

**Instituto Tecnológico y de Estudios Superiores de Monterrey**



**Tecnológico  
de Monterrey**

**BI2009B.300 Procesamiento de imágenes médicas para el diagnóstico**

Profesor: José Gerardo Tamez Peña

Profesora: Martha Rebeca Canales Fiscal

**Diagnóstico por computadora**

Marcela Enriquez López | A01570502

Natalia Verónica Flores Del Río | A01570472

Ana Lucía Soria Cardona | A00827565

Mei Li Luisa Cham Pérez | A01139386

Graciela Alejandra Rincón López | A00827270

A 10 de junio del 2022.

## Introducción

Se estima que anualmente 7,180 personas mueren a causa de melanomas, defunciones que pudieron evitarse si estas se hubieran detectado a tiempo [1]. Los melanomas son caracterizados por ser un lunar con un crecimiento anormal que altera la forma, tamaño, color y borde del lunar; estos pueden desarrollarse en cualquier parte del cuerpo y tomar un color diferente al color esperado de los lunares. Cuando se lleva a cabo la detección temprana de estos se puede remover o recetar el tratamiento ideal para evitar que este se transforme en un melanoma agresivo que puede llegar a convertirse en un melanoma metastásico [2].

Existen múltiples maneras para obtener un diagnóstico temprano de los melanomas, una de las más comunes es las pruebas por imágenes. Las pruebas por imágenes son útiles para identificar anomalías en los lunares, permitiendo la identificación de posibles melanomas [3] [4]. El diagnóstico por computadoras se ha convertido en una herramienta clave en el diagnóstico por medio de imágenes. En los últimos años se ha incrementado el uso de machine learning para poder automatizar y facilitar el diagnóstico de estas condiciones; al usar una base de datos e imágenes de melanomas junto con un modelo para la detección temprana de melanomas y así reducir el número de defunciones que estas causan.

El objetivo de esta práctica es comprender y aplicar los principios de funcionamiento de los modelos de machine learning para la segmentación e identificación de imágenes para así poder diagnosticar de una forma más eficaz y rápida enfermedades como melanomas y prevenir que estas enfermedades tengan mayores repercusiones.

## Marco Teórico

### *K-nearest Neighbor (KNN)*

KNN (K-nearest neighbor) es un algoritmo que se utiliza para resolver problemas de regresión y clasificación de datos, en donde se asume que los datos que son similares están cerca de sí mismos. El funcionamiento de este algoritmo se basa en encontrar las distancias entre un query y todos los números en los datos [5]. Después selecciona un número de ejemplos (K) cercano al query, y por último si se trata de un problema de clasificación, selecciona la etiqueta más frecuente y si es un problema de regresión, promedia las etiquetas. Si se incrementa el valor de K las predicciones se vuelven más estables. Una desventaja de este algoritmo es que entre más datos analice más tiempo se tarda en terminar el proceso, y sus ventajas es que es simple, fácil de implementar y es versátil [6].

### *Naive Bayes*

Naive Bayes es un algoritmo que se basa en el teorema de Bayes y en la probabilidad condicional. Lo que se asume en este algoritmo es que las variables o características son independientes entre sí.

Fórmula del teorema de Bayes:

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)}$$

En donde  $P(h)$  es la probabilidad de que la hipótesis  $h$  sea cierta,  $P(D)$  es la probabilidad de los datos,  $P(D|h)$  es la probabilidad de los datos dado que la probabilidad era cierta y  $P(h|D)$  es la probabilidad de la hipótesis dada los datos [7] [8] [9].

### *Support Vector Machines (SVM)*

SVM (Support vector machines) este clasificador se basa en el concepto de hiperplano que son usados como límites de decisión que ayudan a clasificar los datos, y en el maximal margin classifier que proporciona una distancia asociada al hiperplano, es decir es el hiperplano óptimo. El objetivo principal de este algoritmo es encontrar un hiperplano en el espacio muestral que pueda clasificar los datos de la muestra. Lo que busca encontrar este algoritmo es encontrar el plano que tiene el margen máximo, es decir la máxima distancia entre los datos de dos clases [10].

### *Adaboost*

Adaboost funciona a base de correcciones y crea predictores sencillos en secuencia, es decir que el segundo predictor ajuste lo que el primer predictor no pudo ajustar y así sucesivamente. En general este método es utilizado para reducir la cantidad de errores cuando se hace un análisis predictivo de los datos. Para este método se utiliza la siguiente fórmula:

$$r = \frac{\text{Suma de los pesos de los datos mal clasificados}}{\text{Suma de todos los pesos}}$$

En donde la si la tasa de error es 0 significa que no había muestras mal clasificadas y si la tasa de error es de 1 significa que todas las muestras han sido clasificadas incorrectamente [11].

### *Gradient Boosting Machine (GBM)*

GBM (gradient boosting machine) este método se utiliza para resolver problemas de regresión y de clasificación. Funciona a través de agregar nuevos modelos al conjunto secuencialmente, cada iteración que se realiza se agrega un modelo débil nuevo para así tratarlo con el error de todo el conjunto de datos. Las ventajas de utilizar este método es que da resultados precisos, no se necesitan pre-procesar los datos y también maneja datos faltantes. Las desventajas es que es menos interpretable, puede ser un proceso tardado, puede ocupar mucha memoria en la computadora y como este modelo se enfoca en minimizar los errores esto puede causar que se enfatice demasiado en los valores atípicos y que esto cause un sobreajuste [12] [13].

### *Características de una imagen*

Las características que se pueden extraer de la imagen son la media de la señal, la desviación estándar, oblicuidad y curtosis. La media se obtiene por medio del área total bajo la curva de un ciclo, y la división de esto entre su periodo. La desviación estándar es la medida de la variación con respecto a su media. La oblicuidad indica la simetría de la distribución de la señal. La curtosis muestra el grado de concentración que existe en los valores de una variable alrededor de la zona central.

Se puede obtener también el volumen y la superficie con respecto a la forma. Por parte de la textura, se pueden observar las características Haralick y GLCM. Las Haralick se calculan a partir de la GLCM, que es una matriz que cuenta la ocurrencia de niveles grises en la imagen. Por último se pueden obtener las dimensiones fractales, que miden cuán complicada una imagen “self-similar” es. Mide cuántos puntos hay en un set.

## **Metodología**

Para poder llevar a cabo la segmentación y el entrenamiento de las imágenes de melanomas en las bases de datos otorgadas fue necesario implementar el método de radiónica, método que permite la extracción de características de múltiples imágenes médicas como radiografías para aplicarlas en algoritmos de clasificación de datos.

Este método está compuesto por 5 pasos principales: calibración de imagen, segmentación de imagen, extracción de características, evaluación de características y machine learning. Para la calibración de la imagen se debe de considerar las diferencias que se pueden tener entre los equipos, centros médicos, y los protocolos de imagenología utilizados, por ello es crucial la calibración inicial para que estos factores no interfieran. Seguido de esto va la segmentación de imágenes, en donde se extrae el tejido seleccionado de las imágenes analizadas. Una vez que se completa esto se lleva a cabo la extracción y evaluación de las características, en este paso se extrae una descripción numérica de las características para poder determinar cuales de estas son relevantes para el análisis. Finalmente, se entrena el modelo para poder diferenciar e identificar las características deseadas.

Se hizo uso de las herramientas tecnológicas Matlab y R Studio para poder llevar a cabo el método de radiónica. Los primeros 3 pasos (calibración de imagen, segmentación de imagen, extracción) se llevaron a cabo en Matlab gracias al código proporcionado. En este código se filtraron, procesaron y segmentaron las imágenes que se encontraban en las bases de datos; dentro de este código se encontraron procesos como el shaving, el cual permitió la reducción de vellosidades de la imagen. Así mismo se encontraron otros procesos enfocados a la forma, señales e incluso textura que permitieron la extracción de las características, las cuales se almacenaron en un archivo .csv de gran utilidad para el procesamiento en R Studio junto con el package FRESA CAD. En R Studio se utilizaron métodos de machine learning como KNN, NB, SVM; AdaBoost, y GBM para la evaluación y evaluación estadística de las características para la identificación de melanomas.

## **Resultados**

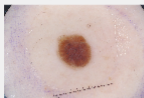
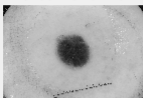
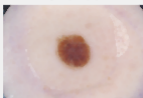
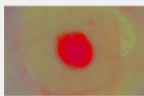

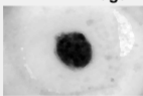
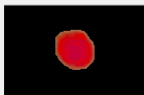



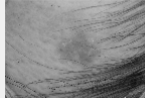

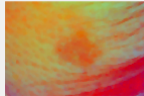





Una vez realizado el análisis se encontraron los siguientes resultados de la segmentación los cuales se encuentran plasmados en las figuras del 1 al 7. La figura 1 y la figura 2 muestran los resultados de la segmentación de dos de las múltiples imágenes de melanomas que se encontraban en la base de datos proporcionada. La figura 1 muestra una segmentación que fue realizada de la manera correcta, mientras que la figura 2 muestra una segmentación de melanoma que fue realizada de una manera errónea. Las principales diferencias entre una segmentación realizada de la manera correcta y errónea es la selección de los bordes de los melanomas, en la segmentación errónea los bordes de los melanomas no fueron identificados por lo cual la segmentación no es buena.

Procesamiento de imágenes Melanoma								
<p><i>Figura 1. Segmentación correcta de imágenes de melanoma</i></p>			<p><i>Figura 2. Segmentación incorrecta de imágenes de melanoma</i></p>					

La figura 3 y 4 muestran los resultados de las segmentación de las diferentes imágenes de Nevus que se encontraban en la base de datos proporcionada. La figura 3 muestra los resultados esperados después de realizar una segmentación correcta de un Nevus, en los resultados se puede observar que se tiene una selección del nevus completa lo cual significa que se tiene una segmentación correcta. Por otro lado la figura 4 muestra una segmentación incorrecta del nevus, ya que no se selecciona de manera completa el borde ni el área interior del nevus lo cual resulta en una segmentación incorrecta.

Procesamiento de imágenes Nevus								
<p><i>Figura 3. Segmentación correcta de imágenes del nevus</i></p>			<p><i>Figura 4. Segmentación incorrecta de imágenes del nevus</i></p>					

Finalmente las figuras 5 y 6 muestran los resultados de las segmentaciones para seborrheic, tanto correctas como incorrectas. La figura 5 muestra los resultados que se obtienen al completar una segmentación correcta, el principal diferenciador es la selección del seborrheic y su área interna. Por otro lado, la figura 6 muestra la segmentación incorrecta del seborrheic, en la cual se distingue por la mala selección del seborrheic. Estos detalles son cruciales para llevar a cabo una segmentación correcta que ayude al machine learning.

Procesamiento de imágenes Seborrheic								
<div>ISIC<sub>0</sub>012845.jpg</div> 			<div>Green Channel</div> 			<div>Shaved</div> 		
<div>Standardized</div> 			<div>Raw Mask</div> 			<div>Min minImage</div> 		
<div>Lesion Mask</div> 			<div>Lesion Sample ROI</div> 			<div>Control ROI</div> 		
<div>Figura 5. Segmentación correcta de imágenes del seborrheic</div>								
<div>ISIC<sub>0</sub>012661.jpg</div> 			<div>Green Channel</div> 			<div>Shaved</div> 		
<div>Standardized</div> 			<div>Raw Mask</div> 			<div>Min minImage</div> 		
<div>Lesion Mask</div> 			<div>Lesion Sample ROI</div> 			<div>Control ROI</div> 		
<div>Figura 6. Segmentación incorrecta de imágenes del seborrheic</div>								

Introducción de información a RStudio

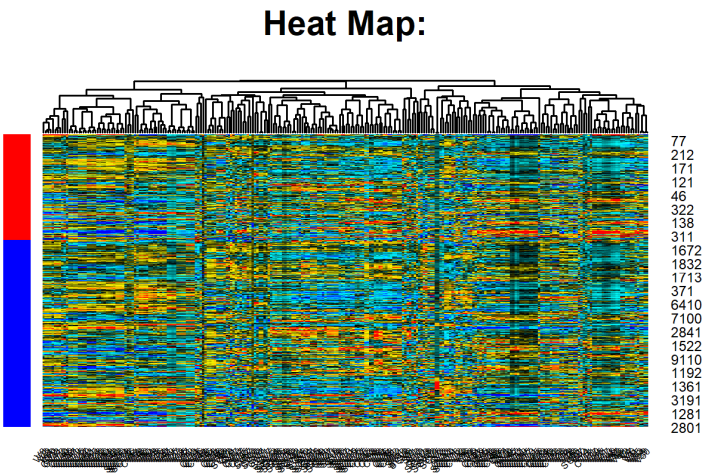


Imagen 7. Heatmap con la distribución de características e información de los datos.

Curvas ROC de máquinas de aprendizaje

De la figura 8 a la figura 12 se pueden encontrar los diferentes resultados de los métodos aplicados para el machine learning en R. La figura 8 muestra la respuesta del método KNN, en el cual se observa un área debajo la curva de aprendizaje de 0.796 y una clase de área bajo la curva de 0.719.

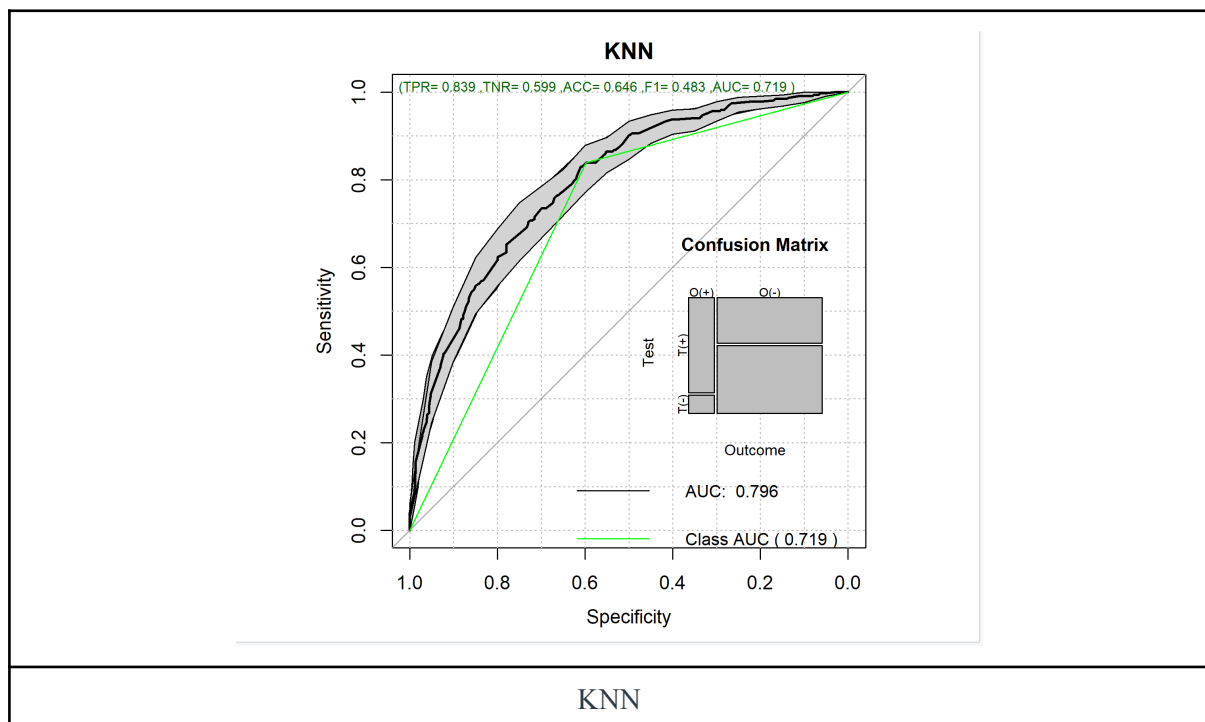


Figura 8: Método KNN en R Studio

La figura 9 muestra los diferentes resultados del método SVM en este método se utilizaron 3 clases de muestreo diferente: Au, Ba, y Pro. El método con Au tiene un área bajo la curva de 0.915, mientras que el método con Ba obtuvo un área bajo la curva de 0.842, y finalmente el método con Pro muestra un área bajo la curva de 0.904.

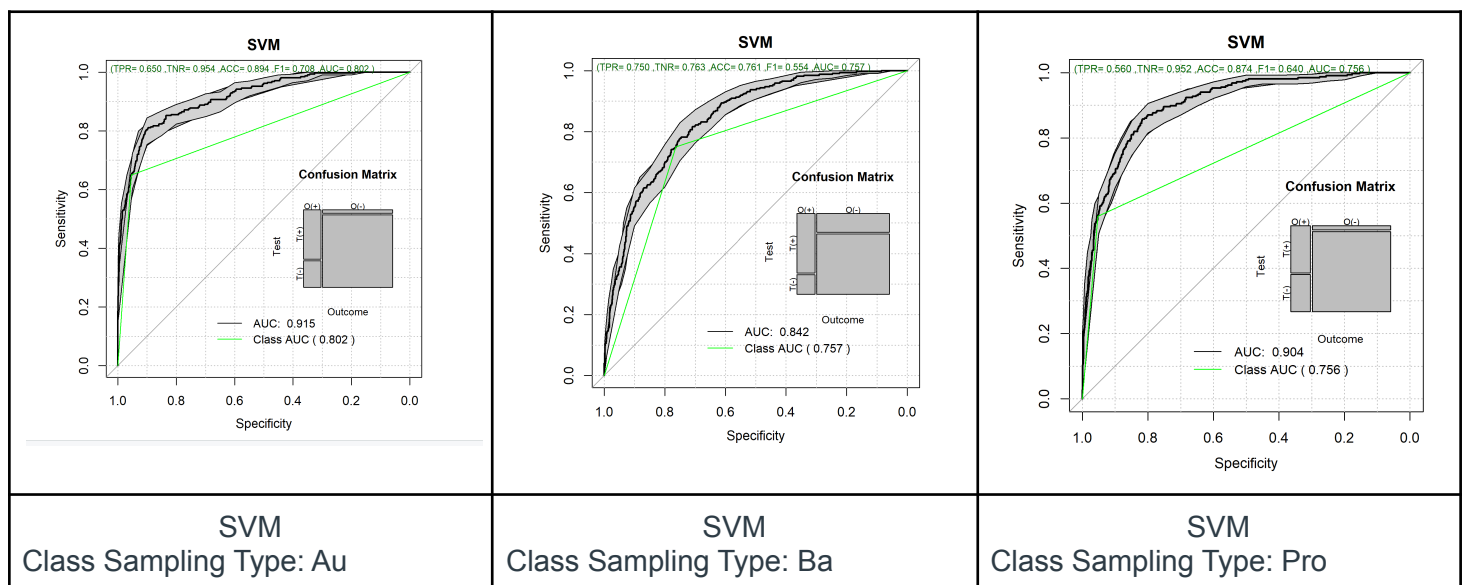


Figura 9: Método de SVM en R Studio

En la figura 10 se puede encontrar las gráficas resultantes al hacer uso del método de Naive Bayes y Naive Bayes BE. A simple vista se pueden encontrar diferencias en la figura de la curva, ya que para

Naive Bayes está curva se ve más aplanada que en Naive Bayes BE. Así mismo en Naive Bayes se tiene un área bajo la curva de 0.859, mientras que en Naive Bayes BE se tiene un área bajo la curva de 0.810.

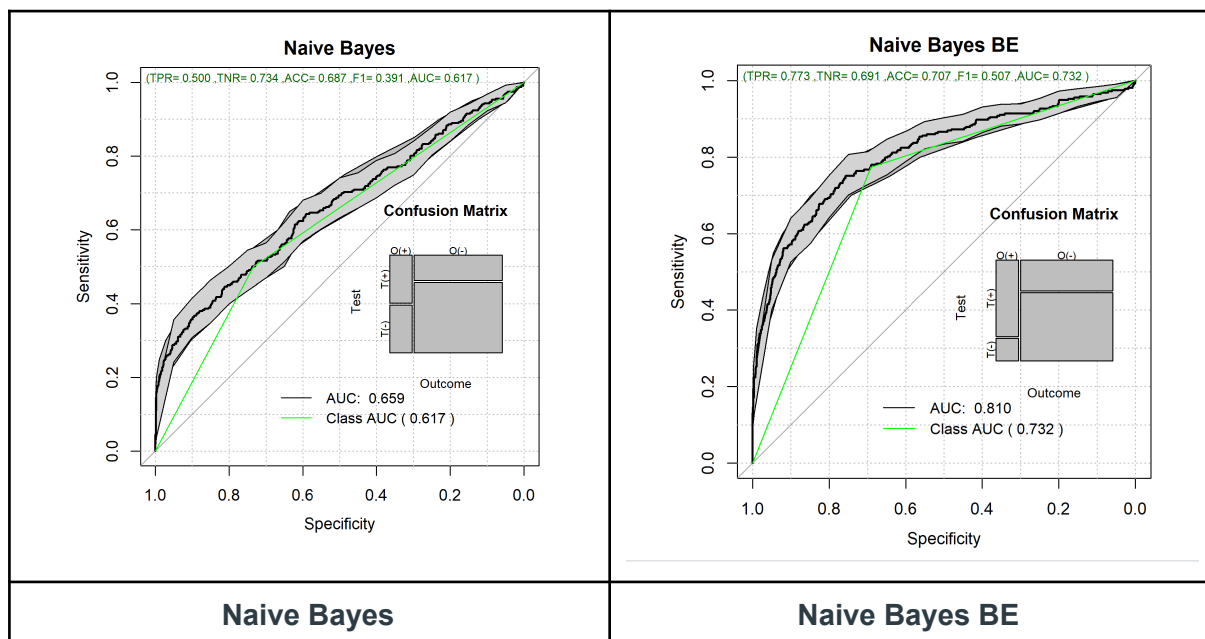


Figura 10: Naive Bayes en R Studio

La figura 11 muestra los resultados obtenidos al aplicar el método Adaboost en R Studio. En este método se obtuvo el valor del área bajo la curva de 0.939 y una clase de área bajo la curva de 0.769. Así mismo a simple vista se puede observar una curva de aprendizaje más marcada o curvada que en los demás métodos.

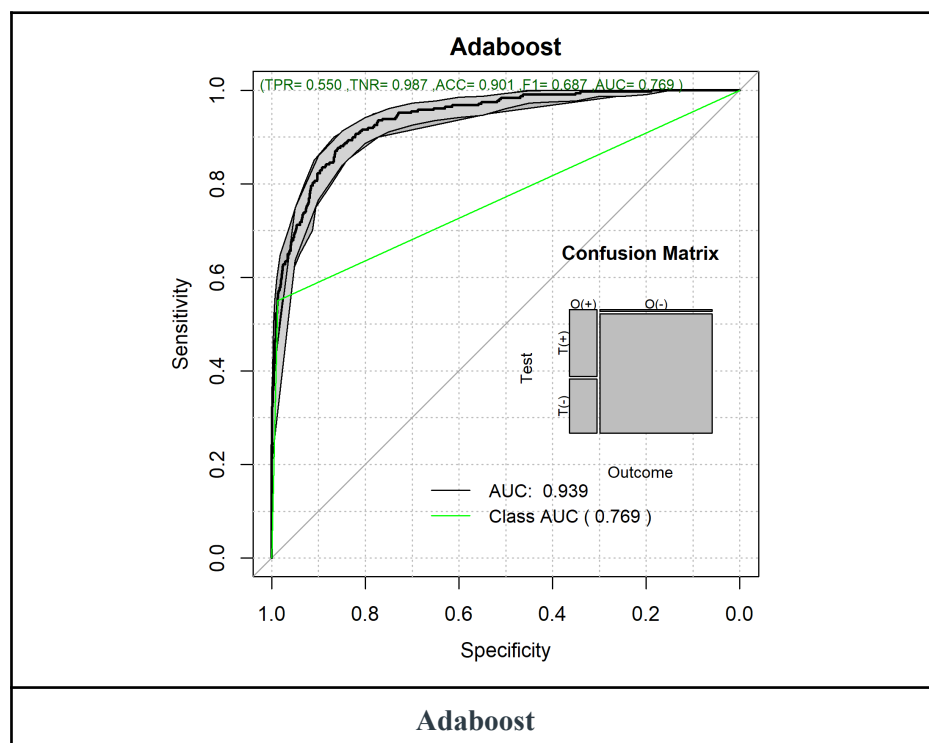


Figura 11: Adaboost en R Studio



Los resultados del método GBM pueden ser visualizados en la figura 12. En este método se puede encontrar un área bajo la curva de 0.926 y una clase de área bajo la curva de 0.754. Al igual que con el resultado obtenido por el método del adaboost, la curva generada es de las que más cerca se encuentran al eje y.

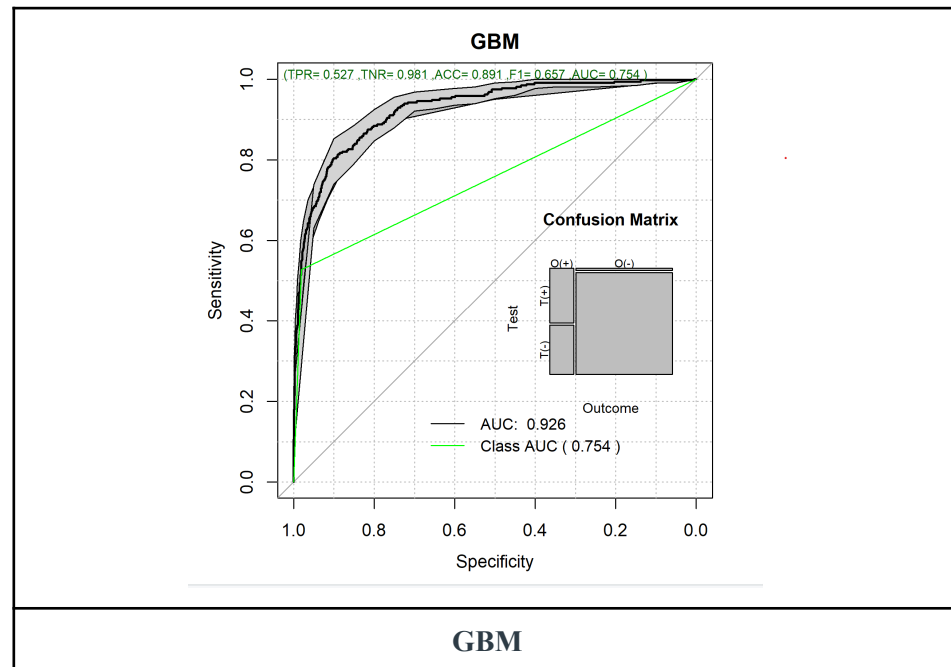


Figura 12: GBM en R Studio

## Discusión

Analizando los resultados obtenidos de tanto la segmentación como el análisis de datos hecho en R Studio, se pudo conocer los mejores métodos de machine learning para la identificación y detección de melanomas. Se encontró que el método Adaboost fue el mejor método que puede utilizarse para entrenar al modelo y obtener resultados óptimos de este; se determinó que el método Adaboost fue el mejor gracias a su área bajo la curva y la cercanía que está tenía al eje y. Además de contar con una curva de aprendizaje con un área bajo la curva de 0.939, siendo la más grande de todos los métodos, cuenta con buenos valores de TPR (True Positive Rate o Tasa Positiva Real ) y TNR 0.987 (True Negative Rate o Tasa Negativa Real). Se tiene que se cuenta con una sensibilidad o TPR de 0.550 , y una especificidad o TNR [14]. El modelo cuenta con una sensibilidad de 0.550, por otro lado, este mismo modelo cuenta con una especificidad de 0.987. Si bien estos valores son buenos, el valor de especificidad es fue el determinante clave en el modelo, ya que esto indica que el modelo tendrá menos probabilidad de detectar erróneamente melanomas cancerígenos.


Además del modelo Adaboost, otro factor crucial para el desarrollo de modelos que permitan la detección de melanomas fue la segmentación. Proceso que abarcaba desde la calibración y normalización de las imágenes para poder obtener una segmentación correcta. Si bien no todas las imágenes pudieron ser segmentadas de la manera esperada, la mayor parte de las imágenes en la base de datos si fueron segmentadas de la manera ideal.

## **Conclusiones**

Dentro de los últimos años se ha observado un gran desarrollo en las aplicaciones del machine learning, especialmente dentro del área de salud. Una de las aplicaciones que ha tenido mayor impacto y desarrollo, ha sido los algoritmos para el diagnóstico temprano de diferentes padecimientos, especialmente de cáncer de piel el cual cuenta con signos visuales que permiten un diagnóstico temprano.

El desarrollo de tecnologías como estas no solo facilitan el diagnóstico para el personal de salud, si no también acercan las herramientas de diagnóstico a los pacientes. Al convertirse en una herramienta clave para la detección temprana de melanomas que puedan evolucionar a cáncer de piel, incluso llegando a convertirse en una alternativa o complemento para los métodos de diagnóstico anteriores como biopsias [3]. Si bien durante esta actividad se pudo conocer un poco de las aplicaciones y el funcionamiento del machine learning y el entrenamiento de sus algoritmos de deep learning para la detección correcta y temprana, estas tecnologías deben seguirse desarrollando para ofrecer un diagnóstico al alcance de todo paciente que lo requiera.

## Referencias

- [1] “Cáncer de piel (no melanomatoso) - Estadísticas,” Cancer.Net, Oct. 07, 2021.  
<https://www.cancer.net/es/tipos-de-c%C3%A1ncer/c%C3%A1ncer-de-piel-no-melanomatoso/estad%C3%ADsticas> (accessed Jun. 10, 2022).
- [2] “Melanoma - Diagnóstico,” *Cancer.Net*, Jun. 03, 2020.  
<https://www.cancer.net/es/tipos-de-c%C3%A1ncer/melanoma/diagn%C3%B3stico> (accessed Jun. 10, 2022).
- [3] N. Almeida Cintra, “Diagnóstico clínico asistido por computadora: avances tecnológicos y su impacto social,” *Computer-aided clinical diagnosis : technological advances and their social impact*, Oct. 2021, Accessed: Jun. 10, 2022. [Online]. Available:  
<https://repositorio.uci.cu/jspui/handle/123456789/9878>
- [4] “Sistema de clasificación de imágenes de melanomas – Predictivos.”  
<http://datamining.dc.uba.ar/predictivos/?p=1074> (accessed Jun. 10, 2022).
- [5] O. Harrison, “Machine Learning Basics with the K-Nearest Neighbors Algorithm,” *Medium*, Jul. 14, 2019.  
<https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761> (accessed Jun. 10, 2022).
- [6] “What is the k-nearest neighbors algorithm? | IBM.” <https://www.ibm.com/topics/knn> (accessed Jun. 10, 2022)
- [7] L. Gonzalez, “Naive Bayes – Teoría,”  *Aprende IA*, Sep. 20, 2019.  
<https://aprendeia.com/naive-bayes-teoria-machine-learning/> (accessed Jun. 10, 2022).
- [8] “Cómo funcionan los clasificadores Naive Bayes: con ejemplos de código de Python.”  
<https://www.freecodecamp.org/espanol/news/como-funcionan-los-clasificadores-naive-bayes-con-ejemplos-de-codigo-de-python/> (accessed Jun. 10, 2022).
- [9] “Cómo funcionan los clasificadores Naive Bayes: con ejemplos de código de Python,” *freeCodeCamp.org*, Apr. 28, 2021.  
<https://www.freecodecamp.org/espanol/news/como-funcionan-los-clasificadores-naive-bayes-con-ejemplos-de-codigo-de-python/> (accessed Jun. 10, 2022).
- [10] R. Gandhi, “Support Vector Machine — Introduction to Machine Learning Algorithms,” *Medium*, Jul. 05, 2018.

<https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47> (accessed Jun. 10, 2022).

[11]

“AdaBoost | Interactive Chaos.”

<https://interactivechaos.com/es/manual/tutorial-de-machine-learning/adaboost> (accessed Jun. 10, 2022).

[12]

“¿Qué es el boosting? Guía de boosting en machine learning | AWS,” *Amazon Web Services, Inc.*

<https://aws.amazon.com/es/what-is/boosting/> (accessed Jun. 10, 2022).

[13]

“Gradient Boosting Machines · UC Business Analytics R Programming Guide.”

[http://uc-r.github.io/gbm\\_regression#proscons](http://uc-r.github.io/gbm_regression#proscons) (accessed Jun. 10, 2022).

[14]

D. H. Deshmukh, T. Ghorpade, and P. Padiya, “Improving classification using preprocessing and machine learning algorithms on NSL-KDD dataset,” in *2015 International Conference on Communication, Information & Computing Technology (ICCICT)*, Jan. 2015, pp. 1–6. doi: [10.1109/ICCICT.2015.7045674](https://doi.org/10.1109/ICCICT.2015.7045674).

## **Anexos**

*Valores obtenidos de los modelos en R Studio*

### **KNN**

Tested: 1610 Avg. Selected: 43.6 Min Tests: 1 Max Tests: 10 Mean Tests: 7.229814 . MAD: 0.4103218

### **cvSVM**

Tested: 1571 Avg. Selected: 214.7 Min Tests: 1 Max Tests: 7 Mean Tests: 3.087206 . MAD: 0.1352272

### **cvSVMba**

Tested: 1613 Avg. Selected: 43.4 Min Tests: 1 Max Tests: 10 Mean Tests: 7.216367 . MAD: 0.3399879

### **cvSVMPro**

Tested: 1578 Avg. Selected: 109.8 Min Tests: 1 Max Tests: 8 Mean Tests: 3.073511 . MAD: 0.1882148

### **NaiveBayes**

Tested: 1573 Avg. Selected: 147.5 Min Tests: 1 Max Tests: 9 Mean Tests: 3.08328 . MAD: 0.3134724

### **cvNBBE**

Tested: 1606 Avg. Selected: 37.5 Min Tests: 1 Max Tests: 10 Mean Tests: 7.247821 . MAD: 0.3384172

### **cvADA**

Tested: 1567 Avg. Selected: 0 Min Tests: 1 Max Tests: 8 Mean Tests: 3.095086 . MAD: 0.2426792

### **cvGBM**

Tested: 1569 Avg. Selected: 0 Min Tests: 1 Max Tests: 8 Mean Tests: 3.091141 . MAD: 0.175629

## **Códigos**

Los códigos utilizados en Matlab y R Studio se encuentran en GitHub