




# DermaScan

Juanjo Medina  
Jesús Cánovas

# Índice

- 1 - Justificación y descripción del proyecto
  - 2 - Obtención de los datos
  - 3 - Limpieza de datos (Preprocesado)
  - 4 - Exploración y visualización de los datos
  - 5 - Preparación de los datos para Machine Learning
  - 6 - Entrenamiento del modelo y comprobación del rendimiento
  - 7 - Procesamiento de Lenguaje Natural
  - 8 - Aplicación web
  - 9 - Conclusiones
  - 10 - Bibliografía
- 



# 1. Justificación y descripción del proyecto

DermaScan es una plataforma web dedicada a la detección de lesiones de la piel.

Nuestro objetivo es que la detección temprana de problemas de piel sea accesible y efectiva.

Los usuarios podrán obtener resultados precisos sobre la naturaleza del tejido cutáneo en la imagen.

Brindando información crucial para la toma de decisiones médicas.

## 2 - Obtención de los datos

- Objetos y imágenes variadas: <https://www.kaggle.com/datasets/greg115/various-tagged-images>
- Piel sanas: <https://www.kaggle.com/datasets/thanatw/isic2019-modded>
- Piel con cancer:  
<https://www.kaggle.com/datasets/hasnainjaved/melanoma-skin-cancer-dataset-of-10000-images>
- Benignos o malignos1:  
<https://www.kaggle.com/datasets/fanconic/skin-cancer-malignant-vs-benign?select=train>
- Benignos o malignos2:  
<https://www.kaggle.com/datasets/hasnainjaved/melanoma-skin-cancer-dataset-of-10000-images>
- Varios tipos de lesiones: <https://www.kaggle.com/datasets/yashjaiswal4559/skin-disease>
- Tipos de cancer: <https://www.kaggle.com/datasets/riyaelizashaju/isic-skin-disease-image-dataset-labelled>
- Twitts: <https://huggingface.co/datasets/pysentimiento/spanish-tweets>
- Web donde se hace web scrapping: <https://www.tutiempo.net/malaga.html?datos=detallados>



### 3. Limpieza de datos (Preprocesado).

#### Modelo de sentimientos:

- Tweets: eliminamos valores y caracteres que no son necesarios:
- iconos, @, # y el texto asociado.
- urls y convertir todo a minúsculas.
- Stop words

```
# Sin las stopwords
stop_words = set(stopwords.words('spanish'))
df['texto'] = df['texto'].apply(lambda x: ' '.join([word.lower() for word in x.split() if word.isalpha() and word.lower() not in stop_words]))
```



### 3. Limpieza de datos (Preprocesado).

El resto de modelos:

Como estamos trabajando con imágenes hemos:

- Modificado el nombre de todas las imágenes a un nombre más corto.
- Eliminar imágenes repetidas
- Mal etiquetadas (si están en las dos categorías).

```
# Renombrar y guardar las imágenes dentro de la subcarpeta
for i, nombre_archivo in enumerate(archivos_en_subcarpeta, start=1):
    # Ruta completa del archivo original y nuevo nombre
    ruta_original = os.path.join(subcarpeta, nombre_archivo)
    nuevo_nombre = f"o{i}.jpg" # Puedes ajustar la extensión según el
    ruta_nueva = os.path.join(subcarpeta, nuevo_nombre)

    # Renombrar el archivo
    os.rename(ruta_original, ruta_nueva)
```

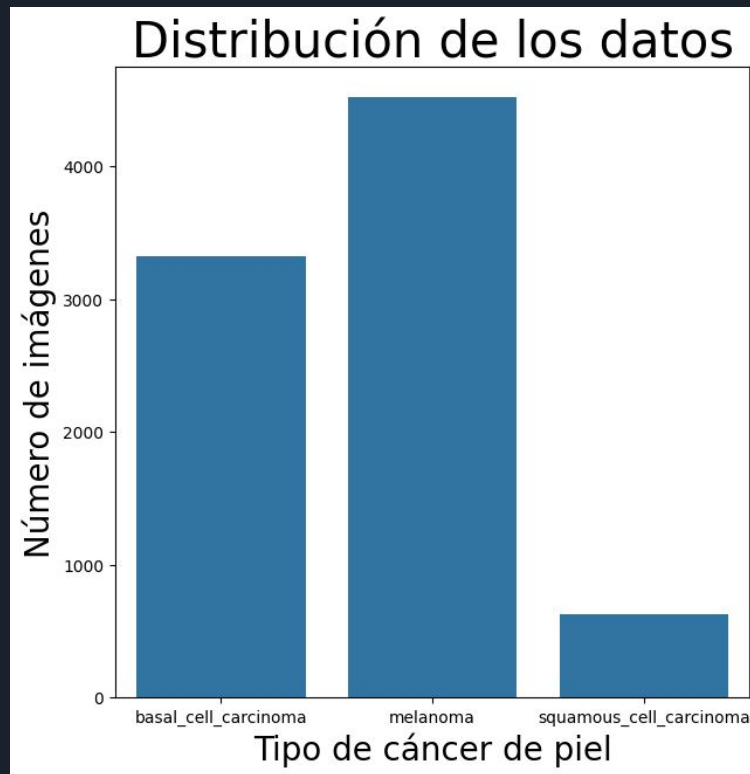
## 4. Exploración y visualización de los datos

Mostramos el contenido del dataset

Train data distribution :

	no. of images
basal_cell_carcinoma	3323
melanoma	4522
squamous_cell_carcinoma	628

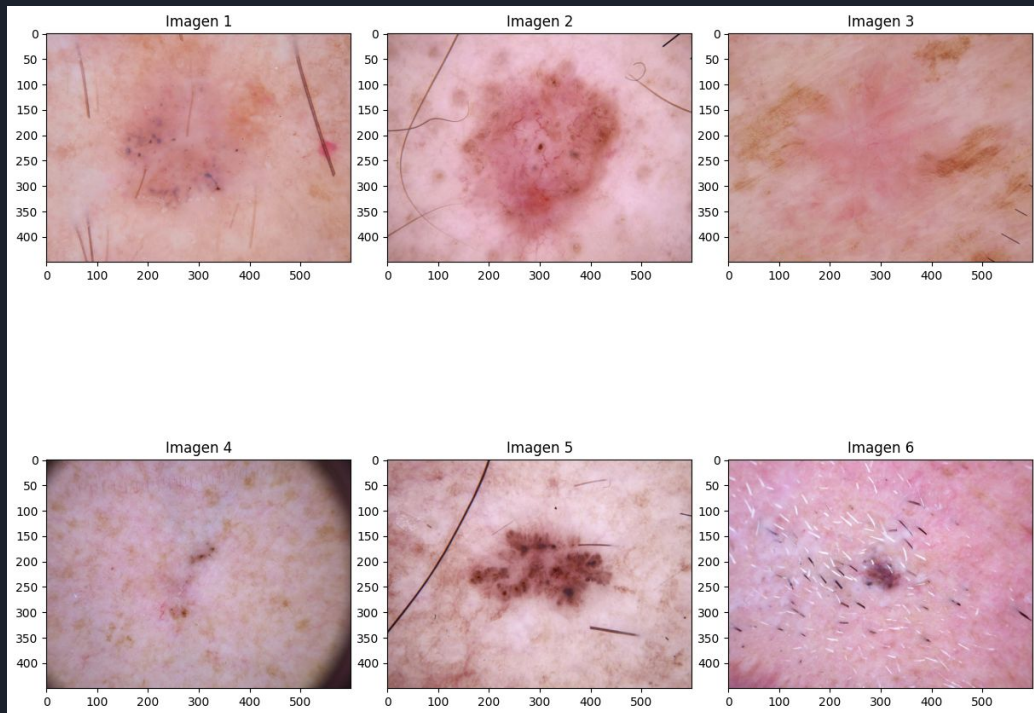
## 4. Exploración y visualización de los datos





# 4. Exploración y visualización de los datos

## Basall Cell Carcinoma





## 5. Preparación de los datos para ML

Redimensionamiento de las imágenes a 150 x 150 píxeles.

Concatenamos todos los datos en arrays para trabajar mejor con ellos y mejorar la velocidad de entrenamiento.

Dividimos los datos en train y test y sus etiquetas correspondientes.

También normalizamos los pixeles dividiendo entre 255 y convertimos la etiquetas a un formato one-hot.

**CustomLearningRateScheduler:** ajusta dinámicamente la tasa de aprendizaje durante el entrenamiento de un modelo.

Se monitorea el **val\_accuracy** con una 'patient' de 1 época.

Si el **val\_accuracy** disminuye se ejecuta la clase disminuyendo el learning rate dividiéndolo entre /2 eso significa que si antes teníamos 0.001 ahora tendríamos 0.0005.



## 5. Preparación de los datos para ML

Checkpoint : va guardando el modelo cada vez que mejora el **val\_accuracy** para que guarde el mejor valor de precisión que ha obtenido en alguna de las Epochs de entrenamiento.

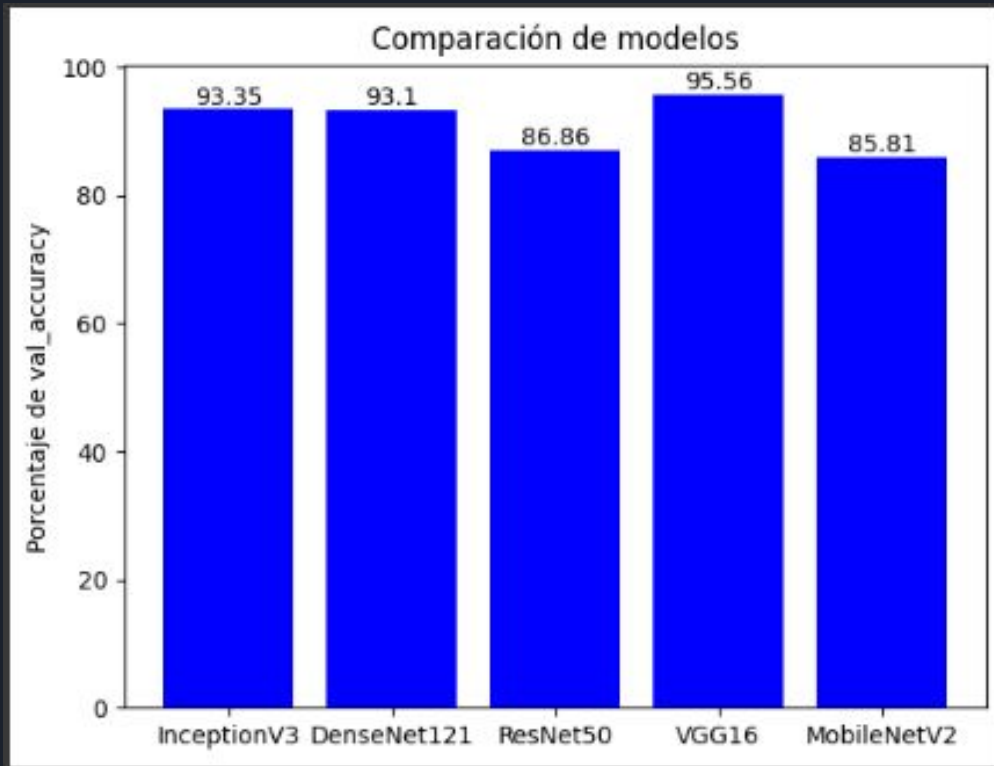
ImageDataGenerator : se encarga de realizar aumento de datos y preprocesamiento para conjuntos de entrenamiento y prueba en clasificación de imágenes.

Mejora la capacidad de generalización del modelo al exponerlo a variaciones en los datos durante el entrenamiento.

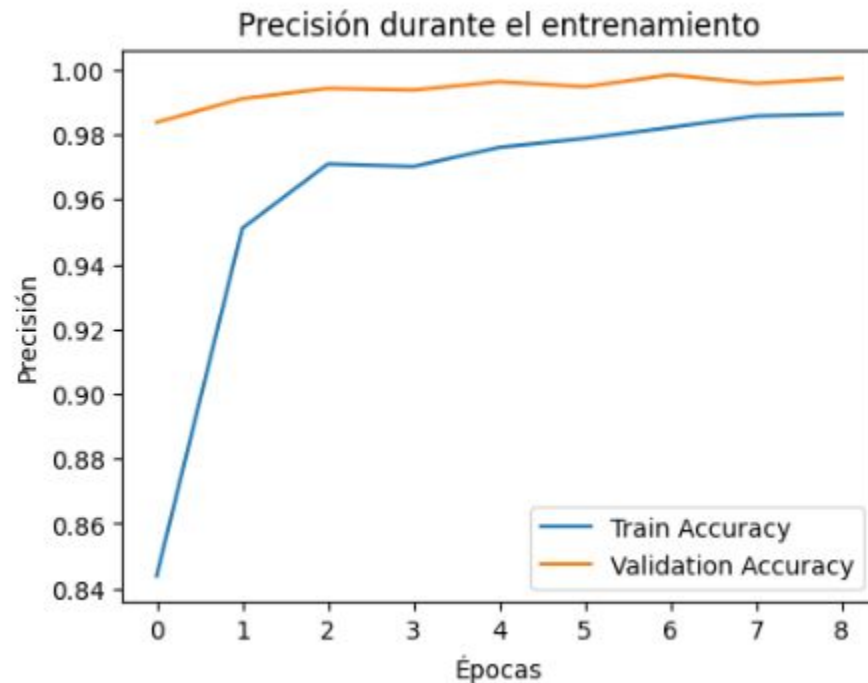
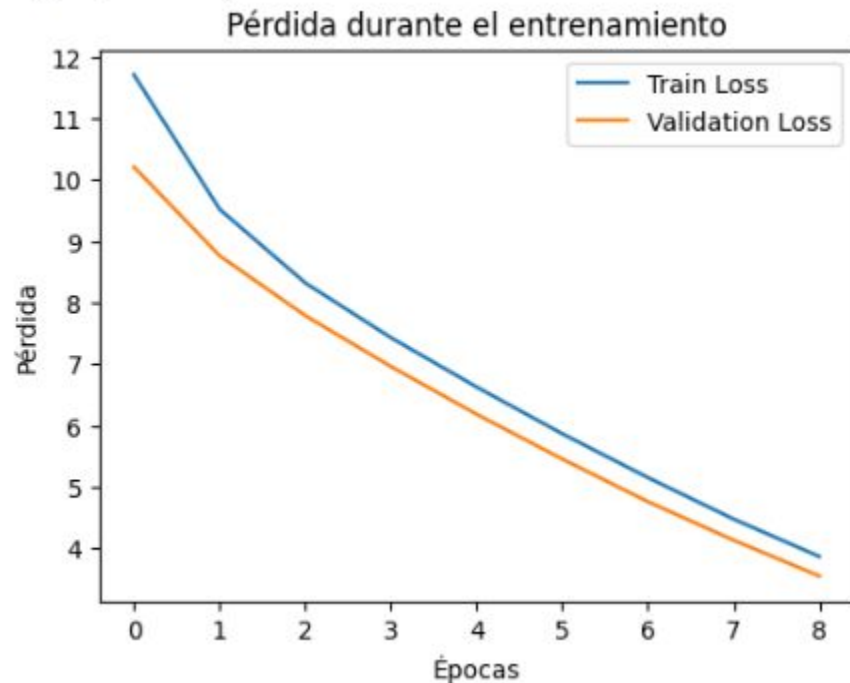
Ayuda a prevenir el sobreajuste al proporcionar más variabilidad en el conjunto de entrenamiento.

## 6 - Entrenamiento y Rendimiento

En algunos modelos hemos utilizado modelos pre entrenados como por ejemplo: VGG16, MobileNetV2 y Xception.



# 6 - Entrenamiento del modelo y comprobación del rendimiento



# Precisión de los modelos

Modelo	Porcentaje de validación
Objeto o piel	99%
Piel sana o piel con lesión	97%
Piel con lesión o cáncer	97%
Benigno o maligno	95%
Clasificación malignos	94%
Clasificación benignos	93%
Analizador de sentimientos	92%

Enlace al esquema: [Esquema](#)

# 7 - Procesamiento de Lenguaje Natural

## Valoraciones

Comenta que te ha parecido la App:

Enviar



¡La puntuación media es de 2.66!

En el ámbito del Procesamiento de Lenguaje Natural, hemos desarrollado un modelo diseñado para detectar sentimientos en texto. Este modelo se integra en nuestra plataforma web, específicamente en un sistema de reseñas con calificación estelar. Al introducir un texto en el modelo, obtendrás una puntuación en una escala del 0 al 5. Una puntuación de 0 indica una valoración muy negativa, 2.5 representa una valoración neutra, y 5 refleja una valoración muy positiva.

Para conseguir crear el modelo primero necesitamos datos de textos con su puntuación lo que hemos hecho es recopilar texto de twitter y crear una serie de funciones para analizar la frase para después añadirlo a un dataset. Estos son los pasos que hemos realizado para crear el modelo

## 8 - Aplicación web



Enlace a nuestra web: [WEB](#)



## 9 - Conclusiones

Ha sido un desafío personal para aplicar nuestros conocimientos de ML.

A pesar de los desafíos encontrados, como la limitación de tiempo y la baja disponibilidad de datos.

Nuestra aplicación web puede ser solo el comienzo de un viaje más largo hacia soluciones más avanzadas y efectivas en el diagnóstico y tratamiento de enfermedades de la piel.

Nos sentimos orgullosos de lo que hemos logrado hasta ahora y la importancia de seguir adelante, explorando nuevas ideas, colaborando con expertos en el campo y haciendo un impacto positivo en la salud y el bienestar de las personas.

DermaScan puede ser el punto de partida para futuras innovaciones que mejoren la vida de millones de personas en todo el mundo.



## 10 -Bibliografía

[Documentación Nvidia Cudnn](#)

[Documentación Tensorflow](#)

[Documentación clasificador de imagenes Tensorflow](#)

[Documentación Streamlit](#)

[Documentación HTML](#)

[Documentación CSS](#)

[Tutoriales Medium](#)

[Documentación Markdown](#)

[Documentación Python 3.9](#)



FIN