

Análisis de imágenes de lesiones causadas por Cáncer de piel: Una aproximación al diagnóstico no supervisado de la gravedad de la enfermedad

Descripción de los datos

Los datos que se van a usar para el desarrollo de este proyecto es una recopilación parcial de todas las imágenes con las que cuenta ISIC, y está compuesta por un total de 2.357 imágenes clasificadas en los diferentes tipos de posibles cánceres de piel (9 clases), siguiendo la distribución que se muestra en la siguiente gráfica:

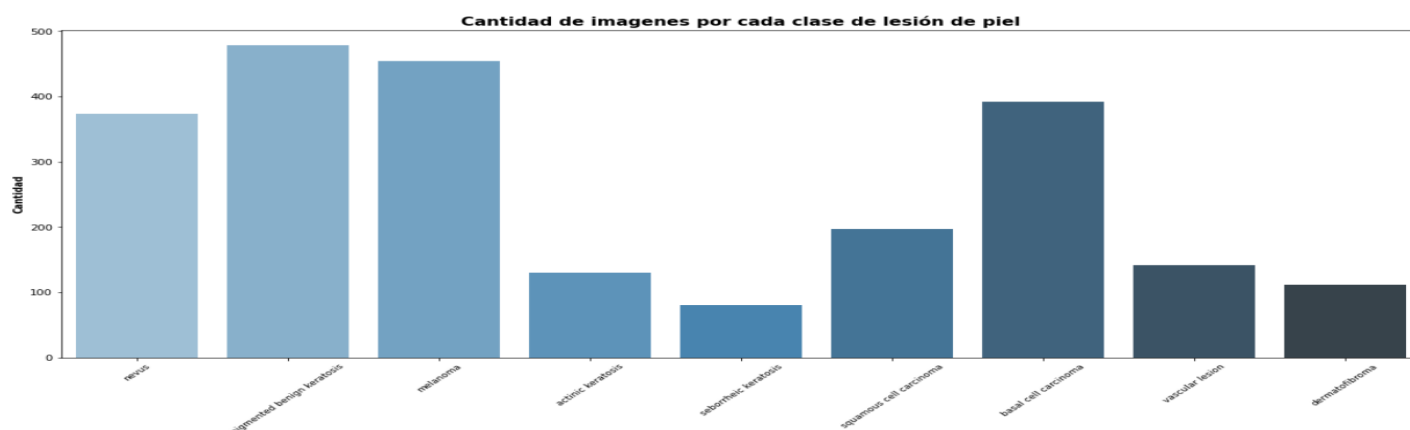


Ilustración 1- Cantidad de imágenes por cada clase de lesión de piel

Number of Train Images 2239

Number of Test Images 118

#####

Train Images Breakdown {'actinic keratosis': 114, 'basal cell carcinoma': 376, 'dermatofibroma': 95, 'melanoma': 438, 'nevus': 376, 'pigmented benign keratosis': 462, 'seborrheic keratosis': 77, 'squamous cell carcinoma': 181, 'vascular lesion': 139}

Test Images Breakdown {'actinic keratosis': 16, 'basal cell carcinoma': 16, 'dermatofibroma': 16, 'melanoma': 16, 'nevus': 16, 'pigmented benign keratosis': 3, 'seborrheic keratosis': 3, 'squamous cell carcinoma': 16, 'vascular lesion': 3}

#####

Classes ['actinic keratosis', 'seborrheic keratosis', 'basal cell carcinoma', 'pigmented benign keratosis', 'nevus', 'squamous cell carcinoma', 'melanoma', 'vascular lesion', 'dermatofibroma']

Ilustración 2- Cantidad de imágenes por cada clase de lesión de piel y por subset (Train/Test)

Por tal motivo, no es necesario realizar limpieza de los datos. Los datos de entrada son imágenes que están compuestas cada una por 3 canales representados en RGB (valores numéricos), y cada una presenta dimensiones de 224*224, generando así un total de 150.528 píxeles de información para cada una. Es posible entonces afirmar que, el dataset resultante tiene un tamaño de (2.357, 150.528). No se cuenta con variables categóricas ni tampoco es necesario imputar datos faltantes.

Dada la dimensión de datos y para realizar la lectura de las fotos en Python se realizó un proceso inicial de reducción de la dimensión de estas, con el apoyo de la función de image_dataset_from_directory de la librería TensorFlow. Esta permite redimensionar el tamaño de cada imagen y agruparlos en un objeto batch. Con esto, se modifica el dataset de entrada y se decide reducir las dimensiones de las imágenes a un tamaño de 150*150, generando así un total de 67.500 píxeles de información para cada imagen. Es posible entonces afirmar que, el dataset resultante final tiene un tamaño de (2.357, 67.500).

Propuesta Metodológica

Tal y como se planteó en la motivación del proyecto, la solución a la problemática en la identificación temprana de melanoma en las personas podría tener una solución basada en metodologías de *Aprendizaje de Máquina No Supervisado*.

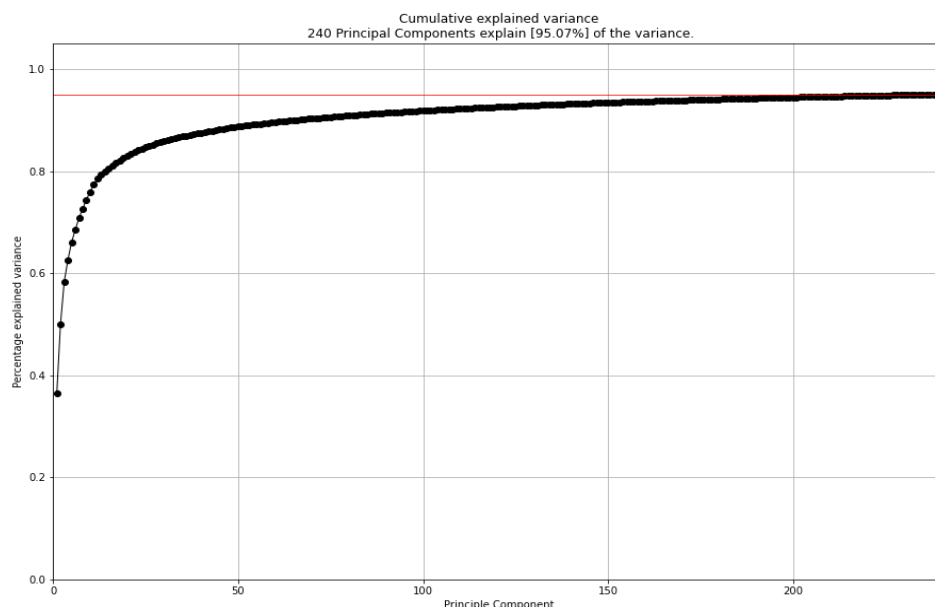
La primera motivación para hacer uso de estas metodologías o algoritmos es debido a las dimensiones que componen cada una de las imágenes, y al tamaño comprendido por todas las imágenes para realizar el entrenamiento de cualquier modelo. Se parte del concepto de que el tamaño que compone cada una de las imágenes es demasiado grande al estar compuestas por un formato RGB, y que, al tener un total de 2.357 imágenes en total, es imposible para un computador tradicional hacer el procesamiento completo del total de los datos. Es por esta razón, que la primera aplicación será la implementación de un algoritmo de **PCA** para buscar reducir el tamaño de los datos a entrenar, intentando penalizar por máximo el 5% total de la varianza de las fotos completas, y de esta forma poder correr de forma óptima los diferentes modelos de clustering.

La segunda aplicación propuesta es la implementación de tres modelos diferentes de Clustering, entre los cuales tenemos uno de **K-Medoides** (para grupos convexos); otro de **clustering jerárquico** (para grupos no convexos); y uno final de **DBSCAN** (para obtener un resultado diferente con grupos no convexos). Dichos modelos se implementarán con la intención de lograr agrupar las imágenes de las lesiones de forma tal que los patrones similares entre estas nos permitan dar una alerta temprana de la presencia de un cáncer de piel maligno o benigno en una persona, o en caso similar, servir a un médico especialista para apoyarse en su toma de decisiones. Es de aclarar que se parte de la idea de que el etiquetado manual de estas imágenes requiere de la revisión de un especialista en el tema, y que al ser tantas imágenes requiere de un costo en tiempo significativamente alto, por lo cual se intenta no acudir a algoritmos de aprendizaje de maquina supervisados, sin embargo, aprovechando que la base que vamos a utilizar para este ejercicio se encuentra etiquetada, estas etiquetas van a ser usadas como métrica para ver qué tan bien se agrupan las imágenes con los diferentes modelos propuestos.

Resultados Parciales

Reducción con PCA, buscando explicar al menos el 95% de la varianza de los datos originales:

Ilustración 3 – Resultado Parcial de PCA



Con lo anterior, se confirma que, utilizando 240 componentes principales y siguiendo el criterio de la varianza explicada, es posible explicar el 95% de la varianza del conjunto total de imágenes de entrenamiento, que como se ha dicho anteriormente, tienen tamaño 150*150.

Luego, utilizando las imágenes reducidas utilizando solo 240 componentes principales, es posible realizar un primer análisis preliminar de clustering utilizando la librería AgglomerativeClustering. Seleccionamos 9 clusters porque son 9 clases las que componen el dataset de entrada. En futuras entregas, validaremos el resultado obtenido con diferentes valores de affinity y linkage.

```
from sklearn.cluster import AgglomerativeClustering
from sklearn.decomposition import PCA

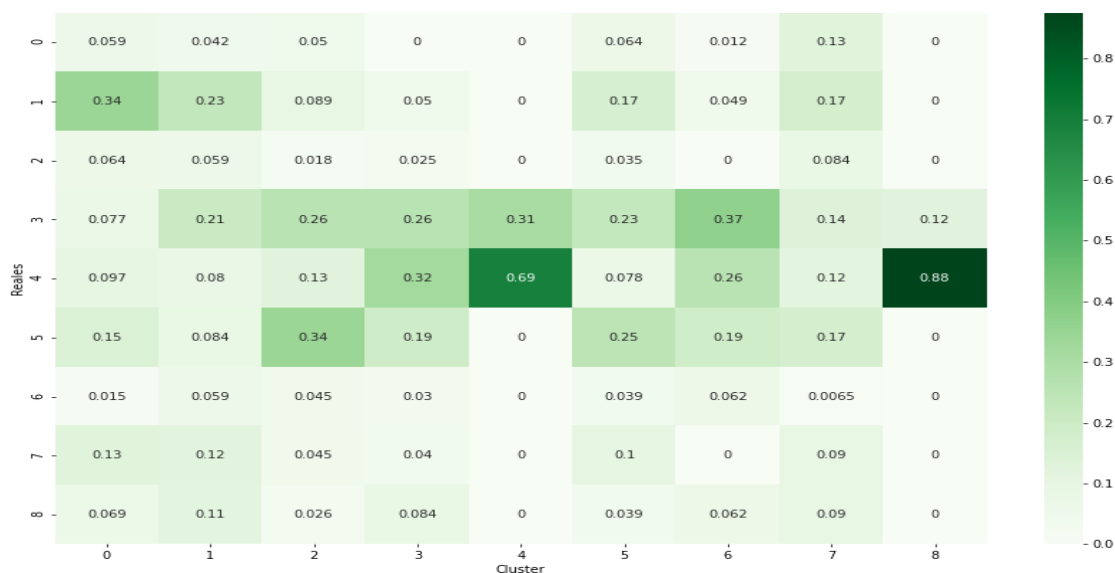
cluster_sk = AgglomerativeClustering(n_clusters=9, affinity='euclidean', linkage='ward')
cluster_sk = cluster_sk.fit_predict(PCA(n_components = 240).fit_transform(X_Train_df))
```

Posteriormente podemos validar por medio de una matriz de confusión que tan bien o mal han quedado distribuidas las imágenes en los clusters (recordando que tenemos marcadas todas las imágenes):

```
from sklearn.metrics import confusion_matrix

cf_matrix = confusion_matrix(y_Train_df, cluster_sk)

fig = plt.figure(figsize = (15, 10), dpi = 70)
s = sb.heatmap(cf_matrix/np.sum(cf_matrix, axis = 0), annot=True, vmin=0, cmap="Greens")
s.set(xlabel='Cluster', ylabel='Reales')
```



Los resultados parciales obtenidos muestran que, por ejemplo, la clase 4 está siendo incluida principalmente en el cluster 4. Las demás clases son agrupadas en clusters diferentes a las que le corresponden. De momento, el resultado parcial obtenido es bastante pobre. Como trabajo futuro, se van a intentar agrupar en menos cantidad de clusters para evaluar los resultados obtenidos de esa forma.