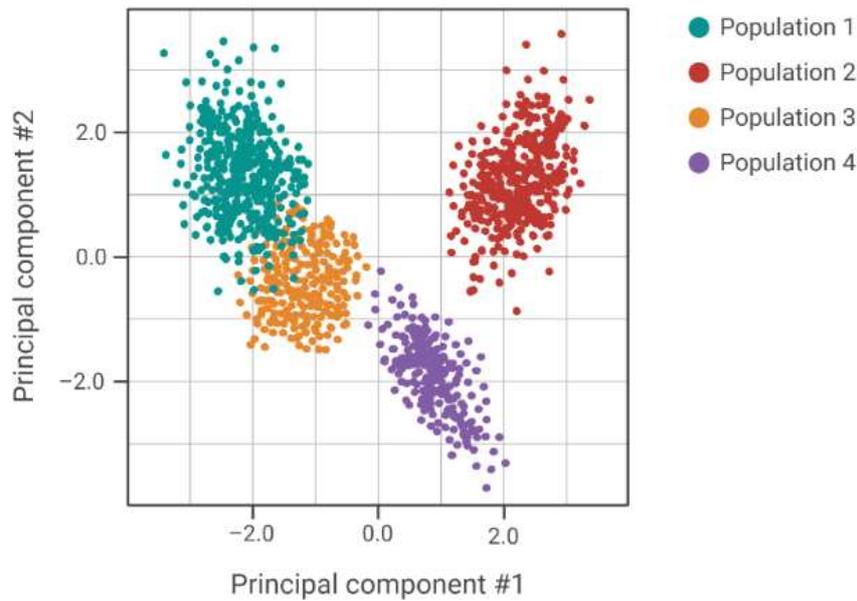


Figura 6-3. Ejemplo de cómo se vería una representación con PCA

6.3.2. T-distributed Stochastic Neighbor Embedding (t-SNE)

t-SNE es una técnica de reducción de dimensionalidad y visualización que se utiliza principalmente para explorar la estructura de datos de alta dimensión en 2D o 3D. t-SNE preserva las relaciones locales entre los puntos de datos, lo que lo hace especialmente útil para visualizar clústeres y la estructura intrínseca de los datos.

Pasos de t-SNE:

Calcular Probabilidades de Similitud: Se calcula una medida de similitud entre pares de puntos en el espacio de alta dimensión.

Minimizar Divergencia KL: Se busca un mapeo en el espacio de menor dimensión que minimice la divergencia de Kullback-Leibler (KL) entre las distribuciones de similitud en los dos espacios.

Uso de t-SNE:

Visualización: Proporciona visualizaciones claras de clústeres y patrones en datos de alta dimensión.

Exploración de Datos: Ayuda a explorar la estructura de los datos y a identificar subgrupos y anomalías.

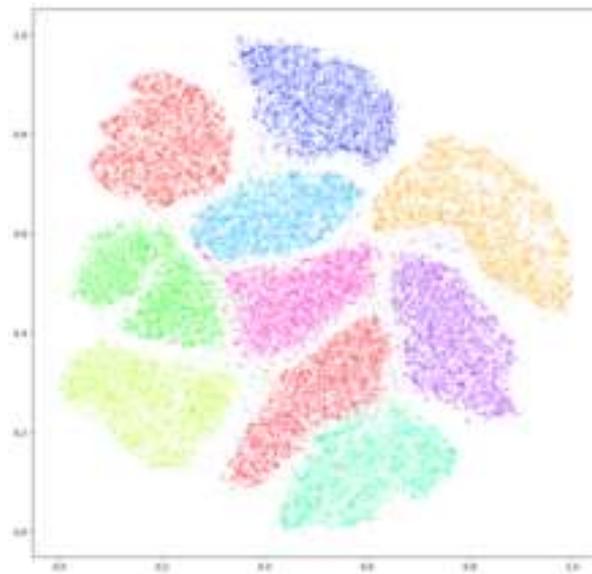


Figura 6-4. Ejemplo de cómo se vería una representación con t-SNE

Todas las figuras aquí presentes son ejemplos de cómo se verían, en el apartado 7.2 se muestra el resultado de aplicarlo a nuestras imágenes.

7 ESTUDIOS Y RESULTADOS RELEVANTES

Estudia las suposiciones que hay detrás de tus acciones.

Luego, estudia las suposiciones que hay tras tus suposiciones.

- Idries Shah -

Este apartado presenta una revisión de estudios significativos y los resultados obtenidos en esta área, enfocándose en cómo las características visuales pueden ser utilizadas para inferir la profundidad en imágenes. La extracción de características de color y textura no solo ha demostrado ser eficaz en la mejora de la precisión de los modelos de clasificación, sino que también ha facilitado avances en diversas aplicaciones prácticas, desde la robótica hasta la fotografía computacional. A continuación, se detallan los enfoques metodológicos más relevantes, los algoritmos utilizados, y los resultados más destacados obtenidos en estudios recientes.

7.1. Materiales

La base de datos usada se compone de 82 imágenes segmentadas, en las que se incluyen tanto superficiales como profundas y mixtas. De las cuales tenemos que superficiales son 44 imágenes, profundas son 12 imágenes y mixtas son las 26 restante.

A posteriori dividimos estas imágenes en solo dos grupos: superficiales y profundas/mixtas. De las 82 imágenes vamos a extraer 5 imágenes de cada grupo mencionado anteriormente para reservarlas para el test posterior.

Quedandonos así 72 imágenes para entrenamiento y 10 imágenes para test. Para la toma de las imágenes ya mencionadas hemos usado el siguiente equipo:

- Una cámara de espectro visible basada en un sensor Sony IMX219 de 8 megapíxeles.
- Una cámara térmica LEPTON que opera en la banda infrarroja de onda larga (LWIR). Después de la calibración, puede tomar imágenes térmicas y medir valores de temperatura absoluta dentro de la escena visual.
- Un termómetro IR Melexis MLX90614D para calibrar la cámara térmica.
- Una pantalla táctil HDMI Waveshare de 5 pulgadas para mostrar las imágenes capturadas por los dos sensores de imagen y controlar la interfaz de usuario. La interfaz permite la adquisición simultánea de imágenes visibles e IR.
- Una batería estándar Poweradd de 5 V para hacer el sistema autónomo y portátil.

7.2. Resultados obtenidos utilizando técnicas de extracción de características de color y textura

Haciendo así un total de 12 características, para un total de 82 imágenes segmentadas a la cuales les hemos eliminado la parte negra de la imagen.



Figura 7-3. Visualización de la imagen original vs la recortada

7.2.1. Métodos de selección características

Método 1: Información Mutua

Características seleccionadas para las 72 imágenes de entrenamiento y los valores que determinan la selección:

Característica	Información Mutua
-----	-----
1 Media de luminosidad (L)	0.2818
2 Media a	0.1770
4 Desviación estándar de L	0.1600
5 Desviación estándar de a	0.1292
6 Desviación estándar de b	0.1080
9 Skewness de a	0.1662
11 Skewness de b	0.1341

Datos extraídos de la selección de características:

- Pérdida de clasificación (classLoss): 0.2222
- Exactitud: 0.7778
- Precisión: 0.7442
- Sensibilidad: 0.8649
- Valor del umbral seleccionado para esta prueba: umbral de im = 0.1

Método 2: t-test

Características seleccionadas para las 72 imágenes de entrenamiento y los valores que determinan la selección:

Característica	Valor de p
-----	-----
1 Media de luminosidad (L)	0.0000
2 Media de a	0.0001
7 Skewness de L	0.0189
9 Skewness de a	0.0021

Datos extraídos de la selección de características:

- Pérdida de clasificación (classLoss): 0.2917
- Exactitud: 0.7083
- Precisión: 0.6905
- Sensibilidad: 0.7838
- Valor del umbral seleccionado para esta prueba: umbral de $p = 0.05$

Método 3: SFS

Características seleccionadas para las 72 imágenes de entrenamiento y los valores que determinan la selección:

Característica	Valor
-----	-----
1 Media de luminosidad (L)	42.1663
3 Media de b	5.0807
4 Desviación estándar de L	6.1921
5 Desviación estándar de a	4.8260

Datos extraídos de la selección de características:

- Pérdida de clasificación (classLoss): 0.2222
- Exactitud: 0.7778
- Precisión: 0.7561
- Sensibilidad: 0.8378
- Valor del umbral seleccionado para esta prueba: criterion value = 0.25

Método 4: SBS

Características seleccionadas para las 72 imágenes de entrenamiento y los valores que determinan la selección:

Característica	Valor
-----	-----
1 Media de luminosidad (L)	42.1663
2 Media de a	16.8568
7 Skewness de L	0.2896
10 Kurtosis de a	2.6483
12 Kurtosis de b	2.8087

Datos extraídos de la selección de características:

- Pérdida de clasificación (classLoss): 0.2500
- Exactitud: 0.7500
- Precisión: 0.7317
- Sensibilidad: 0.8108
- Valor del umbral seleccionado para esta prueba: criterion value = 0.25

¿De dónde sacamos estos valores?

Pérdida de clasificación proviene de la validación cruzada del modelo.

Exactitud = $(TP + TN) / (TP + TN + FP + FN)$;

Precisión = $TP / (TP + FP)$;

Sensibilidad = $TP / (TP + FN)$;

Donde TP = verdadero positivo, TN = verdadero negativo, FP = falsos positivos y FN = falsos negativos.

¿Qué clasificador hemos usado?

Hemos usado el algoritmo de clasificación de máquinas de soporte vectorial (SVM) y evaluamos su precisión o rendimiento con la matriz de confusión.

Resumen de las características seleccionadas por cada método:

	IM	T-TEST	SFS	SBS
MEDIA DE LUMINOSIDAD (L)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
MEDIA DE A	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
MEDIA DE B	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
DESVIACIÓN ESTÁNDAR DE L	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
DESVIACIÓN ESTÁNDAR DE A	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
DESVIACIÓN ESTÁNDAR DE B	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
SKEWNESS DE L	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
KURTOSIS DE L	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
SKEWNESS DE A	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
KURTOSIS DE A	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
SKEWNESS DE B	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
KURTOSIS DE B	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Tabla 7-1. Tabla de las características seleccionadas por cada método

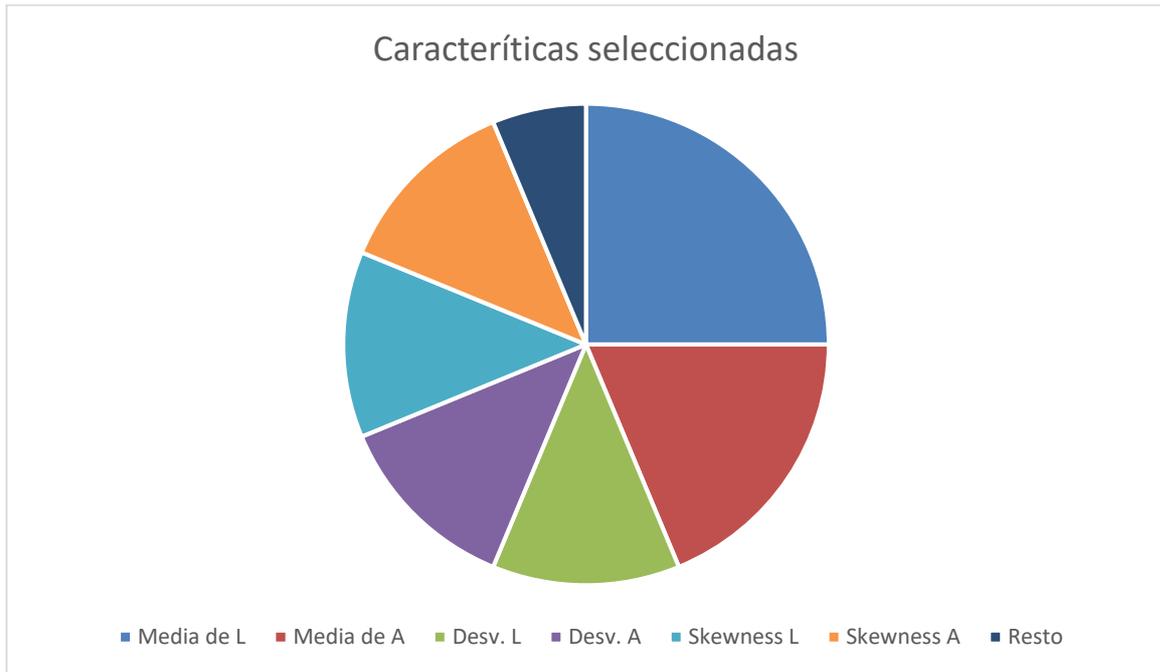


Figura 7-4. Gráfica representativa de las características seleccionadas

Los datos representados nos dan como resultado que la característica más seleccionada es la Media de la luminosidad (L) y que la que no se selecciona en ninguno de los cuatro métodos es Kurtosis de la luminosidad.

Con estos datos obtenidos en la selección de características pasamos a la selección de modelo de clasificación.

7.2.2. Resultados de la clasificación

Gráficas resultantes de la representación de los diferentes métodos de selección combinada con la clasificación a través de SVM tanto para las 72 imágenes de entrenamiento como para las 10 imágenes seleccionadas para hacer el test de las cuales 5 son superficiales y 5 son mixtas/profundas. Hemos escogido los métodos de representación PCA y t-SNE, ya mencionados en el capítulo 6.

Método 1: Información Mutua

Exactitud en test: 0.8000

Precisión en test: 0.8000

Sensibilidad en test: 0.8000

Porcentaje de fallo en test: 20.00%

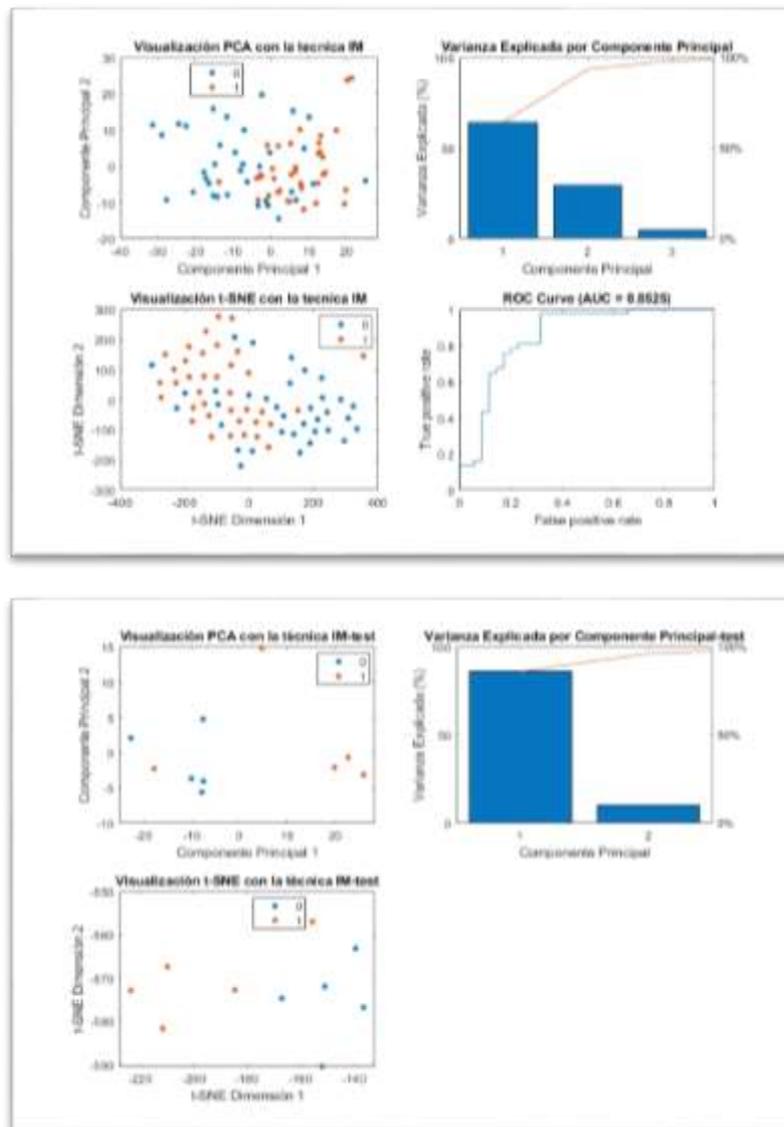


Figura 7-5. Gráfica representativa del método IM sin entrenar vs entrenado

Método 2: t-test

Exactitud en test: 0.7000

Precisión en test: 0.7500

Sensibilidad en test: 0.6000

Porcentaje de fallo en test: 30.00%

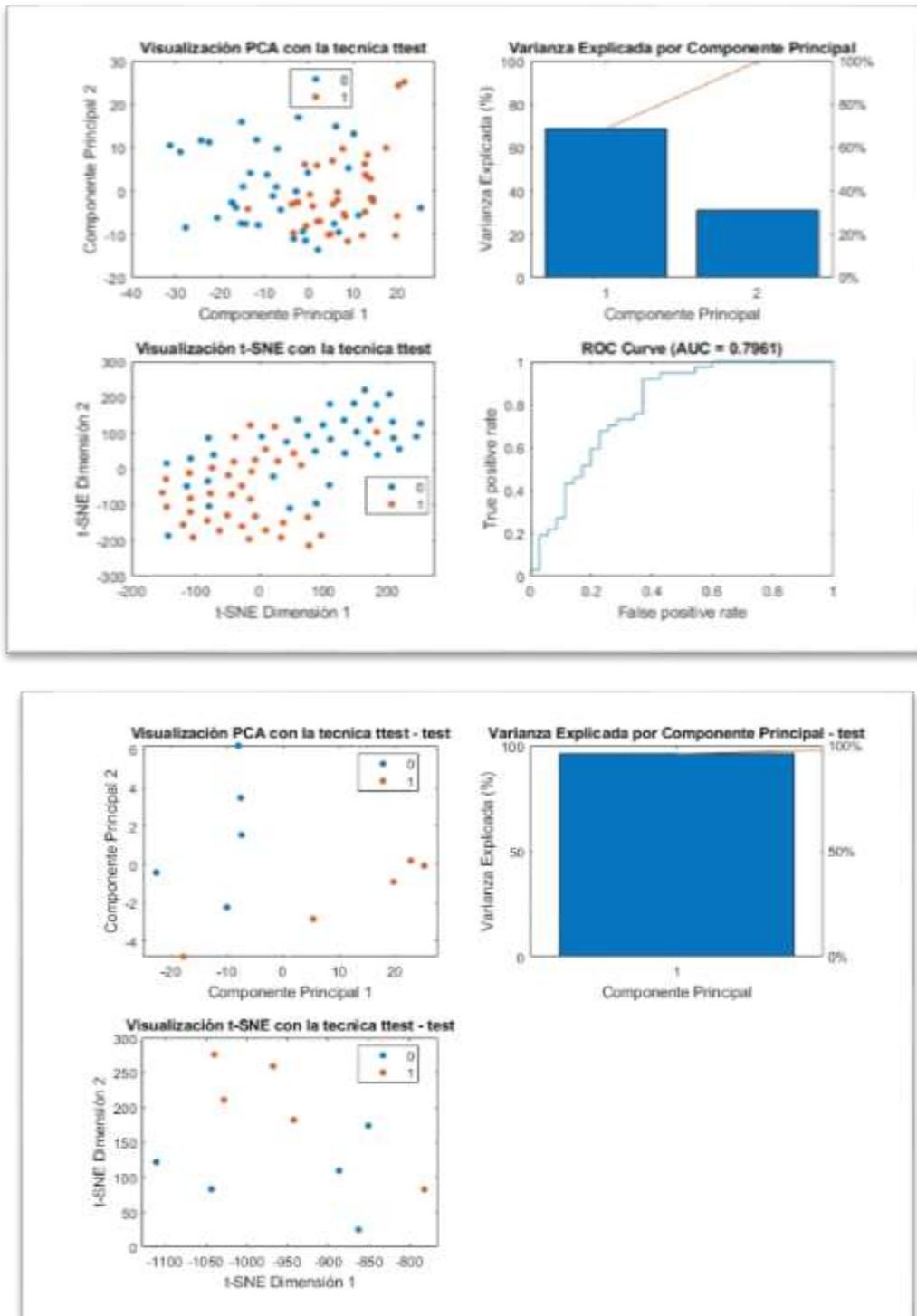


Figura 7-6. Gráfica representativa del método t-Test sin entrenar vs entrenado

Método 3: SFS

Exactitud en test: 0.6000

Precisión en test: 0.6667

Sensibilidad en test: 0.4000

Porcentaje de fallo en test: 40.00%

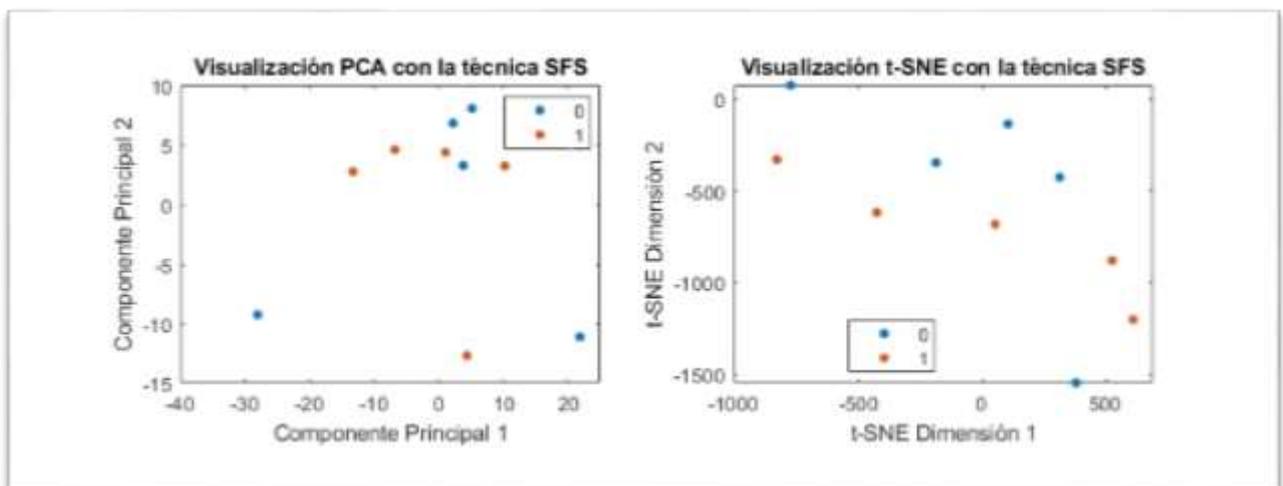
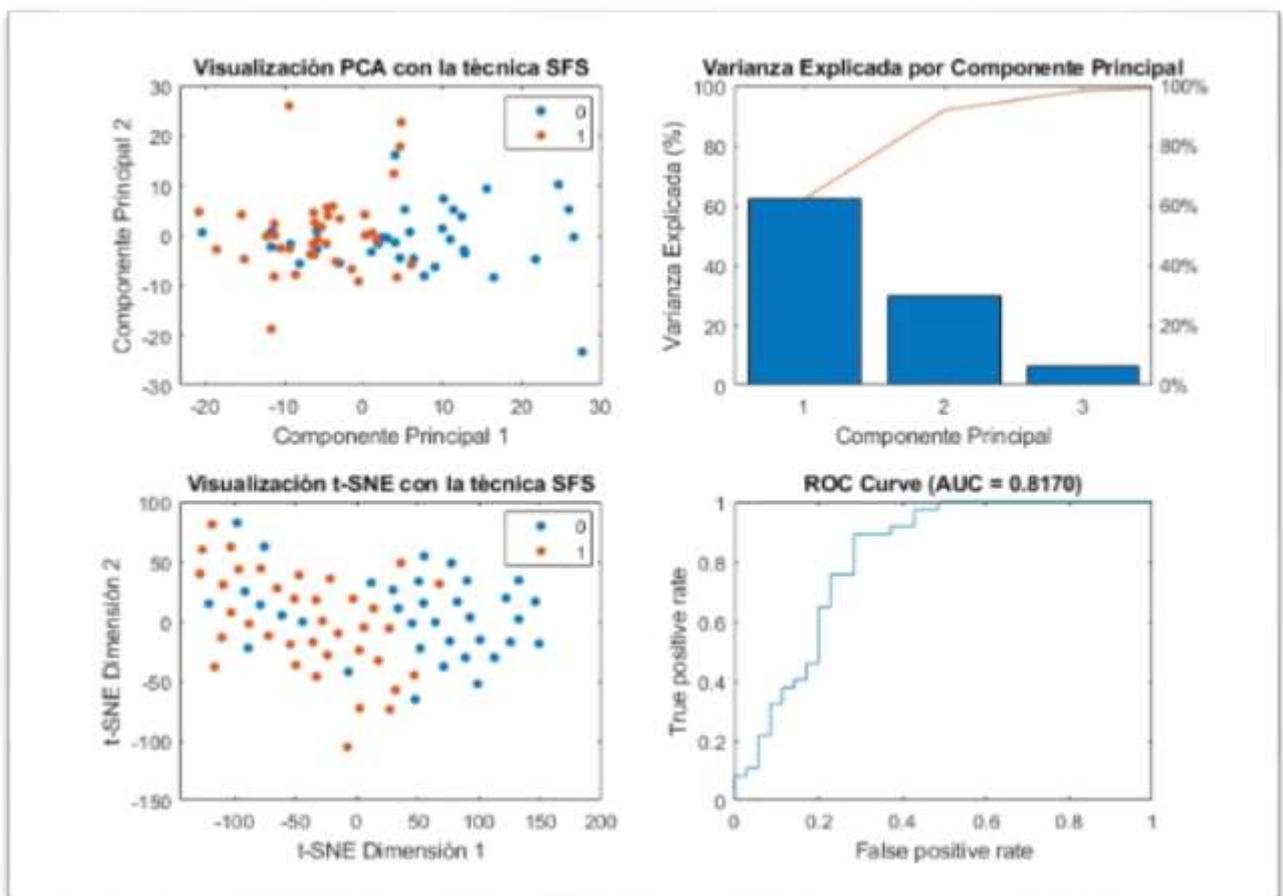


Figura 7-7. Gráfica representativa del método SFS sin entrenar vs entrenado

Método 4: SBS

Exactitud en test: 0.7000

Precisión en test: 0.6667

Sensibilidad en test: 0.8000

Porcentaje de fallo en test: 30.00%

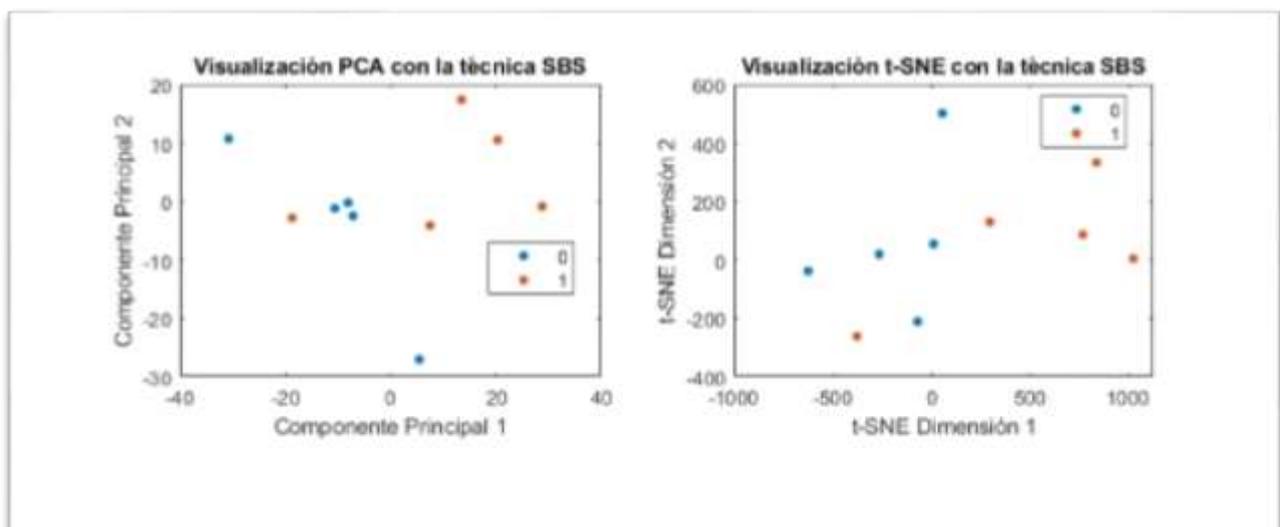
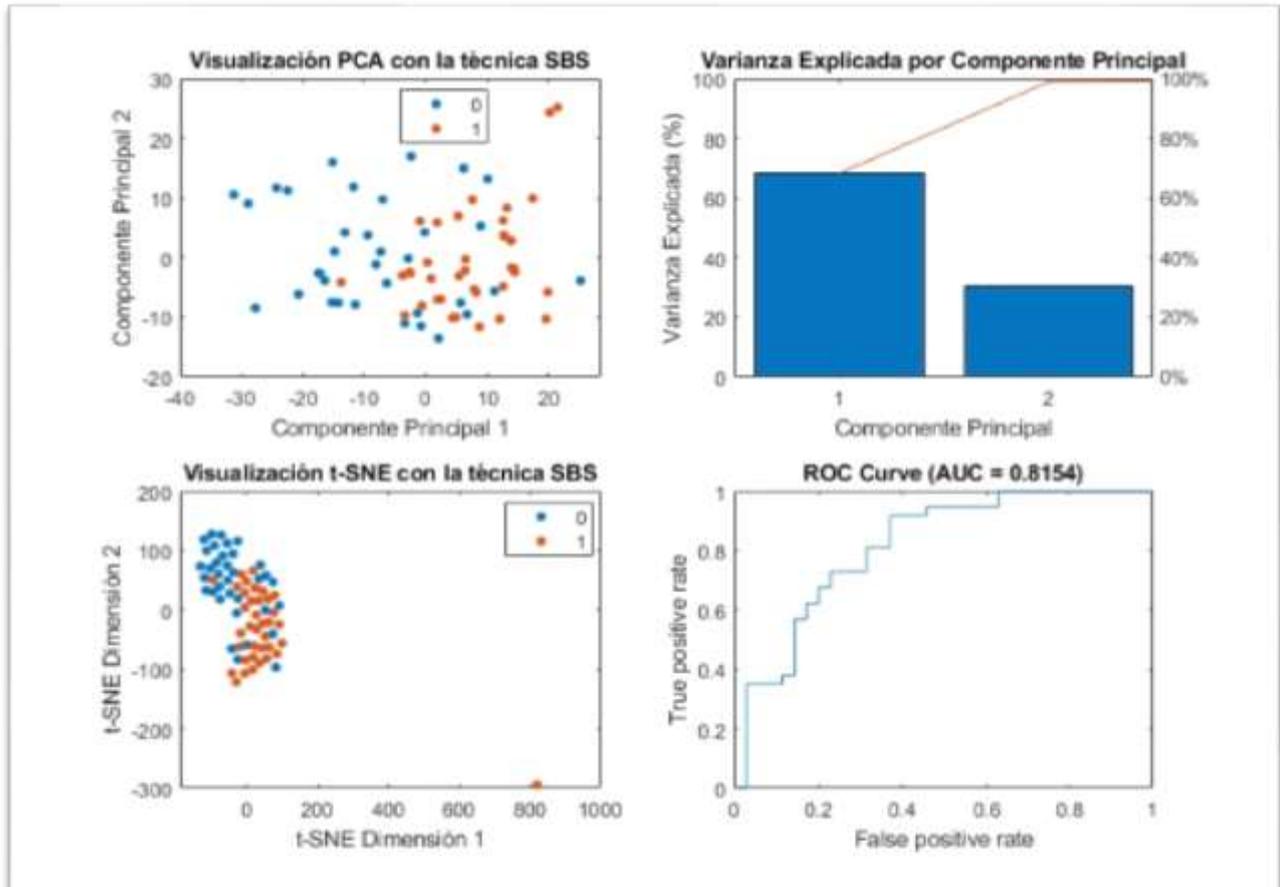


Figura 7-8. Gráfica representativa del método SBS sin entrenar vs entrenado

	IM	T-TEST	SFS	SBS
EXACTITUD	0.8	0.7	0.6	0.7
PRECISIÓN	0.8	0.75	0.6777	0.6777
SENSIBILIDAD	0.8	0.6	0.4	0.8
% DE FALLO	20%	30%	40%	30%

Tabla 7-2. Tabla con todos los resultados de clasificación

Comparando ambos resultados, tanto de la tabla 7-1 como la de la tabla 7-2, vemos una relación entre el número de características seleccionadas por el método con el porcentaje (%) de fallo. A mayor número de características menos porcentaje de fallo, también dependemos de las características seleccionadas, ya que si no coge ninguna característica de skewness o kurtosis (caso de SFS) da un porcentaje de error mayor.

8 DISCUSIÓN Y CONCLUSIONES

Ciencia es todo aquello sobre lo cual siempre cabe discusión.

- José Ortega Y Gasset -

Discusión

El estudio realizado sobre el uso de aprendizaje máquina para el diagnóstico de hemangiomas infantiles ha revelado varios hallazgos interesantes. En particular, hemos comparado diferentes métodos de selección de características y su efectividad en la clasificación de hemangiomas en categorías superficiales o profundas/mixtas.

La Información Mutua se destacó como el método más efectivo, con un error del 20%, significativamente mejor que los otros métodos evaluados: el t-test y SBS (Sequential Backward Selection) con un 30% de error cada uno, y SFS (Sequential Forward Selection) con un 40% de error. Estos resultados sugieren que la Información Mutua es particularmente eficiente en identificar las características más relevantes para la clasificación de hemangiomas infantiles. Esto podría deberse a su capacidad para medir la dependencia entre variables, lo que permite seleccionar características que tienen una mayor relación con la categoría del hemangioma.

El uso de SVM (Máquina de Vectores de Soporte) como modelo de aprendizaje máquina ha sido crucial en este estudio. La SVM es conocida por su robustez y capacidad para manejar conjuntos de datos con un alto número de dimensiones, lo que la hace adecuada para problemas de clasificación complejos como el que abordamos.

Es importante mencionar que el tamaño del conjunto de datos influye directamente en la precisión del modelo. En nuestro estudio, utilizamos 82 imágenes segmentadas, dividiendo el conjunto en 72 imágenes para entrenamiento y 10 imágenes para prueba, con una distribución equitativa de hemangiomas superficiales y profundos en el conjunto de prueba. Esta división permitió una evaluación justa del modelo, pero también limita la generalización de los resultados. Con un conjunto de datos más grande, es probable que la precisión del modelo mejore, ya que el algoritmo tendría más información para aprender y generalizar.

Además, es esencial considerar que otros factores, como la calidad de las imágenes y la precisión en la segmentación, también pueden influir en los resultados. Mejorar estos aspectos podría contribuir a una mayor precisión en el diagnóstico automatizado de hemangiomas infantiles.

Por último, me gustaría mencionar sobre la reducción de dimensionalidad a través de los dos métodos t-SNE y PCA, yo personalmente me quedaría con el método t-SNE (para las selecciones SFS y SBS) ya que se separa mucho mejor visualmente en este espacio transformado. Como digo esto es a título personal, con el otro método (PCA) también se visualizan bien tanto la selección de Información mutua como para el T-Test.

Conclusión

La conclusión a la que llegamos a través del estudio realizado es que, de todos los métodos posibles, el método que ha dado mejor resultado es el de Información Mutua con solo un 20% de error, superando a otros métodos como el t-test y SBS con un 30% de error y SFS con un 40% de error. Bajo estos datos, recomendamos claramente al doctor el uso del método de Información Mutua para evaluar si un hemangioma infantil pertenece a la categoría de superficial o de profunda/mixta, utilizando el aprendizaje máquina con SVM.

Solo nos queda recordar que para este estudio teníamos 82 imágenes segmentadas, de las cuales 10 se usaron para prueba, distribuidas equitativamente entre hemangiomas superficiales y profundos. Esto nos dejó con 72 imágenes de entrenamiento y 10 de prueba. Es importante destacar que, si utilizáramos más imágenes de entrenamiento, el resultado sería aún mejor, ya que proporcionaríamos más datos y variables para entrenar la SVM.

En resumen, este estudio demuestra que la selección adecuada de características, combinada con algoritmos de aprendizaje máquina robustos como la SVM, puede mejorar significativamente el diagnóstico de hemangiomas infantiles. A medida que disponemos de más datos y mejoramos la calidad de las imágenes y la segmentación, podemos esperar que la precisión y la utilidad clínica de estos modelos continúen mejorando.

REFERENCIAS

- [1] Lasierra, N., Alesanco, A., Gilaberte, Y., Magallón, R., & García, J. (2012). Lessons learned after a three- year store and forward teledermatology experience using internet: strengths and limitations. *International Journal of Medical Informatics*, 81(5), 332–343. doi:10.1016/j.ijmedinf.2012.02.008
- [2] Wassef, M.; Blei, F.; Adams, D.; Alomari, A.; Baselga, E.; Berenstein, A.; Burrows, P.; Frieden, I.J.; Garzon, M.C.; Lopez-Gutierrez, J.-C.; et al. Vascular Anomalies Classification: Recommendations from the International Society for the Study of Vascular Anomalies. *Pediatrics* 2015, 136, e203–e214. <https://doi.org/10.1542/peds.2014-3673>.
- [3] Leñero-Bardallo, J.A., Acha, B., ... Bernabéu-Wittel, J., 2022. Thermography as a Method for Bedside Monitoring of Infantile Hemangiomas. *Cancers* 14. doi:10.3390/cancers14215392
- [4] Alves, P.G., Cardoso, J.S., Do Bom-Sucesso, M., 2019. The Challenges of Applying Deep Learning for Hemangioma Lesion Segmentation, in: *Proceedings - European Workshop on Visual Information Processing, EUVIP*. Institute of Electrical and Electronics Engineers Inc. doi:10.1109/EUVIP.2018.8611730
- [5] Acha, B., Serrano, C., ... Roa, L.M., 2005. Segmentation and classification of burn images by color and texture information. *Journal of Biomedical Optics* 10, 034014. doi:10.1117/1.1921227
- [6] Sanz, L.P., 2007. Lesiones vasculares: angiomas. *Protoc la AEP*, 2, pp.65-70.
- [7] García-Mateos, G., Hernández-Hernández, J. L., Escarabajal-Henarejos, D., Jaén-Terrones, S., & Molina-Martínez, J. M. (2015). Study and comparison of color models for automatic image analysis in irrigation management applications. *Agricultural Water Management*, 151, 158–166. doi:10.1016/j.agwat.2014.08.010
- [8] Alcántara-González, J., Boixeda, P., Truchuelo-Díez, M.T., Pérez-García, B., Alonso-Castro, L., & Jaén Olasolo, P. (2012). Hemangiomas infantiles tratados con aplicación secuencial de láser de colorante pulsado y Nd:YAG: estudio retrospectivo. *Actas Dermo-Sifiligráficas*, [doi:10.1016/j.ad.2012.12.010]
- [9] Haralick, R. M., Shanmugam, K., & Dinstein, I. (1973). Textural features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-3(6), 610-621. doi:10.1109/TSMC.1973.4309314
- [10] Soh, L.-K., & Tsatsoulis, C. (1999). Texture analysis of SAR sea ice imagery using gray level co-occurrence matrices. *IEEE Transactions on Geoscience and Remote Sensing*, 37(2 I), 780–795. DOI: 10.1109/36.752194.
- [11] Altman, N.S., 1992. An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, 46(3), pp.175-185

- [12] Cortes, C. and Vapnik, V., 1995. Support-vector networks. *Machine Learning*, 20(3), pp.273-297.
- [13] Haykin, S., 1998. *Neural Networks: A Comprehensive Foundation*. 2nd ed. Upper Saddle River, NJ: Prentice Hall.
- [14] Chandrashekar, G. and Sahin, F., 2014. A survey on feature selection methods. *Electrical and Microelectronic Engineering*, Rochester Institute of Technology, Rochester, NY 14623, USA.

GLOSARIO

IH: Infante hemangioma	xiii
TFG: Trabajo fin de grado	xiii
GLCM: Matriz de co-ocurrencia de niveles de gris	xiii
Hue: Matiz	5
SFS: Selección secuencial hacia delante	5
SBS: Selección secuencial hacia atrás	5
XVAL: Fivefold cross validation	5
TC: Tomografía computarizada	7
IRM: Imágenes por Resonancia Magnética	7
RGB: Modelo de color rojo, verde y azul	9
CMYK: Cian, magenta, amarillo, negro	9
HSL/HSV: Tono, saturación, luminosidad/valor	9
LAB: Luminosidad, a*, b*	11
ICC: Coeficiente de correlación intraclase	12
HOG: Histograma de Gradientes Orientados	14
KNN: K-Nearest Neighbors	17
SVM: Máquinas de Soporte Vectorial	18
ANN: Redes Neuronales Artificiales	19
RFE: Recursive Feature Elimination	21
LASSO : Least Absolute Shrinkage and Selection Operator	22
GBM: Gradient Boosting Machines	22
PCA: Principal Component Analysis	22
MI: Mutual Information	22
T-test: Técnica estadística	25
ANOVA: Análisis de varianza	26
T-SNE: T-distributed Stochastic Neighbor Embedding	28
KL: Divergencia de Kullback-Leibler	28

Código par la versión Matlab 2023b:

Separación de características.m

```
%% Carpeta que contiene las imágenes - TRAINING SET

carpeta_imagenes = 'SEGMENTADAS';
% Leer las imágenes en la carpeta
archivos_imagenes = dir(fullfile(carpeta_imagenes, '*.bmp')); % Cambia la extensión
según el formato de tus imágenes
num_imagenes = length(archivos_imagenes);

% Número de características que se calcularán
num_caracteristicas = 12;
caracteristicas = zeros(num_imagenes, num_caracteristicas); % Inicializar la matriz
con el tamaño correcto

% Cabeceras para las características
headers = {'Mean Lightness', 'Mean a', 'Mean b', 'Std Lightness', 'Std a', 'Std b',
...
          'Skewness L', 'Kurtosis L', 'Skewness a', 'Kurtosis a', 'Skewness b',
          'Kurtosis b'};
contador_imagenes_mostradas = 0;
for i = 1:num_imagenes
    nombre_imagen = archivos_imagenes(i).name;
    ruta_imagen = fullfile(carpeta_imagenes, nombre_imagen);

    % Leer la imagen
    imagen_rgb = imread(ruta_imagen);

    % Obtener las dimensiones de la imagen
    [f, c, ch] = size(imagen_rgb);

    % Pasar la imagen a un vector con reshape de tamaño [1, f*c, 3]
    Ilist = reshape(imagen_rgb, [1, f * c, ch]);

    % Encontrar los píxeles que son distintos de cero
    [index_non_zero] = find(Ilist(1, :, 1) ~= 0);

    % Extraer los píxeles no negros
    I_non_zero = Ilist(1, index_non_zero, :);

    % Mostrar las primeras 10 imágenes
    if contador_imagenes_mostradas < 10
        figure;
        subplot(1, 2, 1);
        imshow(imagen_rgb);
        title('Imagen Original');

        % Mostrar los píxeles no negros como una imagen
        imagen_no_negra = zeros(f, c, ch, 'uint8');
```

```

imagen_no_negra(index_non_zero) = Ilist(1, index_non_zero, 1);
imagen_no_negra(index_non_zero + f * c) = Ilist(1, index_non_zero, 2);
imagen_no_negra(index_non_zero + 2 * f * c) = Ilist(1, index_non_zero, 3);

subplot(1, 2, 2);
imshow(imagen_no_negra);
title('Píxeles no negros');

contador_imagenes_mostradas = contador_imagenes_mostradas + 1;
end

% Convertir los píxeles no negros a espacio de color Lab
I_non_zero_lab = rgb2lab(reshape(I_non_zero, [length(index_non_zero), ch]));

% Extraer las componentes L*, a*, y b*
L = I_non_zero_lab(:, 1);
a = I_non_zero_lab(:, 2);
b = I_non_zero_lab(:, 3);

% Calcular las características solicitadas
mean_lightness = mean(L);
mean_a = mean(a);
mean_b = mean(b);
std_lightness = std(double(L));
std_a = std(double(a));
std_b = std(double(b));
skewness_L = skewness(L);
kurtosis_L = kurtosis(L);
skewness_a = skewness(a);
kurtosis_a = kurtosis(a);
skewness_b = skewness(b);
kurtosis_b = kurtosis(b);

% Almacenar las características en la matriz
caracteristicas(i, :) = [mean_lightness, mean_a, mean_b, std_lightness, std_a,
std_b, ...
skewness_L, kurtosis_L, skewness_a, kurtosis_a,
skewness_b, kurtosis_b];
end

% Imprimir los resultados en forma de tabla
espacio_entre_nombre_y_caracteristica = 40; % Ajustar este valor según sea necesario
fprintf('%-*s', espacio_entre_nombre_y_caracteristica, 'Image Name');
for j = 1:num_caracteristicas
    fprintf('%-15s', headers{j});
end
fprintf('\n');

for i = 1:num_imagenes
    fprintf('%-*s', espacio_entre_nombre_y_caracteristica,
archivos_imagenes(i).name);
    for j = 1:num_caracteristicas
        fprintf('%-15.4f', caracteristicas(i, j));
    end
    fprintf('\n');
end

% Guardar las características y etiquetas en un archivo .mat
save('caracteristicas.mat', 'caracteristicas');
```

```
%% Vamos a crear etiquetas - TRAINING TEST
% Definir las rutas de las carpetas
carpeta_segmentadas = 'SEGMENTADAS';
carpeta_superficiales = 'SUPERFICIALES';

% Obtener nombres de archivos de ambas carpetas
archivos_segmentadas = dir(fullfile(carpeta_segmentadas, '*.bmp'));
archivos_superficiales = dir(fullfile(carpeta_superficiales, '*.jpg'));
archivos_superficiales = [archivos_superficiales; dir(fullfile(carpeta_superficiales,
 '*.png'))];

% Extraer solo los nombres de los archivos (sin extensión)
nombres_segmentadas = {archivos_segmentadas.name};
nombres_superficiales = {archivos_superficiales.name};

% Inicializar celda para almacenar etiquetas
num_imagenes = length(archivos_segmentadas);
etiquetas = zeros(num_imagenes, 1); % Inicializar etiquetas;

% Comparar nombres de archivos y etiquetar
for i = 1:length(archivos_segmentadas)
    nombre_archivo = archivos_segmentadas(i).name;
    [~, nombre_sin_extension, ~] = fileparts(nombre_archivo);

    % Verificar si el archivo está también en carpeta superficiales
    if any(strcmp([nombre_sin_extension '.jpg'], nombres_superficiales)) ||
any(strcmp([nombre_sin_extension '.png'], nombres_superficiales))
        etiquetas(i) = 1;
    else
        etiquetas(i) = 0;
    end
end
% Mostrar tabla con los resultados
disp('Tabla de etiquetas:');
fprintf('-----\n');
fprintf('| Nombre de Archivo   | Etiqueta |\n');
fprintf('-----\n');
for i = 1:num_imagenes
    fprintf('| %-20s | %-8d |\n', archivos_segmentadas(i).name, etiquetas(i));
end
fprintf('-----\n');
% Guardar las etiquetas en un archivo separado
save('etiquetas_segmentadas.mat', 'etiquetas');
```

TEST.m

```

% TEST CON LAS 10 IMAGENES QUE NO SE HAN INCLUIDO %%
% Carpeta que contiene las imágenes - TEST SET
carpeta_imagenes_test = 'Excluidas';
% Leer las imágenes en la carpeta
archivos_imagenes_test = dir(fullfile(carpeta_imagenes_test, '*.bmp')); % Cambia la
extensión según el formato de tus imágenes
num_imagenes_test = length(archivos_imagenes_test);

% Número de características que se calcularán
num_caracteristicas_test = 12;
caracteristicas_test = zeros(num_imagenes_test, num_caracteristicas_test); %
Inicializar la matriz con el tamaño correcto

% Cabeceras para las características
headers = {'Mean Lightness', 'Mean a', 'Mean b', 'Std Lightness', 'Std a', 'Std b',
...
          'Skewness L', 'Kurtosis L', 'Skewness a', 'Kurtosis a', 'Skewness b',
          'Kurtosis b'};
contador_imagenes_mostradas = 0;
for i = 1:num_imagenes_test
    nombre_imagen_test = archivos_imagenes_test(i).name;
    ruta_imagen_test = fullfile(carpeta_imagenes_test, nombre_imagen_test);

    % Leer la imagen
    imagen_rgb_test = imread(ruta_imagen_test);

    % Obtener las dimensiones de la imagen
    [f, c, ch] = size(imagen_rgb_test);

    % Pasar la imagen a un vector con reshape de tamaño [1, f*c, 3]
    Ilist = reshape(imagen_rgb_test, [1, f * c, ch]);

    % Encontrar los píxeles que son distintos de cero
    [index_non_zero] = find(Ilist(1, :, 1) ~= 0);

    % Extraer los píxeles no negros
    I_non_zero = Ilist(1, index_non_zero, :);

    % Mostrar las primeras 10 imágenes
    if contador_imagenes_mostradas < 10
        figure;
        subplot(1, 2, 1);
        imshow(imagen_rgb_test);
        title('Imagen Original');

        % Mostrar los píxeles no negros como una imagen
        imagen_no_negra = zeros(f, c, ch, 'uint8');
        imagen_no_negra(index_non_zero) = Ilist(1, index_non_zero, 1);
        imagen_no_negra(index_non_zero + f * c) = Ilist(1, index_non_zero, 2);
        imagen_no_negra(index_non_zero + 2 * f * c) = Ilist(1, index_non_zero, 3);

        subplot(1, 2, 2);
        imshow(imagen_no_negra);
        title('Píxeles no negros');
    end
end

```

```

        contador_imagenes_mostradas = contador_imagenes_mostradas + 1;
    end

    % Convertir los píxeles no negros a espacio de color Lab
    I_non_zero_lab_test = rgb2lab(reshape(I_non_zero, [length(index_non_zero), ch]));

    % Extraer las componentes L*, a*, y b*
    L = I_non_zero_lab_test(:, 1);
    a = I_non_zero_lab_test(:, 2);
    b = I_non_zero_lab_test(:, 3);

    % Calcular las características solicitadas
    mean_lightness = mean(L);
    mean_a = mean(a);
    mean_b = mean(b);
    std_lightness = std(double(L));
    std_a = std(double(a));
    std_b = std(double(b));
    skewness_L = skewness(L);
    kurtosis_L = kurtosis(L);
    skewness_a = skewness(a);
    kurtosis_a = kurtosis(a);
    skewness_b = skewness(b);
    kurtosis_b = kurtosis(b);
    % Almacenar las características en la matriz
    características_test(i, :) = [mean_lightness, mean_a, mean_b, std_lightness,
std_a, std_b, ...
                                skewness_L, kurtosis_L, skewness_a, kurtosis_a,
skewness_b, kurtosis_b];
end

% Imprimir los resultados en forma de tabla
espacio_entre_nombre_y_caracteristica = 40; % Ajustar este valor según sea necesario
fprintf('%-*s', espacio_entre_nombre_y_caracteristica, 'Image Name');
for j = 1:num_caracteristicas_test
    fprintf('%-15s', headers{j});
end
fprintf('\n');

for i = 1:num_imagenes_test
    fprintf('%-*s', espacio_entre_nombre_y_caracteristica,
archivos_imagenes_test(i).name);
    for j = 1:num_caracteristicas_test
        fprintf('%-15.4f', características_test(i, j));
    end
    fprintf('\n');
end

% Guardar las características y etiquetas en un archivo .mat
save('caracteristicas_test.mat', 'caracteristicas_test');
load('caracteristicas_test.mat', 'caracteristicas_test');
%% ----- %%
%% ----- %%
%% Vamos a crear etiquetas - TRAINING TEST
% Definir las rutas de las carpetas
carpeta_segmentadas_test = 'Excluidas';
carpeta_superficiales_test = 'SUPERFICIALES';

% Obtener nombres de archivos de ambas carpetas
archivos_segmentadas_test = dir(fullfile(carpeta_segmentadas_test, '*.bmp'));

```

```

archivos_superficiales_test = dir(fullfile(carpeta_superficiales_test, '*.jpg'));
archivos_superficiales_test = [archivos_superficiales_test;
dir(fullfile(carpeta_superficiales_test, '*.png'))];

% Extraer solo los nombres de los archivos (sin extensión)
nombres_segmentadas_test = {archivos_segmentadas_test.name};
nombres_superficiales_test = {archivos_superficiales_test.name};

% Inicializar celda para almacenar etiquetas
num_imagenes_test = length(archivos_segmentadas_test);
etiquetas_test = zeros(num_imagenes_test, 1); % Inicializar etiquetas;

% Comparar nombres de archivos y etiquetar
for i = 1:length(archivos_segmentadas_test)
    nombre_archivo_test = archivos_segmentadas_test(i).name;
    [~, nombre_sin_extension_test, ~] = fileparts(nombre_archivo_test);

    % Verificar si el archivo está también en carpeta superficiales
    if any(strcmp([nombre_sin_extension_test '.jpg'], nombres_superficiales_test)) ||
any(strcmp([nombre_sin_extension_test '.png'], nombres_superficiales_test))
        etiquetas_test(i) = 1;
    else
        etiquetas_test(i) = 0;
    end
end
% Mostrar tabla con los resultados
disp('Tabla de etiquetas:');
fprintf('-----\n');
fprintf('| Nombre de Archivo | Etiqueta |\n');
fprintf('-----\n');
for i = 1:num_imagenes_test
    fprintf('| %-20s | %-8d |\n', archivos_segmentadas_test(i).name,
etiquetas_test(i));
end
fprintf('-----\n');
% Guardar las etiquetas en un archivo separado
save('etiquetas_segmentadas_test.mat', 'etiquetas_test');

```

Información Mutua.m

```

% Cargar las características
load('caracteristicas.mat', 'caracteristicas');
load('etiquetas_segmentadas.mat', 'etiquetas');
load('caracteristicas_test.mat', 'caracteristicas_test');
load('etiquetas_segmentadas_test.mat', 'etiquetas_test');
% Supongamos que tienes un conjunto de características almacenadas en la matriz
'caracteristicas'
% y un vector de etiquetas para cada imagen en 'etiquetas'.

% Normalizar las características (opcional pero a menudo recomendado)
caracteristicas_normalizadas = zscore(caracteristicas);

% Calcular la información mutua para cada característica con respecto a las etiquetas
num_caracteristicas = size(caracteristicas, 2);
mutual_info = zeros(1, num_caracteristicas);

for i = 1:num_caracteristicas
    mutual_info(i) = mutualinfo(etiquetas, caracteristicas_normalizadas(:, i));
end

% Seleccionar las características con mayor información mutua
umbral = 0.1; % Umbral de información mutua para la selección
indices_seleccionados = mutual_info > umbral;
caracteristicas_seleccionadas_IM = caracteristicas(:, indices_seleccionados);

% Mostrar las características seleccionadas y sus valores de información mutua
fprintf('%-25s%-15s\n', 'Característica', 'Información Mutua');
fprintf('%-25s%-15s\n', '-----', '-----');
for i = 1:num_caracteristicas
    if indices_seleccionados(i)
        fprintf('%-25d%-15.4f\n', i, mutual_info(i));
    end
end

% Guardar las características seleccionadas en un archivo separado
save('caracteristicas_seleccionadas_IM.mat', 'caracteristicas_seleccionadas_IM');

%% GRAFICAR CON PCA
figure;
% Aplicar PCA para reducir la dimensionalidad a 2 dimensiones
[coeff, score, ~, ~, explained] = pca(caracteristicas_seleccionadas_IM);
pca_features = score(:, 1:2); % Tomar las dos primeras componentes principales

% Graficar utilizando PCA
subplot(2, 2, 1);
gscatter(pca_features(:, 1), pca_features(:, 2), etiquetas);
xlabel('Componente Principal 1');
ylabel('Componente Principal 2');
title('Visualización PCA con la tecnica IM');

% Mostrar el porcentaje de varianza explicada por cada componente principal
subplot(2, 2, 2);
pareto(explained);
xlabel('Componente Principal');
ylabel('Varianza Explicada (%)');
title('Varianza Explicada por Componente Principal');

```

```

%% GRAFICAR CON t-SNE
% Aplicar t-SNE para reducir la dimensionalidad a 2 dimensiones
subplot(2, 2, 3);
rng(1); % Establecer una semilla aleatoria para reproducibilidad
tsne_features = tsne(caracteristicas_seleccionadas_IM, 'NumDimensions', 2);

% Graficar utilizando t-SNE

gscatter(tsne_features(:, 1), tsne_features(:, 2), etiquetas);
xlabel('t-SNE Dimensión 1');
ylabel('t-SNE Dimensión 2');
title('Visualización t-SNE con la tecnica IM');

%% Entrenamiento del SVM para IM
subplot(2, 2, 4);
% Seleccionar las características relevantes
x1 = caracteristicas_seleccionadas_IM; %caracteristicas(:, [1, 2, 4, 5, 6, 9, 11]);
y1 = etiquetas;

% Entrenar el modelo SVM
mod_IM = fitcsvm(x1, y1);

% Validación cruzada del modelo
CVSVMModel = crossval(mod_IM);
classLoss = kfoldLoss(CVSVMModel);
fprintf('Pérdida de clasificación (classLoss): %.4f\n', classLoss);

% Predicciones y métricas
[label, score] = kfoldPredict(CVSVMModel);
confMat = confusionmat(y1, label);

TP = confMat(2,2); % Verdaderos positivos
TN = confMat(1,1); % Verdaderos negativos
FP = confMat(1,2); % Falsos positivos
FN = confMat(2,1); % Falsos negativos

accuracy = (TP + TN) / (TP + TN + FP + FN);
precision = TP / (TP + FP);
recall = TP / (TP + FN);

fprintf('Exactitud: %.4f\n', accuracy);
fprintf('Precisión: %.4f\n', precision);
fprintf('Sensibilidad: %.4f\n', recall);

% Curva ROC
[X, Y, T, AUC] = perfcurve(y1, score(:,2), 1);

plot(X, Y);
xlabel('False positive rate');
ylabel('True positive rate');
title(sprintf('ROC Curve (AUC = %.4f)', AUC));

% Guardar el modelo entrenado
save('modelo_SVM_IM.mat', 'mod_IM');

%% ---- TEST ----
caracteristicas_normalizadas_test = zscore(caracteristicas_test);

% Calcular la información mutua para cada característica con respecto a las etiquetas

```

```

num_caracteristicas_test = size(caracteristicas_test, 2);
mutual_info_test = zeros(1, num_caracteristicas_test);

for i = 1:num_caracteristicas_test
    mutual_info_test(i) = mutualinfo(etiquetas_test,
    caracteristicas_normalizadas_test(:, i));
end

% Ordenar las características por su información mutua (de mayor a menor)
[sorted_mutual_info, sorted_indices] = sort(mutual_info_test, 'descend');

% Seleccionar las 7 características con mayor información mutua
num_caracteristicas_seleccionadas = 7;
indices_seleccionados_test = sorted_indices(1:num_caracteristicas_seleccionadas);
caracteristicas_seleccionadas_IM_test = caracteristicas_test(:,
indices_seleccionados_test);

% Mostrar las características seleccionadas y sus valores de información mutua
fprintf('%-25s%-15s\n', 'Característica-test', 'Información Mutua-test');
fprintf('%-25s%-15s\n', '-----', '-----');
for i = 1:num_caracteristicas_seleccionadas
    fprintf('%-25d%-15.4f\n', indices_seleccionados_test(i), sorted_mutual_info(i));
end

% Guardar las características seleccionadas en un archivo separado
save('caracteristicas_seleccionadas_IM_test.mat',
'caracteristicas_seleccionadas_IM_test');

% Graficar utilizando PCA y t-SNE
figure;

% Aplicar PCA para reducir la dimensionalidad a 2 dimensiones
[coeff, score, ~, ~, explained] = pca(caracteristicas_seleccionadas_IM_test);
pca_features = score(:, 1:2); % Tomar las dos primeras componentes principales

% Graficar utilizando PCA
subplot(2, 2, 1);
gscatter(pca_features(:, 1), pca_features(:, 2), etiquetas_test);
xlabel('Componente Principal 1');
ylabel('Componente Principal 2');
title('Visualización PCA con la técnica IM-test');

% Mostrar el porcentaje de varianza explicada por cada componente principal
subplot(2, 2, 2);
pareto(explained);
xlabel('Componente Principal');
ylabel('Varianza Explicada (%)');
title('Varianza Explicada por Componente Principal-test');

% Aplicar t-SNE para reducir la dimensionalidad a 2 dimensiones
subplot(2, 2, 3);
rng(1); % Establecer una semilla aleatoria para reproducibilidad
tsne_features = tsne(caracteristicas_seleccionadas_IM_test, 'NumDimensions', 2);

% Graficar utilizando t-SNE
gscatter(tsne_features(:, 1), tsne_features(:, 2), etiquetas_test);
xlabel('t-SNE Dimensión 1');
ylabel('t-SNE Dimensión 2');
title('Visualización t-SNE con la técnica IM-test');
%% Estadísticas del modelo

```

```

% Paso 1: Cargar el modelo SVM entrenado
load('modelo_SVM_IM.mat', 'mod_IM');

% Paso 2: Cargar datos de características y etiquetas de test
load('caracteristicas_test.mat', 'caracteristicas_test');
load('etiquetas_segmentadas_test.mat', 'etiquetas_test');

% Paso 3: Aplicar las características seleccionadas del modelo entrenado a los datos
de test
datos_reducidos_im_test = caracteristicas_test(:, indices_seleccionados);
%datos_reducidos_im_test = caracteristicas(:, indices_seleccionados);

% Paso 4: Predecir las etiquetas utilizando el modelo SVM entrenado
predicciones = predict(mod_IM, datos_reducidos_im_test);

% Paso 5: Calcular las métricas de evaluación

% Matriz de confusión
confMat_test = confusionmat(etiquetas_test, predicciones);

TP_test = confMat_test(2,2); % Verdaderos positivos
TN_test = confMat_test(1,1); % Verdaderos negativos
FP_test = confMat_test(1,2); % Falsos positivos
FN_test = confMat_test(2,1); % Falsos negativos

accuracy_test = (TP_test + TN_test) / (TP_test + TN_test + FP_test + FN_test);
precision_test = TP_test / (TP_test + FP_test);
recall_test = TP_test / (TP_test + FN_test);

fprintf('Exactitud en test: %.4f\n', accuracy_test);
fprintf('Precisión en test: %.4f\n', precision_test);
fprintf('Sensibilidad en test: %.4f\n', recall_test);
% Calcular el porcentaje de fallo
porcentaje_fallo_test = (1 - accuracy_test) * 100;

fprintf('Porcentaje de fallo en test: %.2f%%\n', porcentaje_fallo_test);
%% Función de información mutua
function I = mutualinfo(X, Y)
    % Calcular la información mutua entre dos variables discretas X e Y
    Hx = entropy(X);
    Hy = entropy(Y);
    Hxy = joint_entropy(X, Y);
    I = Hx + Hy - Hxy;
end

function H = entropy(X)
    % Calcular la entropía de una variable discreta X
    p = histcounts(X, 'Normalization', 'probability');
    p = p(p > 0);
    H = -sum(p .* log2(p));
end

function H = joint_entropy(X, Y)
    % Calcular la entropía conjunta entre dos variables discretas X e Y
    pxy = histcounts2(X, Y, 'Normalization', 'probability');
    pxy = pxy(pxy > 0);
    H = -sum(pxy .* log2(pxy));
end

```

T-TEST.m

```

% Cargar las características
load('caracteristicas.mat', 'caracteristicas');
load('etiquetas_segmentadas.mat', 'etiquetas');

load('caracteristicas_test.mat', 'caracteristicas_test');
load('etiquetas_segmentadas_test.mat', 'etiquetas_test');
% Supongamos que tienes las características almacenadas en 'caracteristicas'
% y las etiquetas en 'etiquetas' (ya creadas según tu criterio)

% Calcular el t-test para cada característica con respecto a las etiquetas
[~, p_values] = ttest2(caracteristicas(etiquetas==0,:),
caracteristicas(etiquetas==1,:));

% Establecer un umbral de significancia
umbral_p_value = 0.05; % Umbral de significancia para el p-value

% Seleccionar las características con p-values significativos
indices_seleccionados = p_values < umbral_p_value;
caracteristicas_seleccionadas_ttest = caracteristicas(:, indices_seleccionados);

% Mostrar las características seleccionadas y sus valores de p
fprintf('%-25s%-15s\n', 'Característica', 'Valor de p');
fprintf('%-25s%-15s\n', '-----', '-----');
for i = 1:length(p_values)
    if indices_seleccionados(i)
        fprintf('%-25d%-15.4f\n', i, p_values(i));
    end
end

% Guardar las características seleccionadas en un archivo separado
save('caracteristicas_seleccionadas_ttest.mat',
'caracteristicas_seleccionadas_ttest');
load('caracteristicas_seleccionadas_ttest.mat',
'caracteristicas_seleccionadas_ttest');
%% GRAFICAR CON PCA
figure;
% Aplicar PCA para reducir la dimensionalidad a 2 dimensiones
[coeff, score, ~, ~, explained] = pca(caracteristicas_seleccionadas_ttest);
pca_features = score(:, 1:2); % Tomar las dos primeras componentes principales(:,1:2)

% Graficar utilizando PCA
subplot(2, 2, 1);
gscatter(pca_features(:, 1), pca_features(:, 2), etiquetas);
xlabel('Componente Principal 1');
ylabel('Componente Principal 2');
title('Visualización PCA con la tecnica ttest');

% Mostrar el porcentaje de varianza explicada por cada componente principal
subplot(2, 2, 2);
pareto(explained);
xlabel('Componente Principal');
ylabel('Varianza Explicada (%)');
title('Varianza Explicada por Componente Principal');

%% GRAFICAR CON t-SNE

```

```

% Aplicar t-SNE para reducir la dimensionalidad a 2 dimensiones
subplot(2, 2, 3);
rng(1); % Establecer una semilla aleatoria para reproducibilidad
tsne_features = tsne(caracteristicas_seleccionadas_ttest, 'NumDimensions', 2);

% Graficar utilizando t-SNE

gscatter(tsne_features(:, 1), tsne_features(:, 2), etiquetas);
xlabel('t-SNE Dimensión 1');
ylabel('t-SNE Dimensión 2');
title('Visualización t-SNE con la tecnica ttest');
%% Entrenamiento del SVM para ttest
subplot(2, 2, 4);
% Seleccionar las características relevantes
x2 = caracteristicas_seleccionadas_ttest; %caracteristicas(:, [1, 2, 7, 9]);
y2 = etiquetas;

% Entrenar el modelo SVM
mod_tt = fitcsvm(x2, y2);

% Validación cruzada del modelo
CVSVMModel = crossval(mod_tt);
classLoss = kfoldLoss(CVSVMModel);
fprintf('Pérdida de clasificación (classLoss): %.4f\n', classLoss);

% Predicciones y métricas
[label, score] = kfoldPredict(CVSVMModel);
confMat = confusionmat(y2, label);

TP = confMat(2,2); % Verdaderos positivos
TN = confMat(1,1); % Verdaderos negativos
FP = confMat(1,2); % Falsos positivos
FN = confMat(2,1); % Falsos negativos

accuracy = (TP + TN) / (TP + TN + FP + FN);
precision = TP / (TP + FP);
recall = TP / (TP + FN);

fprintf('Exactitud: %.4f\n', accuracy);
fprintf('Precisión: %.4f\n', precision);
fprintf('Sensibilidad: %.4f\n', recall);

% Curva ROC
[X, Y, T, AUC] = perfcurve(y2, score(:,2), 1);

plot(X, Y);
xlabel('False positive rate');
ylabel('True positive rate');
title(sprintf('ROC Curve (AUC = %.4f)', AUC));

% Guardar el modelo entrenado
save('modelo_SVM_tt.mat', 'mod_tt');

%% TEST
% Calcular el t-test para cada característica con respecto a las etiquetas
[~, p_values_test] = ttest2(caracteristicas_test(etiquetas_test == 0, :),
caracteristicas_test(etiquetas_test == 1, :));

% Establecer un umbral de significancia

```

```

umbral_p_value = 0.05; % Umbral de significancia para el p-value

% Seleccionar las características con p-values significativos
indices_seleccionados_test = p_values_test < umbral_p_value;
caracteristicas_seleccionadas_ttest_test = caracteristicas_test(:,
indices_seleccionados_test);

% Mostrar las características seleccionadas y sus valores de p
fprintf('%-25s%-15s\n', 'Característica - test', 'Valor de p - test');
fprintf('%-25s%-15s\n', '-----', '-----');
for i = 1:length(p_values_test)
    if indices_seleccionados_test(i)
        fprintf('%-25d%-15.4f\n', i, p_values_test(i));
    end
end

% Guardar las características seleccionadas en un archivo separado
save('caracteristicas_seleccionadas_ttest_test.mat',
'caracteristicas_seleccionadas_ttest_test');

load('caracteristicas_seleccionadas_ttest_test.mat',
'caracteristicas_seleccionadas_ttest_test');
%% GRAFICAR CON PCA
figure;
% Aplicar PCA para reducir la dimensionalidad a 2 dimensiones
[coeff, score, ~, ~, explained] = pca(caracteristicas_seleccionadas_ttest_test);
pca_features = score(:, 1:2); % Tomar las dos primeras componentes principales(:,1:2)

% Graficar utilizando PCA
subplot(2, 2, 1);
gscatter(pca_features(:, 1), pca_features(:, 2), etiquetas_test);
xlabel('Componente Principal 1');
ylabel('Componente Principal 2');
title('Visualización PCA con la tecnica ttest - test');

% Mostrar el porcentaje de varianza explicada por cada componente principal
subplot(2, 2, 2);
pareto(explained);
xlabel('Componente Principal');
ylabel('Varianza Explicada (%)');
title('Varianza Explicada por Componente Principal - test');

%% GRAFICAR CON t-SNE
% Aplicar t-SNE para reducir la dimensionalidad a 2 dimensiones
subplot(2, 2, 3);
rng(1); % Establecer una semilla aleatoria para reproducibilidad
tsne_features_test = tsne(caracteristicas_seleccionadas_ttest_test, 'NumDimensions',
2);

% Graficar utilizando t-SNE

gscatter(tsne_features_test(:, 1), tsne_features_test(:, 2), etiquetas_test);
xlabel('t-SNE Dimensión 1');
ylabel('t-SNE Dimensión 2');
title('Visualización t-SNE con la tecnica ttest - test');
%% Estadísticas del modelo
% Paso 1: Cargar el modelo SVM entrenado
load('modelo_SVM_tt.mat', 'mod_tt');

% Paso 2: Cargar datos de características y etiquetas de test

```

```
load('caracteristicas_test.mat', 'caracteristicas_test');
load('etiquetas_segmentadas_test.mat', 'etiquetas_test');

% Paso 3: Aplicar las características seleccionadas del modelo entrenado a los datos
de test
datos_reducidos_tt_test = caracteristicas_test(:, indices_seleccionados);

% Paso 4: Predecir las etiquetas utilizando el modelo SVM entrenado
predicciones = predict(mod_tt, datos_reducidos_tt_test);

% Paso 5: Calcular las métricas de evaluación

% Matriz de confusión
confMat_test = confusionmat(etiquetas_test, predicciones);

TP_test = confMat_test(2,2); % Verdaderos positivos
TN_test = confMat_test(1,1); % Verdaderos negativos
FP_test = confMat_test(1,2); % Falsos positivos
FN_test = confMat_test(2,1); % Falsos negativos

accuracy_test = (TP_test + TN_test) / (TP_test + TN_test + FP_test + FN_test);
precision_test = TP_test / (TP_test + FP_test);
recall_test = TP_test / (TP_test + FN_test);

fprintf('Exactitud en test: %.4f\n', accuracy_test);
fprintf('Precisión en test: %.4f\n', precision_test);
fprintf('Sensibilidad en test: %.4f\n', recall_test);
% Calcular el porcentaje de fallo
porcentaje_fallo_test = (1 - accuracy_test) * 100;

fprintf('Porcentaje de fallo en test: %.2f%%\n', porcentaje_fallo_test);
```

SFS.m

```

% Cargar las características y etiquetas
load('caracteristicas.mat', 'caracteristicas');
load('etiquetas_segmentadas.mat', 'etiquetas');

% Definir el número de características a seleccionar
num_caracteristicas_seleccionar = 12; % Ajustar según sea necesario

% Definir la función de evaluación
fun = @(train_data, train_labels, test_data, test_labels) ...
    sum(predict(fitcsvm(train_data, train_labels), test_data) ~= test_labels);

% Aplicar Sequential Feature Selection (SFS)
opts = statset('display', 'iter'); % Mostrar el proceso iterativo
[fs, historial] = sequentialfs(fun, caracteristicas, etiquetas, 'cv', 5, 'options',
opts);

% Guardar las características seleccionadas en un archivo separado
save('caracteristicas_seleccionadas_sfs.mat', 'fs');

% Cargar las características seleccionadas después de SFS (suponiendo que están
guardadas en 'caracteristicas_seleccionadas_sfs.mat')
load('caracteristicas_seleccionadas_sfs.mat', 'fs');

% Mostrar las características seleccionadas
disp('Características seleccionadas :');
disp(find(fs));

% Obtener los datos reducidos con las características seleccionadas
datos_reducidos_sfs = caracteristicas(:, fs);

% Mostrar las características seleccionadas y sus valores
fprintf('%-25s%-15s\n', 'Característica', 'Valor');
fprintf('%-25s%-15s\n', '-----', '-----');
for i = 1:size(caracteristicas, 2)
    if fs(i)
        fprintf('%-25d%-15.4f\n', i, mean(caracteristicas(:, i)));
    end
end

% GRAFICAR CON PCA
figure;

% Aplicar PCA para reducir la dimensionalidad a 2 dimensiones
[coeff, score, ~, ~, explained] = pca(datos_reducidos_sfs);

% Verificar que 'score' tenga al menos 2 columnas
if size(score, 2) >= 2
    % Graficar utilizando PCA
    subplot(2, 2, 1);
    gscatter(score(:, 1), score(:, 2), etiquetas);
    xlabel('Componente Principal 1');
    ylabel('Componente Principal 2');
    title('Visualización PCA con la técnica SFS');
else

```

```

    disp('La reducción de dimensionalidad PCA no tiene suficientes componentes
principales.');
```

```

end

% Mostrar el porcentaje de varianza explicada por cada componente principal
subplot(2, 2, 2);
pareto(explained);
xlabel('Componente Principal');
ylabel('Varianza Explicada (%)');
title('Varianza Explicada por Componente Principal');
```

```

% GRAFICAR CON t-SNE

% Aplicar t-SNE para reducir la dimensionalidad a 2 dimensiones
rng(1); % Establecer una semilla aleatoria para reproducibilidad
tsne_features = tsne(datos_reducidos_sfs, 'NumDimensions', 2);

% Graficar utilizando t-SNE
subplot(2, 2, 3);
gscatter(tsne_features(:, 1), tsne_features(:, 2), etiquetas);
xlabel('t-SNE Dimensión 1');
ylabel('t-SNE Dimensión 2');
title('Visualización t-SNE con la técnica SFS');
```

```

%% Entrenamiento del SVM para sfs
% Seleccionar las características relevantes
x3 = caracteristicas(:, fs);
y3 = etiquetas;

% Entrenar el modelo SVM
mod_sfs = fitcsvm(x3, y3);

% Validación cruzada del modelo
CVSVMModel = crossval(mod_sfs);
classLoss = kfoldLoss(CVSVMModel);
fprintf('Pérdida de clasificación (classLoss): %.4f\n', classLoss);

% Predicciones y métricas
[label, score] = kfoldPredict(CVSVMModel);
confMat = confusionmat(y3, label);

TP = confMat(2,2); % Verdaderos positivos
TN = confMat(1,1); % Verdaderos negativos
FP = confMat(1,2); % Falsos positivos
FN = confMat(2,1); % Falsos negativos

accuracy = (TP + TN) / (TP + TN + FP + FN);
precision = TP / (TP + FP);
recall = TP / (TP + FN);

fprintf('Exactitud: %.4f\n', accuracy);
fprintf('Precisión: %.4f\n', precision);
fprintf('Sensibilidad: %.4f\n', recall);

% Curva ROC
[X, Y, T, AUC] = perfcurve(y3, score(:,2), 1);
subplot(2, 2, 4);
plot(X, Y);
```

```

xlabel('False positive rate');
ylabel('True positive rate');
title(sprintf('ROC Curve (AUC = %.4f)', AUC));

% Guardar el modelo entrenado
save('modelo_SVM_sfs.mat', 'mod_sfs');

%% TEST
% Cargar datos de características y etiquetas
load('caracteristicas_test.mat', 'caracteristicas_test');
load('etiquetas_segmentadas_test.mat', 'etiquetas_test');

% Definir el número de características a seleccionar para la prueba
num_caracteristicas_seleccionar_test = sum(fs); % Ajustar según tus necesidades

% Aplicar Sequential Feature Selection (SFS) para seleccionar características
[fs_test, historial_test] = sequentialfs(fun, caracteristicas_test, etiquetas_test,
'cv', 5, 'NFeatures', num_caracteristicas_seleccionar_test);

% Guardar las características seleccionadas como una matriz en un archivo MAT
save('caracteristicas_seleccionadas_sfs_test.mat', 'fs_test');

% Mostrar el número de características seleccionadas
fprintf('Número de características seleccionadas: %d\n', sum(fs_test));

% Mostrar las características seleccionadas y sus valores
fprintf('%-25s%-15s\n', 'Características - test', 'Valor - test');
fprintf('%-25s%-15s\n', '-----', '-----');
for i = 1:size(caracteristicas_test, 2)
    if fs_test(i)
        fprintf('%-25d%-15.4f\n', i, mean(caracteristicas_test(:, i)));
    end
end
%%
% Paso 1: Cargar el modelo SVM entrenado
load('modelo_SVM_sfs.mat', 'mod_sfs');

% Paso 2: Cargar datos de características y etiquetas de test
load('caracteristicas_test.mat', 'caracteristicas_test');
load('etiquetas_segmentadas_test.mat', 'etiquetas_test');

% Paso 3: Aplicar las características seleccionadas del modelo entrenado a los datos
de test
datos_reducidos_sfs_test = caracteristicas_test(:, fs);

% Paso 4: Predecir las etiquetas utilizando el modelo SVM entrenado
predicciones = predict(mod_sfs, datos_reducidos_sfs_test);

% Paso 5: Calcular las métricas de evaluación

% Matriz de confusión
confMat_test = confusionmat(etiquetas_test, predicciones);

TP_test = confMat_test(2,2); % Verdaderos positivos
TN_test = confMat_test(1,1); % Verdaderos negativos
FP_test = confMat_test(1,2); % Falsos positivos
FN_test = confMat_test(2,1); % Falsos negativos

accuracy_test = (TP_test + TN_test) / (TP_test + TN_test + FP_test + FN_test);
precision_test = TP_test / (TP_test + FP_test);

```

```
recall_test = TP_test / (TP_test + FN_test);

fprintf('Exactitud en test: %.4f\n', accuracy_test);
fprintf('Precisión en test: %.4f\n', precision_test);
fprintf('Sensibilidad en test: %.4f\n', recall_test);

% Calcular el porcentaje de fallo
porcentaje_fallo_test = (1 - accuracy_test) * 100;

fprintf('Porcentaje de fallo en test: %.2f%%\n', porcentaje_fallo_test);

%% GRAFICAR CON PCA
figure;

% Aplicar PCA para reducir la dimensionalidad a 2 dimensiones
[coeff, score, ~, ~, explained] = pca(datos_reducidos_sfs_test);

% Verificar que 'score' tenga al menos 2 columnas
if size(score, 2) >= 2
    % Graficar utilizando PCA
    subplot(2, 2, 1);
    gscatter(score(:, 1), score(:, 2), etiquetas_test);
    xlabel('Componente Principal 1');
    ylabel('Componente Principal 2');
    title('Visualización PCA con la técnica SFS');
else
    disp('La reducción de dimensionalidad PCA no tiene suficientes componentes principales.');
```

```
end

% GRAFICAR CON t-SNE
if size(datos_reducidos_sfs_test, 2) >= 2
    % Aplicar t-SNE para reducir la dimensionalidad a 2 dimensiones
    rng(1); % Establecer una semilla aleatoria para reproducibilidad
    tsne_features_test = tsne(datos_reducidos_sfs_test, 'NumDimensions', 2);

    % Graficar utilizando t-SNE
    subplot(2, 2, 2);
    gscatter(tsne_features_test(:, 1), tsne_features_test(:, 2), etiquetas_test);
    xlabel('t-SNE Dimensión 1');
    ylabel('t-SNE Dimensión 2');
    title('Visualización t-SNE con la técnica SFS');
else
    disp('No hay suficientes características seleccionadas para aplicar t-SNE.');
```

```
end
```

SBS.m

```

% Cargar las características
load('caracteristicas.mat', 'caracteristicas');
load('etiquetas_segmentadas.mat', 'etiquetas');
load('caracteristicas_test.mat', 'caracteristicas_test');
load('etiquetas_segmentadas_test.mat', 'etiquetas_test');

% Supongamos que tienes las características almacenadas en 'caracteristicas'
% y las etiquetas en 'etiquetas' (ya creadas según tu criterio)

% Definir el número de características a seleccionar
num_caracteristicas_seleccionar = 12; % Ajustar según sea necesario

% Definir la función de clasificación (aquí se usa SVM como ejemplo)
modelo_clasificacion = fitcsvm(caracteristicas, etiquetas); % esto en cada metodo im
ttest sfs y sbs %caracteristicas72

%modelo_clasificacion2 = fitcsvm(caracteristicas10, etiquetas); % esto en cada metodo
im ttest sfs y sbs

% Aplicar Sequential Backward Selection (SBS)
fun = @(train_data, train_labels, test_data, test_labels) ...
    sum(predict(fitcknn(train_data, train_labels), test_data) ~= test_labels);

[bs, historial] = sequentialfs(fun, caracteristicas, etiquetas, 'Direction',
'backward');

% Guardar las características seleccionadas en un archivo separado
save('caracteristicas_seleccionadas_sbs.mat', 'bs');

% Cargar las características seleccionadas después de SBS (suponiendo que están
guardadas en 'caracteristicas_seleccionadas_sbs.mat')
load('caracteristicas_seleccionadas_sbs.mat', 'bs');

%% Mostrar las características seleccionadas
disp('Características seleccionadas:');
disp(find(bs));

% Obtener los datos reducidos con las características seleccionadas
datos_reducidos = caracteristicas(:, bs);

% Mostrar las características seleccionadas y sus valores
fprintf('%-25s%-15s\n', 'Característica', 'Valor');
fprintf('%-25s%-15s\n', '-----', '-----');
for i = 1:size(caracteristicas, 2)
    if bs(i)
        fprintf('%-25d%-15.4f\n', i, mean(caracteristicas(:, i)));
    end
end

% % GRAFICAR CON PCA
figure;
% Aplicar PCA para reducir la dimensionalidad a 2 dimensiones

```

```

[coeff, score, ~, ~, explained] = pca(datos_reducidos);

% Graficar utilizando PCA
subplot(2, 2, 1);
gscatter(score(:, 1), score(:, 2), etiquetas);
xlabel('Componente Principal 1');
ylabel('Componente Principal 2');
title('Visualización PCA con la técnica SBS');

% Mostrar el porcentaje de varianza explicada por cada componente principal
subplot(2, 2, 2);
pareto(explained);
xlabel('Componente Principal');
ylabel('Varianza Explicada (%)');
title('Varianza Explicada por Componente Principal');

% % GRAFICAR CON t-SNE
% Aplicar t-SNE para reducir la dimensionalidad a 2 dimensiones
rng(1); % Establecer una semilla aleatoria para reproducibilidad
tsne_features = tsne(datos_reducidos, 'NumDimensions', 2);

% Graficar utilizando t-SNE
subplot(2, 2, 3);
gscatter(tsne_features(:, 1), tsne_features(:, 2), etiquetas);
xlabel('t-SNE Dimensión 1');
ylabel('t-SNE Dimensión 2');
title('Visualización t-SNE con la técnica SBS');
%% Entrenamiento del SVM para ttest
% Seleccionar las características relevantes
x4 = características(:, [1, 2, 7, 10, 12]);%características(:, [1, 2, 3, 5, 6, 9, 10,
12]);
y4 = etiquetas;

% Entrenar el modelo SVM
mod_sbs = fitcsvm(x4, y4);

% Validación cruzada del modelo
CVSVMModel = crossval(mod_sbs);
classLoss = kfoldLoss(CVSVMModel);
fprintf('Pérdida de clasificación (classLoss): %.4f\n', classLoss);

% Predicciones y métricas
[label, score] = kfoldPredict(CVSVMModel);
confMat = confusionmat(y4, label);

TP = confMat(2,2); % Verdaderos positivos
TN = confMat(1,1); % Verdaderos negativos
FP = confMat(1,2); % Falsos positivos
FN = confMat(2,1); % Falsos negativos

accuracy = (TP + TN) / (TP + TN + FP + FN);
precision = TP / (TP + FP);
recall = TP / (TP + FN);

fprintf('Exactitud: %.4f\n', accuracy);
fprintf('Precisión: %.4f\n', precision);
fprintf('Sensibilidad: %.4f\n', recall);

% Curva ROC
[X, Y, T, AUC] = perfcurve(y4, score(:,2), 1);

```

```

subplot(2, 2, 4);
plot(X, Y);
xlabel('False positive rate');
ylabel('True positive rate');
title(sprintf('ROC Curve (AUC = %.4f)', AUC));

% Guardar el modelo entrenado
save('modelo_SVM_sbs.mat', 'mod_sbs');

%% TEST
% Supongamos que tienes las características almacenadas en 'caracteristicas'
% y las etiquetas en 'etiquetas' (ya creadas según tu criterio)
% Ahora procedemos con el modelo de test

% Definir la función de clasificación para el modelo de test
modelo_clasificacion_test = fitcsvm(caracteristicas_test, etiquetas_test);

% Aplicar Sequential Backward Selection (SBS) en el conjunto de test
[bs_test, historial_test] = sequentialfs(fun, caracteristicas_test, etiquetas_test,
'Direction', 'backward');

% Guardar las características seleccionadas en un archivo separado
save('caracteristicas_seleccionadas_sbs_test.mat', 'bs_test');

% Cargar las características seleccionadas después de SBS en el conjunto de test
load('caracteristicas_seleccionadas_sbs_test.mat', 'bs_test');

% Mostrar las características seleccionadas en el conjunto de test
disp('Características seleccionadas - test:');
disp(find(bs_test));

% Obtener los datos reducidos con las características seleccionadas en el conjunto de
test
datos_reducidos_test = caracteristicas_test(:, bs_test);

% Mostrar las características seleccionadas y sus valores en el conjunto de test
fprintf('%-25s%-15s\n', 'Característica-test', 'Valor-test');
fprintf('%-25s%-15s\n', '-----', '-----');
for i = 1:size(caracteristicas_test, 2)
    if bs_test(i)
        fprintf('%-25d%-15.4f\n', i, mean(caracteristicas_test(:, i)));
    end
end
%%
% Paso 1: Cargar el modelo SVM entrenado
load('modelo_SVM_sbs.mat', 'mod_sbs');

% Paso 2: Cargar datos de características y etiquetas de test
load('caracteristicas_test.mat', 'caracteristicas_test');
load('etiquetas_segmentadas_test.mat', 'etiquetas_test');

% Paso 3: Aplicar las características seleccionadas del modelo entrenado a los datos
de test
datos_reducidos_sbs_test = caracteristicas_test(:, bs);

% Paso 4: Predecir las etiquetas utilizando el modelo SVM entrenado
predicciones = predict(mod_sbs, datos_reducidos_sbs_test);

% Paso 5: Calcular las métricas de evaluación

```

```

% Matriz de confusión
confMat_test = confusionmat(etiquetas_test, predicciones);

TP_test = confMat_test(2,2); % Verdaderos positivos
TN_test = confMat_test(1,1); % Verdaderos negativos
FP_test = confMat_test(1,2); % Falsos positivos
FN_test = confMat_test(2,1); % Falsos negativos

accuracy_test = (TP_test + TN_test) / (TP_test + TN_test + FP_test + FN_test);
precision_test = TP_test / (TP_test + FP_test);
recall_test = TP_test / (TP_test + FN_test);

fprintf('Exactitud en test: %.4f\n', accuracy_test);
fprintf('Precisión en test: %.4f\n', precision_test);
fprintf('Sensibilidad en test: %.4f\n', recall_test);

% Calcular el porcentaje de fallo
porcentaje_fallo_test = (1 - accuracy_test) * 100;

fprintf('Porcentaje de fallo en test: %.2f%%\n', porcentaje_fallo_test);
%% GRAFICAR CON PCA
figure;

% Aplicar PCA para reducir la dimensionalidad a 2 dimensiones
[coeff, score, ~, ~, explained] = pca(datos_reducidos_sbs_test);

% Verificar que 'score' tenga al menos 2 columnas
if size(score, 2) >= 2
    % Graficar utilizando PCA
    subplot(2, 2, 1);
    gscatter(score(:, 1), score(:, 2), etiquetas_test);
    xlabel('Componente Principal 1');
    ylabel('Componente Principal 2');
    title('Visualización PCA con la técnica SBS');
else
    disp('La reducción de dimensionalidad PCA no tiene suficientes componentes
principales.');
```

```

end

% GRAFICAR CON t-SNE
if size(datos_reducidos_sbs_test, 2) >= 2
    % Aplicar t-SNE para reducir la dimensionalidad a 2 dimensiones
    rng(1); % Establecer una semilla aleatoria para reproducibilidad
    tsne_features_test = tsne(datos_reducidos_sbs_test, 'NumDimensions', 2);

    % Graficar utilizando t-SNE
    subplot(2, 2, 2);
    gscatter(tsne_features_test(:, 1), tsne_features_test(:, 2), etiquetas_test);
    xlabel('t-SNE Dimensión 1');
    ylabel('t-SNE Dimensión 2');
    title('Visualización t-SNE con la técnica SBS');
else
    disp('No hay suficientes características seleccionadas para aplicar t-SNE.');
```

```

end

```

