



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

**Evidencia**

## **Reporte formal actividad: Diagnóstico por Computadora**

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

**Grupo: 201**

<b>NOMBRE</b>	<b>MATRÍCULA</b>
Esther Aimeé Delgado Jiménez	A00827948
Alejandro Villalobos Zepeda	A01023762
Luis Orlando Santos Cruz	A00827603
Valentina Maldonado Gonzalez	A00827929
Andrea Cristina Cavero Arrivasplata	A01759770

**Equipo: 3**

**Profesor: Dr. Jose Tamez.**

## Introducción

Mundialmente cada año ocurren alrededor de 55,500 muertes por cáncer debido al melanoma cutáneo.[1] El melanoma es un tipo mortal de cáncer de piel y, aunque representa solo el 4% de todos los cánceres de piel, es responsable del 75% de todas las muertes por cáncer de piel.[1] En sus primeras etapas de diagnóstico el melanoma puede ser curado, pero si este se realiza de maneras tardías, el melanoma puede crecer más profundamente en la piel y propagarse sin control alguno. La presencia de los melanocitos en cualquier parte del cuerpo causan el melanoma y la exposición intensiva de la piel a la radiación ultravioleta es considerada la principal causa del melanoma.[2] Las herramientas de diagnóstico automático son de vital importancia para los médicos. Los dermatólogos expertos tienden a utilizar la dermatoscopia para el diagnóstico pero se estima que la precisión del diagnóstico de melanoma no es muy certera, alrededor de un 75%. [2]

Las herramientas de diagnóstico automático son de vital importancia para los médicos. Los dermatólogos expertos tienden a utilizar la dermatoscopia para el diagnóstico pero se estima que la precisión del diagnóstico de melanoma no es muy certera, alrededor de un 75%. El diagnóstico asistido por computadora es muy útil para aumentar la precisión y velocidad de dicho diagnóstico. [2]Ciertos softwares dentro de una computadora pueden extraer información relevante, como variación de color, asimetría, características de textura, que facilitan y ayudan a el diagnostico ya que muestra datos que no pueden ser fácilmente percibidos por los ojos humanos. [2]

Algunos pasos clave que se han establecido para el diagnóstico de melanoma se basan en el *computer-vision diagnosis* y entre ellos se encuentran: la adquisición de la imagen de la lesión de la piel, segmentación de la lesión cutánea a partir de la región de la piel, extracción de características de la lesión y clasificación de dichas características.[2] La segmentación o detección de bordes es el proceso de separar la lesión de la piel circundante para formar la región de interés. [2]

En el presente reporte se realizó una segmentación de las lesiones de piel y sus características, de igual manera se entrenó una máquina tradicional tomando como base características radiónicas para determinar el melanoma/tipo de lesión. Todo esto con el propósito de evaluar y reportar el desempeño de la correcta clasificación de melanoma de dos métodos de Aprendizaje por Computadora, mediante una matriz de confusión y curvas ROC de las dos Máquinas de aprendizaje.

## Métodos de Machine Learning:

KNN (K-Nearest-Neighbor): Es un algoritmo de Machine Learning basado en instancia de tipo supervisado, el cual puede utilizarse para clasificar nuevas muestras las cuales pueden ser valores discretos e igualmente, puede ser usado para predecir nuevos valores, es decir, hacer regresión. Se considera un método sencillo, lo cual lo hace ideal para Machine Learning, debido a que sirve específicamente para clasificar ciertos valores, mientras busca los puntos de datos más similares, los cuáles son aprendidos durante el entrenamiento del modelo. [3]

NB (Naive Bayes): Es un algoritmo de Machine Learning considerado simple. Este mismo, utiliza la regla de Bayes en conjunto con una fuerte suposición de que todos los atributos son independientes (condicionalmente) dada la clase. Dicha suposición de independencia, normalmente se ve violada en la práctica, sin embargo, Naive Bayes constantemente ofrece una precisión competitiva de clasificación. Normalmente, Naive Bayes es ampliamente aplicado en la práctica, debido a su eficiencia computacional y otras características ideales. [7]

SVM (Support vector machine): Es un algoritmo de Machine Learning supervisado basado en la teoría del aprendizaje estadístico y un tipo de técnica de machine learning. Este algoritmo selecciona las muestras de entrenamiento en un conjunto de subconjuntos característicos para que la clasificación del subconjunto de caracteres sea equivalente a la división de todo el conjunto de datos.[6] Se ha utilizado el SVM para resolver diferentes problemas de clasificación con éxito en muchas aplicaciones. Una de las ventajas de utilizar SVM es que realiza sus predicciones en un corto tiempo. SVM también es más capaz de resolver muestras más pequeñas, problemas de no linealidad y alta dimensionalidad en comparación con otros algoritmos.[6]

Adaboost: Es una técnica de Machine Learning que se desarrolló en 1997 y desde entonces se ha utilizado principalmente para abordar problemas de clasificación binaria. Este algoritmo se centra en mejorar el poder de predicción convirtiendo un número de alumnos débiles en alumnos fuertes. El principal objetivo del algoritmo adaboost es crear un modelo en la dataset de entrenamiento y luego realizar un segundo modelo para rectificar los errores que están presentes en el primer modelo. Este proceso se repite hasta que los errores se minimizan y se considera que la data set se predice de manera correcta. [4]

GBM (Understanding Gradient Boosting): Existe una gran similitud entre el algoritmo Adaboost y el de GBM. En los dos casos se crean diferentes modelos de entrenamiento para eliminar los errores que existen en los modelos creados y repetir el proceso hasta que el error de los modelos sea disminuido en su mayoría. La diferencia fundamental radica en que la principal diferencia entre ellos es cómo los dos algoritmos identifican las deficiencias de los modelos débiles (por ejemplo, árboles de decisión). Mientras que el modelo AdaBoost identifica las deficiencias mediante el uso de puntos de datos de alto peso, el aumento de gradiente realiza lo mismo mediante el uso de gradientes en la función de pérdida ( $y=ax+b+e$ , e necesita una mención especial ya que es el término de error). La función de pérdida es una medida que indica qué tan buenos son los coeficientes del modelo para ajustar los datos subyacentes. [5]

## **Metodología**

Primeramente, fue necesario realizar la descarga de la carpeta llamada “SKIN CANCER” que contiene distintos algoritmos de distintos softwares (como de RStudio y Matlab) y distintas imágenes.

Posteriormente, se realizó la búsqueda de los códigos de Matlab llamados “*ProcessMelanomaImages.m*” y “*loadAndProcessQuantitateImage.m*” en la carpeta “*SKINCANCER>MatlabScripts*” e igualmente la búsqueda de las imágenes en la carpeta de “*SKINCANCER>skin-lesions>train>melanoma*”. Una vez realizado esto, fue necesario correr el código de “*ProcessMelanomaImages.m*” asegurándose que la carpeta en la que se encuentra dicho código esté en el path de selección.

Siguiendo con el proceso, fue necesario descargar los softwares de RStudio y R, e igualmente, es importante que dentro del software de RStudio se descargue FRESA.CAD y los paquetes necesarios para su adecuado funcionamiento y así poder correr el código llamado “*MelanomaVsSeborrheic.Rmd*”.

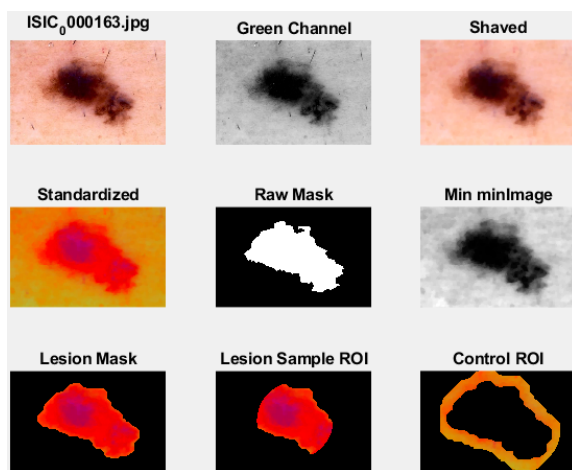
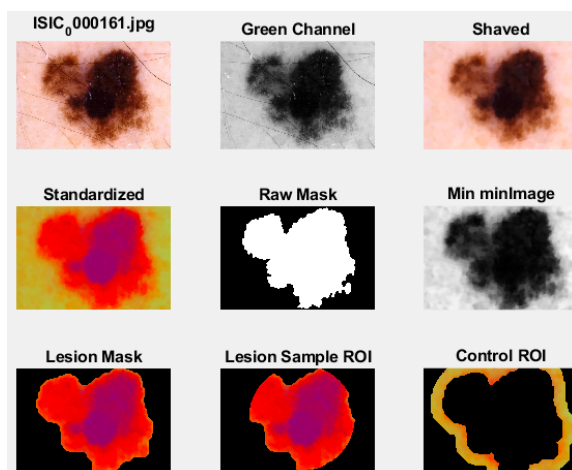
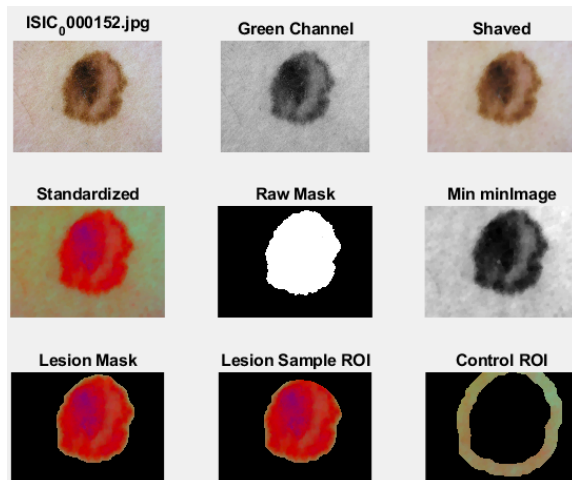
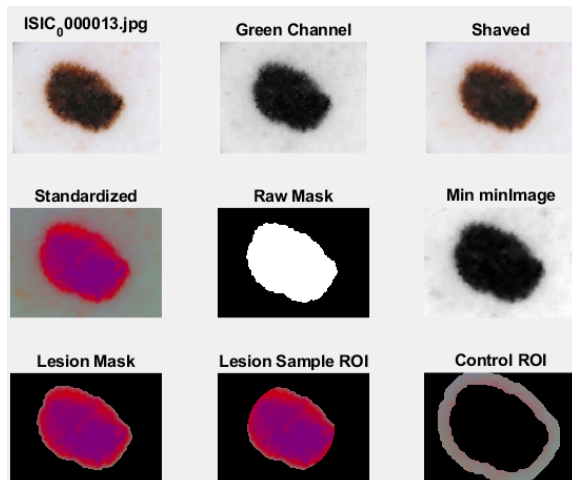
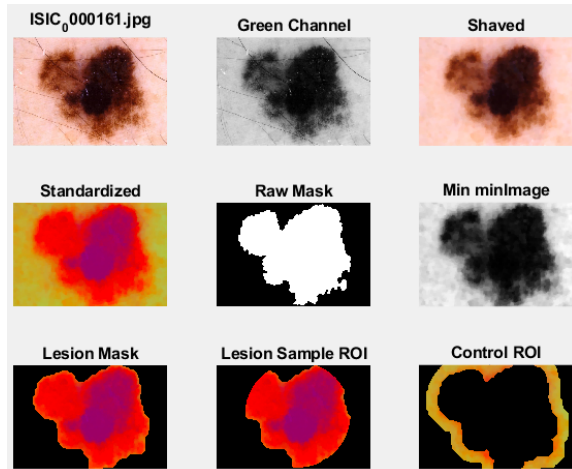
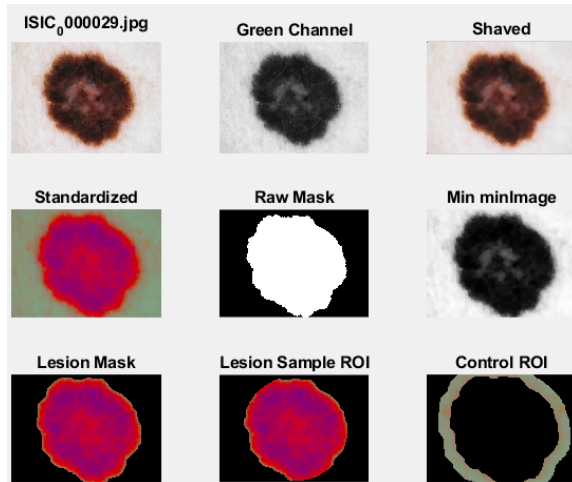
Comenzando con el código, es imprescindible que todas las segmentaciones deban estar balanceadas y también cambiar al path del mismo usuario del path de los archivos que están en la sección de “*Loading datasets*”. Y para finalizar con el proceso, es de vital importancia correr por secciones el código e ir tomando evidencia de las gráficas obtenidas, ya que, una vez obteniendo dichos resultados se hará una discusión de ellos. Es importante mencionar que, después de correr el código de las segmentaciones se pudieron extraer características (las cuales utilizan distintas medidas de radios de 2,4,8 y 16 píxeles) como señal (media, standard, deviation, skewness y kurtosis), textura (GLCM y Haralick features), fractal dimensions y forma (volumen, superficie, etc).

## **Resultados y discusión**

### **MATLAB**

Se realizó la segmentación de melanomas en MATLAB mediante Machine Learning. Para ello, primero se definió la ruta de la carpeta que contiene las imágenes a utilizar para el entrenamiento del programa. Luego, se definieron dos matrices en blanco para crear bases de datos a partir de las imágenes de lesión y control. En estas bases de datos se almacenaron parámetros importantes para el análisis estadístico como la media, desviación estándar, el área de la superficie del melanoma identificado, etc. Todo esto se logró con ayuda de la función `loadAndProcessQuantitateImage`. Esta función permite realizar el preprocesamiento y segmentación de las imágenes.

## Ejemplos de una buena segmentación



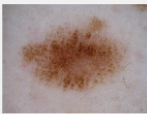
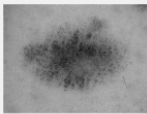
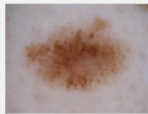



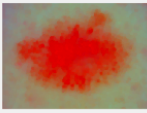

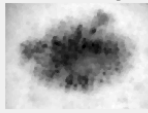


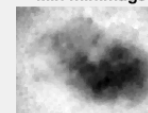
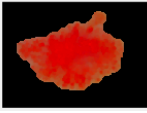
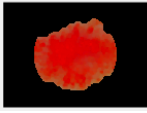
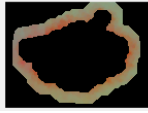



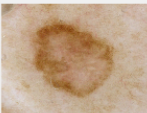
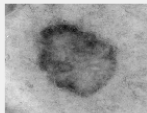

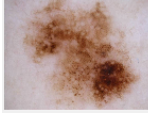
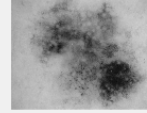
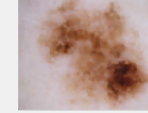
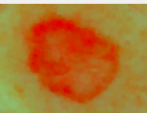

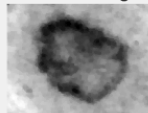
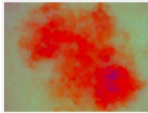

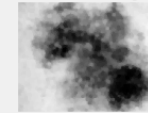



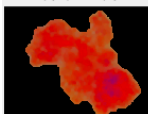
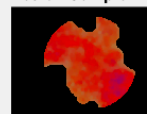
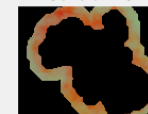
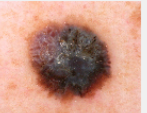
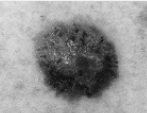
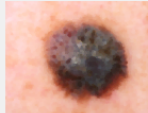
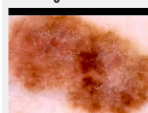
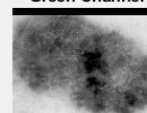
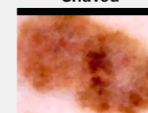
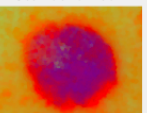

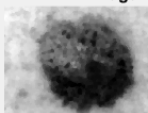
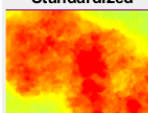


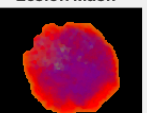
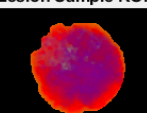




Ejemplos de una mala segmentación								
<b>ISIC<sub>0</sub>000022.jpg</b>			<b>ISIC<sub>0</sub>000026.jpg</b>					
	<b>Green Channel</b> 	<b>Shaved</b> 		<b>Green Channel</b> 	<b>Shaved</b> 			
<b>Standardized</b> 	<b>Raw Mask</b> 	<b>Min minImage</b> 	<b>Standardized</b> 	<b>Raw Mask</b> 	<b>Min minImage</b> 			
<b>Lesion Mask</b> 	<b>Lesion Sample ROI</b> 	<b>Control ROI</b> 	<b>Lesion Mask</b> 	<b>Lesion Sample ROI</b> 	<b>Control ROI</b> 			
<b>ISIC<sub>0</sub>000153.jpg</b>			<b>ISIC<sub>0</sub>000049.jpg</b>					
	<b>Green Channel</b> 	<b>Shaved</b> 		<b>Green Channel</b> 	<b>Shaved</b> 			
<b>Standardized</b> 	<b>Raw Mask</b> 	<b>Min minImage</b> 	<b>Standardized</b> 		<b>Min minImage</b> 			
<b>Lesion Mask</b> 	<b>Lesion Sample ROI</b> 	<b>Control ROI</b> 	<b>Lesion Mask</b> 	<b>Lesion Sample ROI</b> 	<b>Control ROI</b> 			
<b>ISIC<sub>0</sub>000142.jpg</b>			<b>ISIC<sub>0</sub>000160.jpg</b>					
	<b>Green Channel</b> 	<b>Shaved</b> 		<b>Green Channel</b> 	<b>Shaved</b> 			
<b>Standardized</b> 	<b>Raw Mask</b> 	<b>Min minImage</b> 	<b>Standardized</b> 	<b>Raw Mask</b> 	<b>Min minImage</b> 			
<b>Lesion Mask</b> 	<b>Lesion Sample ROI</b> 	<b>Control ROI</b> 	<b>Lesion Mask</b> 	<b>Lesion Sample ROI</b> 	<b>Control ROI</b> 			

Tabla 1. Resultados de segmentación

Las imágenes que presentaron una buena segmentación tienen la característica de que el melanoma identificado tiene un color oscuro, es uniforme, con volumen muy pequeño, y bordes definidos. En cambio, las segmentaciones identificadas como malas tienen como

característica que los melanomas analizados son irregulares en cuanto a la forma, no tienen bordes definidos, el color es más claro, y/o tienen volumen prominente.

Con base al código utilizado anteriormente para la comparación entre segmentación, se procedió a obtener las características extraídas como la señal, forma, textura y dimensiones fractales. En las siguientes tablas plasmamos los resultados obtenidos durante la recolección de estos datos.

<b>Media</b>	0.0614
<b>Desviación Estándar</b>	0.0922
<b>Skewness</b>	0.0659
<b>Kurtosis</b>	0.125

**Tabla 2.** Características de la señal

<b>Volumen</b>	3.4279e+4
<b>Superficie</b>	9.0089e+4
<b>Compacidad</b>	0.6491
<b>Gradiente</b>	244.4302

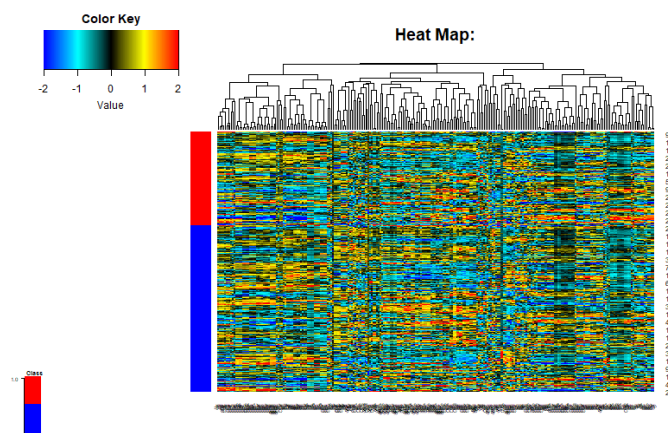
**Tabla 3.** Características de forma

<b>GLCM Slopes</b>	1.7084, 1.6368, 1.6726, 1.4793, 1.5581, 1.6082
<b>Media GLCM Slopes</b>	1.6106

**Tabla 4.** Características de Textura y dimensiones fractales

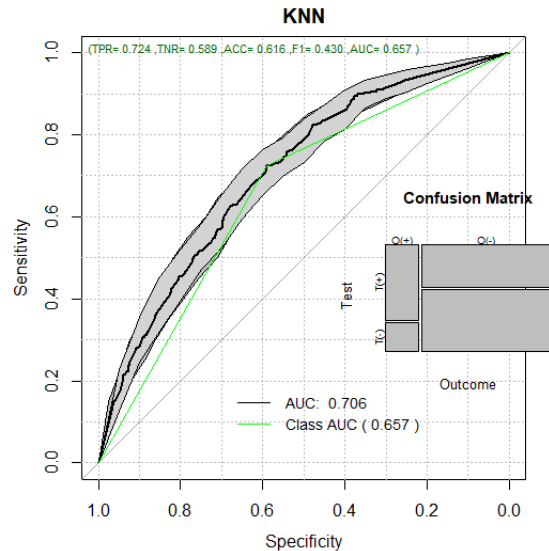
## R Studio

Con base a las simulaciones realizadas en R-Studio, se presentaron los siguientes resultados para los distintos métodos aplicados. El primero de ellos es el Heat Map (figura 1), en cual es una representación en dos dimensiones, que para entenderlo se utiliza un código de colores (“Color Key”) con la finalidad de observar si existe o no un patrón dentro de los datos.



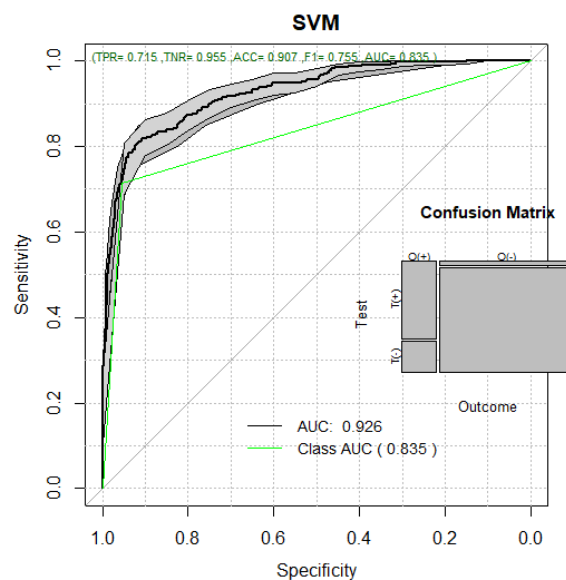
**Figura 1.** Heat Map





**Figura 2. Gráfica del Método KNN**

Para entender el comportamiento del método KNN (figura 2), debemos recurrir a analizar mediante la curva ROC y su matriz de confusión cuales son las características que se pueden interpretar a partir de la simulación. A través de esta comparamos la especificidad y sensibilidad, buscando el punto de corte. En este caso en especificidad se encuentra en 0.6 y la sensibilidad en 0.7, por lo que podemos determinar que este modelo es más sensible a tomar decisiones sin saber si está tomando buenas decisiones basadas en las especificaciones. Para comprobar este análisis, hacemos uso de dos coeficientes llamados F1 Score y AUC. El primero de ellos mientras más cercano a 1 significa que hay mayor precisión en el método, en este caso fue de 0.430. Mientras que el AUC, nos indica que tan confiable es el método, mientras más cercano a 1 es lo mejor, en este caso fue de 0.657.



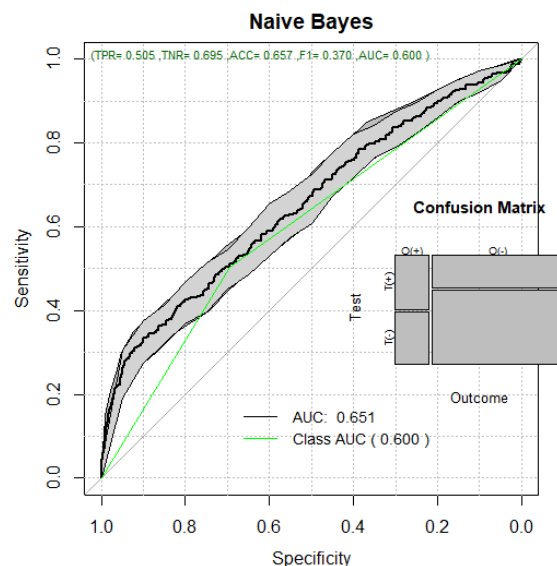
**Figura 3. Gráfica del Método SVM**

Continuando con el método SVM (figura 3), detectamos después de la simulación que la especificidad se encuentra en 1.0 y la sensibilidad en 0.7, por lo que podemos determinar que



este modelo es específico en la toma de decisiones, lo cual nos asegurará que sea más exacto con los sujetos que detecte con melanomas.

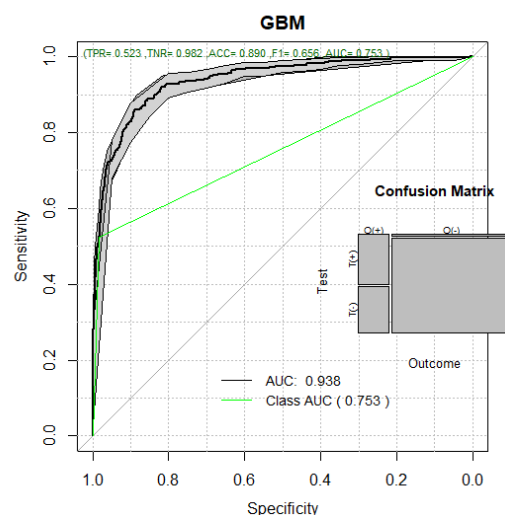
Para comprobar este análisis, hacemos uso de dos coeficientes llamados F1 Score y AUC. El primero de ellos mientras más cercano a 1 significa que hay mayor precisión en el método, en este caso fue de 0.755. Mientras que el AUC, nos indica que tan confiable es el método, mientras más cercano a 1 es lo mejor, en este caso fue de 0.835.



**Figura 4.** Gráfica del Método Naive Bayes

Continuando con el método Naive Bayes (figura 4), detectamos después de la simulación que la especificidad se encuentra en 0.7 y la sensibilidad en 0.5, por lo que podemos determinar que este modelo es específico en la toma de decisiones, lo cual nos asegurará que sea más exacto con los sujetos que detecte con melanomas.

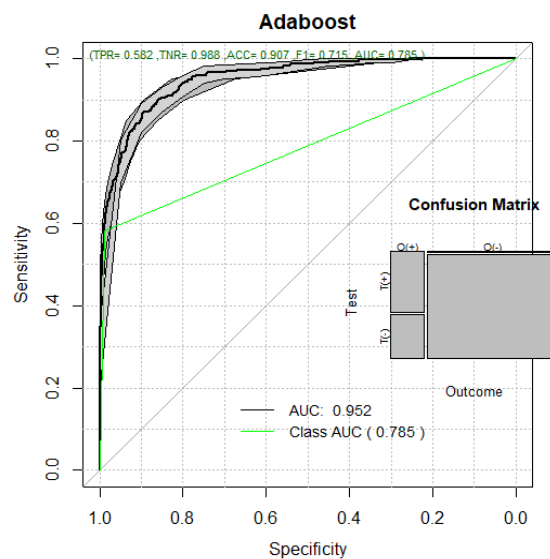
Para comprobar este análisis, hacemos uso de dos coeficientes llamados F1 Score y AUC. El primero de ellos mientras más cercano a 1 significa que hay mayor precisión en el método, en este caso fue de 0.370. Mientras que el AUC, nos indica que tan confiable es el método, mientras más cercano a 1 es lo mejor, en este caso fue de 0.600.



**Figura 5. Gráfica del Método GBM**

Continuando con el método GBM (figura 5), detectamos después de la simulación que la especificidad se encuentra en 1.0 y la sensibilidad en 0.5, por lo que podemos determinar que este modelo es específico en la toma de decisiones, lo cual nos asegurará que sea más exacto con los sujetos que detecte con melanomas.

Para comprobar este análisis, hacemos uso de dos coeficientes llamados F1 Score y AUC. El primero de ellos mientras más cercano a 1 significa que hay mayor precisión en el método, en este caso fue de 0.656. Mientras que el AUC, nos indica que tan confiable es el método, mientras más cercano a 1 es lo mejor, en este caso fue de 0.753.



**Figura 6. Gráfica del Método Adaboost**

Continuando con el método Adaboost (figura 6), detectamos después de la simulación que la especificidad se encuentra en 1.0 y la sensibilidad en 0.6, por lo que podemos determinar que este modelo es específico en la toma de decisiones, lo cual nos asegurará que sea más exacto con los sujetos que detecte con melanomas.

Para comprobar este análisis, hacemos uso de dos coeficientes llamados F1 Score y AUC. El primero de ellos mientras más cercano a 1 significa que hay mayor precisión en el método, en este caso fue de 0.715. Mientras que el AUC, nos indica que tan confiable es el método, mientras más cercano a 1 es lo mejor, en este caso fue de 0.785.

## Conclusión

A partir del análisis realizado de todos los métodos de machine learning aplicados para realizar la segmentación en esta actividad se pudo concluir que el método de SVM fue el que arrojó mejores resultados, ya que presenta una especificidad de 1.0 y 0.7 de sensibilidad. Mientras que el AUC arrojó un valor de 0.87. También es relevante mencionar que en la matriz de confusión los valores de verdaderos positivos fueron altos, de la misma manera que los valores de verdaderos negativos. Estos fueron los mejores resultados que se obtuvieron de los diferentes métodos analizados.

A pesar de esto es importante mencionar que de acuerdo con la investigación realizada de los diferentes algoritmos de machine learning que estamos empleando, el adaboost debería de ser el que presente mejores resultados. Se puede concluir que el motivo principal de que esto suceda es que se modificaron los parámetros de los parámetros que se tomarán para entrenar al algoritmo. En nuestro caso se utilizó un valor de 0.9 con una cantidad de 100 repeticiones y el método de “average”. Si se modifican estos parámetros probablemente se podrían obtener resultados diferentes y esto también demuestra la importancia de qué tipo de parámetros se le dan al algoritmo para que se resuelvan.

A partir de todo lo mencionado se puede concluir que ninguno de los métodos empleados tienen una especificidad ni sensibilidad suficiente para poder ser empleados en un contexto clínico en donde se realicen diagnósticos que estén completamente dependientes de lo que el programa arroja, ya que el número de falsos positivos es muy alto en cualquiera de los casos como para poder definir qué se hará un trabajo clínico seguro. A pesar de esto, se puede demostrar que el desarrollo de algoritmos especializados en el diagnóstico médico es un área de machine learning muy grande y de mucha oportunidad. Consideramos que si se trabaja en un algoritmo especializado para segmentación de imágenes médicas será posible tener resultados lo suficientemente buenos como para poder implementar en un ambiente clínico. A pesar de esto consideramos que también se puede realizar un estudio en donde se comparen la segmentaciones contra segmentaciones hechas por humanos, y así mejorar la capacidad del algoritmo y poder así pensar en implementarlo en un ambiente clínico.

## Referencias

- [1] Schadendorf, D., van Akkooi, A. C., Berking, C., Griewank, K. G., Gutzmer, R., Hauschild, A., ... & Ugurel, S. (2018). Melanoma. The Lancet, 392(10151), 971-984.
- [2] Jain, S., & Pise, N. (2015). Computer aided melanoma skin cancer detection using image processing. Procedia Computer Science, 48, 735-740.
- [3] Aprende Machine Learning. (2020, 15 julio). *Algoritmo k-Nearest Neighbor*. <https://www.aprendemachinelearning.com/clasificar-con-k-nearest-neighbor-ejemplo-en-python/>
- [4] AdaBoost Algorithm - A Complete Guide for Beginners - Analytics Vidhya. (2021). Retrieved 7 May 2022, from <https://www.analyticsvidhya.com/blog/2021/09/adaboost-algorithm-a-complete-guide-for-beginners/>
- [5] Understanding Gradient Boosting Machines. (2018). Retrieved 7 May 2022, from <https://towardsdatascience.com/understanding-gradient-boosting-machines-9be756fe76ab>
- [6] Abdullah, D. M., & Abdulazeez, A. M. (2021). Machine Learning Applications based on SVM Classification A Review. Qubahan Academic Journal, 1(2), 81-90.

[7] Webb, G., I. (2011). *Naïve Bayes*. SpringerLink. [https://link.springer.com/referenceworkentry/10.1007/978-0-387-30164-8\\_576?error=cookies\\_not\\_supported&code=93032c47-84ac-4aec-8ed9-4c5b5fb831f0](https://link.springer.com/referenceworkentry/10.1007/978-0-387-30164-8_576?error=cookies_not_supported&code=93032c47-84ac-4aec-8ed9-4c5b5fb831f0)

## Anexos

En la siguiente liga se puede acceder al github con el cual estuvimos trabajando en este proyecto: <https://github.com/AlejandroVZTec/Reto-Imagen>

## Código utilizado en Matlab:

```
melanomapath = 'E:\SKINCCANCER\skin-lesions\train\melanoma\';
allquantitativeLesionData = [];
allquantitativeControlData = [];

imageFile = 'ISIC_0000013.jpg'; loadAndProcessQuantitateImage;
imageFile = 'ISIC_0000022.jpg'; loadAndProcessQuantitateImage;
imageFile = 'ISIC_0000026.jpg'; loadAndProcessQuantitateImage;
imageFile = 'ISIC_0000029.jpg'; loadAndProcessQuantitateImage;
imageFile = 'ISIC_0000040.jpg'; loadAndProcessQuantitateImage;
imageFile = 'ISIC_0000043.jpg'; loadAndProcessQuantitateImage;
imageFile = 'ISIC_0000046.jpg'; loadAndProcessQuantitateImage;
imageFile = 'ISIC_0000049.jpg'; loadAndProcessQuantitateImage;
imageFile = 'ISIC_0000054.jpg'; loadAndProcessQuantitateImage;
imageFile = 'ISIC_0000056.jpg'; loadAndProcessQuantitateImage;
imageFile = 'ISIC_0000139.jpg'; loadAndProcessQuantitateImage;
imageFile = 'ISIC_0000141.jpg'; loadAndProcessQuantitateImage;
imageFile = 'ISIC_0000142.jpg'; loadAndProcessQuantitateImage;
imageFile = 'ISIC_0000143.jpg'; loadAndProcessQuantitateImage;
imageFile = 'ISIC_0000145.jpg'; loadAndProcessQuantitateImage;
imageFile = 'ISIC_0000148.jpg'; loadAndProcessQuantitateImage;
imageFile = 'ISIC_0000149.jpg'; loadAndProcessQuantitateImage;
imageFile = 'ISIC_0000150.jpg'; loadAndProcessQuantitateImage;
imageFile = 'ISIC_0000151.jpg'; loadAndProcessQuantitateImage;
imageFile = 'ISIC_0000152.jpg'; loadAndProcessQuantitateImage;
imageFile = 'ISIC_0000153.jpg'; loadAndProcessQuantitateImage;
imageFile = 'ISIC_0000155.jpg'; loadAndProcessQuantitateImage;
imageFile = 'ISIC_0000156.jpg'; loadAndProcessQuantitateImage;
imageFile = 'ISIC_0000157.jpg'; loadAndProcessQuantitateImage;
imageFile = 'ISIC_0000159.jpg'; loadAndProcessQuantitateImage;
imageFile = 'ISIC_0000160.jpg'; loadAndProcessQuantitateImage;
imageFile = 'ISIC_0000161.jpg'; loadAndProcessQuantitateImage;
imageFile = 'ISIC_0000162.jpg'; loadAndProcessQuantitateImage;
imageFile = 'ISIC_0000163.jpg'; loadAndProcessQuantitateImage;
imageFile = 'ISIC_0000164.jpg'; loadAndProcessQuantitateImage;
imageFile = 'ISIC_0000165.jpg'; loadAndProcessQuantitateImage;
imageFile = 'ISIC_0000166.jpg'; loadAndProcessQuantitateImage;
imageFile = 'ISIC_0000167.jpg'; loadAndProcessQuantitateImage;
imageFile = 'ISIC_0000169.jpg'; loadAndProcessQuantitateImage;
imageFile = 'ISIC_0000171.jpg'; loadAndProcessQuantitateImage;
imageFile = 'ISIC_0000172.jpg'; loadAndProcessQuantitateImage;
imageFile = 'ISIC_0000174.jpg'; loadAndProcessQuantitateImage;
imageFile = 'ISIC_0000176.jpg'; loadAndProcessQuantitateImage;
imageFile = 'ISIC_0000278.jpg'; loadAndProcessQuantitateImage;
imageFile = 'ISIC_0000285.jpg'; loadAndProcessQuantitateImage;
imageFile = 'ISIC_0000288.jpg'; loadAndProcessQuantitateImage;
```

[illegible]

[illegible]



[illegible]



[illegible]

```

imageFile = 'ISIC_0014324.jpg'; loadAndProcessQuantitateImage;
imageFile = 'ISIC_0014325.jpg'; loadAndProcessQuantitateImage;
imageFile = 'ISIC_0014327.jpg'; loadAndProcessQuantitateImage;
imageFile = 'ISIC_0014331.jpg'; loadAndProcessQuantitateImage;
imageFile = 'ISIC_0014337.jpg'; loadAndProcessQuantitateImage;
imageFile = 'ISIC_0014347.jpg'; loadAndProcessQuantitateImage;
imageFile = 'ISIC_0014357.jpg'; loadAndProcessQuantitateImage;
imageFile = 'ISIC_0014360.jpg'; loadAndProcessQuantitateImage;
imageFile = 'ISIC_0014361.jpg'; loadAndProcessQuantitateImage;
imageFile = 'ISIC_0014366.jpg'; loadAndProcessQuantitateImage;
imageFile = 'ISIC_0014372.jpg'; loadAndProcessQuantitateImage;
imageFile = 'ISIC_0014385.jpg'; loadAndProcessQuantitateImage;
imageFile = 'ISIC_0014395.jpg'; loadAndProcessQuantitateImage;
imageFile = 'ISIC_0014422.jpg'; loadAndProcessQuantitateImage;
imageFile = 'ISIC_0014476.jpg'; loadAndProcessQuantitateImage;
imageFile = 'ISIC_0014480.jpg'; loadAndProcessQuantitateImage;
imageFile = 'ISIC_0014486.jpg'; loadAndProcessQuantitateImage;
imageFile = 'ISIC_0014501.jpg'; loadAndProcessQuantitateImage;
imageFile = 'ISIC_0014507.jpg'; loadAndProcessQuantitateImage;
imageFile = 'ISIC_0014518.jpg'; loadAndProcessQuantitateImage;
imageFile = 'ISIC_0014525.jpg'; loadAndProcessQuantitateImage;
imageFile = 'ISIC_0014527.jpg'; loadAndProcessQuantitateImage;
imageFile = 'ISIC_0014543.jpg'; loadAndProcessQuantitateImage;
imageFile = 'ISIC_0014545.jpg'; loadAndProcessQuantitateImage;
imageFile = 'ISIC_0014557.jpg'; loadAndProcessQuantitateImage;
imageFile = 'ISIC_0014722.jpg'; loadAndProcessQuantitateImage;
imageFile = 'ISIC_0014735.jpg'; loadAndProcessQuantitateImage;
imageFile = 'ISIC_0014788.jpg'; loadAndProcessQuantitateImage;
imageFile = 'ISIC_0014795.jpg'; loadAndProcessQuantitateImage;
imageFile = 'ISIC_0014832.jpg'; loadAndProcessQuantitateImage;
imageFile = 'ISIC_0014891.jpg'; loadAndProcessQuantitateImage;
imageFile = 'ISIC_0014897.jpg'; loadAndProcessQuantitateImage;
imageFile = 'ISIC_0014919.jpg'; loadAndProcessQuantitateImage;
imageFile = 'ISIC_0014951.jpg'; loadAndProcessQuantitateImage;
imageFile = 'ISIC_0014975.jpg'; loadAndProcessQuantitateImage;
imageFile = 'ISIC_0014983.jpg'; loadAndProcessQuantitateImage;
imageFile = 'ISIC_0014987.jpg'; loadAndProcessQuantitateImage;
imageFile = 'ISIC_0015005.jpg'; loadAndProcessQuantitateImage;
imageFile = 'ISIC_0015045.jpg'; loadAndProcessQuantitateImage;
imageFile = 'ISIC_0015079.jpg'; loadAndProcessQuantitateImage;
imageFile = 'ISIC_0015082.jpg'; loadAndProcessQuantitateImage;
imageFile = 'ISIC_0015109.jpg'; loadAndProcessQuantitateImage;
imageFile = 'ISIC_0015110.jpg'; loadAndProcessQuantitateImage;
imageFile = 'ISIC_0015166.jpg'; loadAndProcessQuantitateImage;
imageFile = 'ISIC_0015189.jpg'; loadAndProcessQuantitateImage;
imageFile = 'ISIC_0015190.jpg'; loadAndProcessQuantitateImage;
imageFile = 'ISIC_0015200.jpg'; loadAndProcessQuantitateImage;
imageFile = 'ISIC_0015204.jpg'; loadAndProcessQuantitateImage;
imageFile = 'ISIC_0015219.jpg'; loadAndProcessQuantitateImage;
imageFile = 'ISIC_0015284.jpg'; loadAndProcessQuantitateImage;

writematrix(allquantitativeLesionData, 'MelanomaLesionFeatures.csv')
writematrix(allquantitativeControlData, 'MelanomaControlFeatures.csv')

```

## Código de R Studio

---

```

title: "MelanomavsSeborreic"
author: "Jose Tamez"
date: "4/9/2021"
output: html_document
editor_options:
  chunk_output_type: console
---
{r setup, include=FALSE}
knitr::opts_chunk$set(collapse = TRUE, warning = FALSE, message = FALSE,comment =
"#>")

## Melanoma vs Seborreic_Keratosis
{r functions,echo = FALSE }
library("FRESA.CAD")
library("e1071")
library("fastAdaboost")
library(gbm)
library(caret)

{r traincontrol}
tunningctrl <- trainControl(
  method = "repeatedcv",
  number = 5,
  repeats = 3
)
noTuningControl <- trainControl(method = "none")

## Loading data sets
{r FRESA Map, results = "hide", dpi=300, fig.height= 6.0, fig.width= 8.0}
op <- par(no.readonly = TRUE)
MelanomaFeatures <-
read.csv("D:/MATLAB/SKINCCANCER/MatlabScripts/MelanomaLesionFeatures.csv",
header=FALSE)
SeborrheicFeatures <-
read.csv("D:/MATLAB/SKINCCANCER/MatlabScripts/SeborrheicLesionFeatures.csv",
header=FALSE)
NevusFeatures <-
read.csv("D:/MATLAB/SKINCCANCER/MatlabScripts/NevusLesionFeatures.csv",
header=FALSE)
MelanomaControlFeatures <-
read.csv("D:/MATLAB/SKINCCANCER/MatlabScripts/MelanomaControlFeatures.csv",
header=FALSE)
SeborrheicControlFeatures <-
read.csv("D:/MATLAB/SKINCCANCER/MatlabScripts/SeborrheicControlFeatures.csv",
header=FALSE)
NevusControlFeatures <-
read.csv("D:/MATLAB/SKINCCANCER/MatlabScripts/NevusControlFeatures.csv",
header=FALSE)
colnames(MelanomaControlFeatures) <-
paste("C",colnames(MelanomaControlFeatures),sep="")

```

```

colnames(SeborrheicControlFeatures)
paste("C",colnames(SeborrheicControlFeatures),sep="")
colnames(NevusControlFeatures) <- paste("C",colnames(NevusControlFeatures),sep="")
sum(is.na(MelanomaFeatures))
sum(is.na(SeborrheicFeatures))
sum(is.na(NevusFeatures))
sum(is.na(MelanomaControlFeatures))
sum(is.na(NevusControlFeatures))
channel1 <- c(1:32)
channel2 <- c(33:64)
channel3 <- c(65:96)
MMelanomaFeatures <- ( MelanomaFeatures[,channel1] +
                        MelanomaFeatures[,channel2] +
                        MelanomaFeatures[,channel3] )/3.0
SMelanomaFeatures <- ( abs(MelanomaFeatures[,channel1] - MMelanomaFeatures) +
                        abs(MelanomaFeatures[,channel2] - MMelanomaFeatures) +
                        abs(MelanomaFeatures[,channel3] - MMelanomaFeatures) )/3.0
colnames(MMelanomaFeatures) <- paste("M",colnames(MMelanomaFeatures),sep="")
colnames(SMelanomaFeatures) <- paste("S",colnames(SMelanomaFeatures),sep="")
MelanomaFeatures
cbind(MelanomaFeatures,SMelanomaFeatures/(0.01+abs(MMelanomaFeatures)))
MNevusFeatures <- ( NevusFeatures[,channel1] +
                    NevusFeatures[,channel2] +
                    NevusFeatures[,channel3] )/3.0
SNevusFeatures <- ( abs(NevusFeatures[,channel1] - MNevusFeatures) +
                    abs(NevusFeatures[,channel2] - MNevusFeatures) +
                    abs(NevusFeatures[,channel3] - MNevusFeatures) )/3.0
colnames(MNevusFeatures) <- paste("M",colnames(MNevusFeatures),sep="")
colnames(SNevusFeatures) <- paste("S",colnames(SNevusFeatures),sep="")
NevusFeatures <- cbind(NevusFeatures,SNevusFeatures/(0.001+abs(MNevusFeatures)))
MSeborrheicFeatures <- ( SeborrheicFeatures[,channel1] +
                        SeborrheicFeatures[,channel2] +
                        SeborrheicFeatures[,channel3] )/3.0
SSeborrheicFeatures <- ( abs(SeborrheicFeatures[,channel1] - MSeborrheicFeatures) +
                        abs(SeborrheicFeatures[,channel2] - MSeborrheicFeatures) +
                        abs(SeborrheicFeatures[,channel3] - MSeborrheicFeatures) )/3.0
colnames(MSeborrheicFeatures) <- paste("M",colnames(MSeborrheicFeatures),sep="")
colnames(SSeborrheicFeatures) <- paste("S",colnames(SSeborrheicFeatures),sep="")
SeborrheicFeatures
cbind(SeborrheicFeatures,SSeborrheicFeatures/(0.001+abs(MSeborrheicFeatures)))
MMelanomaControlFeatures <- ( MelanomaControlFeatures[,channel1] +
                              MelanomaControlFeatures[,channel2] +
                              MelanomaControlFeatures[,channel3] )/3.0
SMelanomaControlFeatures <- (abs(MelanomaControlFeatures[,channel1]
MMelanomaControlFeatures) +
                              abs(MelanomaControlFeatures[,channel2] - MMelanomaControlFeatures)
+
                              abs(MelanomaControlFeatures[,channel3] -
MMelanomaControlFeatures))/3.0

```

```

colnames(MMelanomaControlFeatures) <-
paste("M",colnames(MMelanomaControlFeatures),sep="")
colnames(SMelanomaControlFeatures) <-
paste("S",colnames(SMelanomaControlFeatures),sep="")
MelanomaControlFeatures <-
cbind(MelanomaControlFeatures,SMelanomaControlFeatures/(0.001+abs(MMelanomaContr
olFeatures)))
MNevusControlFeatures <- ( NevusControlFeatures[,channel1] +
      NevusControlFeatures[,channel2] +
      NevusControlFeatures[,channel3] )/3.0
SNevusControlFeatures <- (abs(NevusControlFeatures[,channel1] - MNevusControlFeatures)
+
      abs(NevusControlFeatures[,channel2] - MNevusControlFeatures) +
      abs(NevusControlFeatures[,channel3] - MNevusControlFeatures))/3.0
colnames(MNevusControlFeatures) <-
paste("M",colnames(MNevusControlFeatures),sep="")
colnames(SNevusControlFeatures) <- paste("S",colnames(SNevusControlFeatures),sep="")
NevusControlFeatures <-
cbind(NevusControlFeatures,SNevusControlFeatures/(0.001+abs(MNevusControlFeatures)))
MSeborrheicControlFeatures <- ( SeborrheicControlFeatures[,channel1] +
      SeborrheicControlFeatures[,channel2] +
      SeborrheicControlFeatures[,channel3] )/3.0
SSeborrheicControlFeatures <- (abs(SeborrheicControlFeatures[,channel1]
MSeborrheicControlFeatures) +
      abs(SeborrheicControlFeatures[,channel2] - MSeborrheicControlFeatures)
+
      abs(SeborrheicControlFeatures[,channel3] -
MSeborrheicControlFeatures))/3.0
colnames(MSeborrheicControlFeatures) <-
paste("M",colnames(MSeborrheicControlFeatures),sep="")
colnames(SSeborrheicControlFeatures) <-
paste("S",colnames(SSeborrheicControlFeatures),sep="")
SeborrheicControlFeatures <- cbind(SeborrheicControlFeatures,

SSeborrheicControlFeatures/(0.001+abs(MSeborrheicControlFeatures)))
CtrDiff <- MelanomaFeatures[,1:ncol(MelanomaControlFeatures)] -
MelanomaControlFeatures;
colnames(CtrDiff) <- colnames(MelanomaControlFeatures)
MelanomaFeatures <- cbind(MelanomaFeatures,CtrDiff)
CtrDiff <- NevusFeatures[,1:ncol(MelanomaControlFeatures)] - NevusControlFeatures;
colnames(CtrDiff) <- colnames(NevusControlFeatures)
NevusFeatures <- cbind(NevusFeatures,CtrDiff)
CtrDiff <- SeborrheicFeatures[,1:ncol(MelanomaControlFeatures)] -
SeborrheicControlFeatures;
colnames(CtrDiff) <- colnames(SeborrheicControlFeatures)
SeborrheicFeatures <- cbind(SeborrheicFeatures,CtrDiff)
MelanomaFeatures <- MelanomaFeatures[complete.cases(MelanomaFeatures),]
NevusFeatures <- NevusFeatures[complete.cases(NevusFeatures),]
SeborrheicFeatures <- SeborrheicFeatures[complete.cases(SeborrheicFeatures),]
MelanomaFeatures$class <- rep(1,nrow(MelanomaFeatures))

```

```

NevusFeatures$Class <- rep(0,nrow(NevusFeatures))
SeborrheicFeatures$Class <- rep(0,nrow(SeborrheicFeatures))
MelanomaSeborrheicNevus <- rbind(MelanomaFeatures,SeborrheicFeatures,NevusFeatures)
#MelanomaSeborrheicNevus <- rbind(MelanomaFeatures,SeborrheicFeatures)
#MelanomaSeborrheicNevus <- rbind(MelanomaFeatures,NevusFeatures)
table(MelanomaSeborrheicNevus$Class)

```

```
## The Heatmap
```

```

{r, dpi=300, fig.height= 6.0, fig.width= 8.0}
par(op)
boxplot(MelanomaSeborrheicNevus)
hm <- heatMaps(Outcome = "Class",
#       data = MelanomaSeborrheicNevus,
       data = MelanomaSeborrheicNevus[1:900,],
       title = "Heat Map:",Scale = TRUE,
       hCluster = "col",cexRow = 0.75,cexCol = 0.5,srtCol = 45)
par(op)

```

```
## Learning Melanoma with KNN
```

```

{r}
cvKNN <- randomCV(MelanomaSeborrheicNevus,"Class",
  KNN_method,
  trainFraction = 0.70,
  repetitions = 100,
  classSamplingType = "Ba",
  featureSelectionFunction = univariate_KS,
  featureSelection.control = list(pvalue=0.05,limit= -1),
  kn=5
)

```

```
## Plot performance
```

```

{r}
performace <- predictionStats_binary(cvKNN$medianTest,"KNN")
par(op)

```

```
{r}
```

```

cvSVM <- randomCV(MelanomaSeborrheicNevus,"Class",
  e1071::svm,
  asFactor = TRUE,
  trainFraction = 0.70,
  repetitions = 100,
  classSamplingType = "Ba",
  featureSelectionFunction = univariate_KS,
  featureSelection.control = list(pvalue=0.05,limit= -1),
  probability=TRUE
)

```

```
## SVM performance
```

```

{r}
performace <- predictionStats_binary(cvSVM$medianTest,"SVM")

```

```
par(op)
```

```
## Learning Melanoma with Naive Bayes
```

```
{r}  
cvNB <- randomCV(MelanomaSeborrheicNevus,"Class",  
  NAIVE_BAYES,  
  trainFraction = 0.70,  
  repetitions = 100,  
  classSamplingType = "Ba",  
  featureSelectionFunction = univariate_KS,  
  featureSelection.control = list(pvalue=0.05,limit=0.1),  
#    pca=FALSE  
)
```

```
## Plot performance
```

```
{r}  
performace <- predictionStats_binary(cvNB$medianTest,"Naive Bayes")  
par(op)
```

```
## Learning Melanoma with Adaboost
```

```
{r}  
cvADA <- randomCV(MelanomaSeborrheicNevus,"Class",  
  adaboost,  
  trainFraction = 0.70,  
  repetitions = 100,  
  classSamplingType = "Ba",  
  nIter=50)
```

```
## Plot performance
```

```
{r}  
performace <- predictionStats_binary(cvADA$medianTest,"Adaboost")  
par(op)
```

```
{r}  
cvgbm <- randomCV(MelanomaSeborrheicNevus,"Class",  
  train,  
  asFactor = TRUE,  
  trainFraction = 0.70,  
  repetitions = 100,  
  classSamplingType = "Ba",  
  method = "gbm",  
  trControl = tuningctrl,  
#    preProc = c("center", "scale"),  
  verbose = FALSE  
)
```

```
{r}  
performace <- predictionStats_binary(cvgbm$medianTest,"GBM")  
par(op)
```