

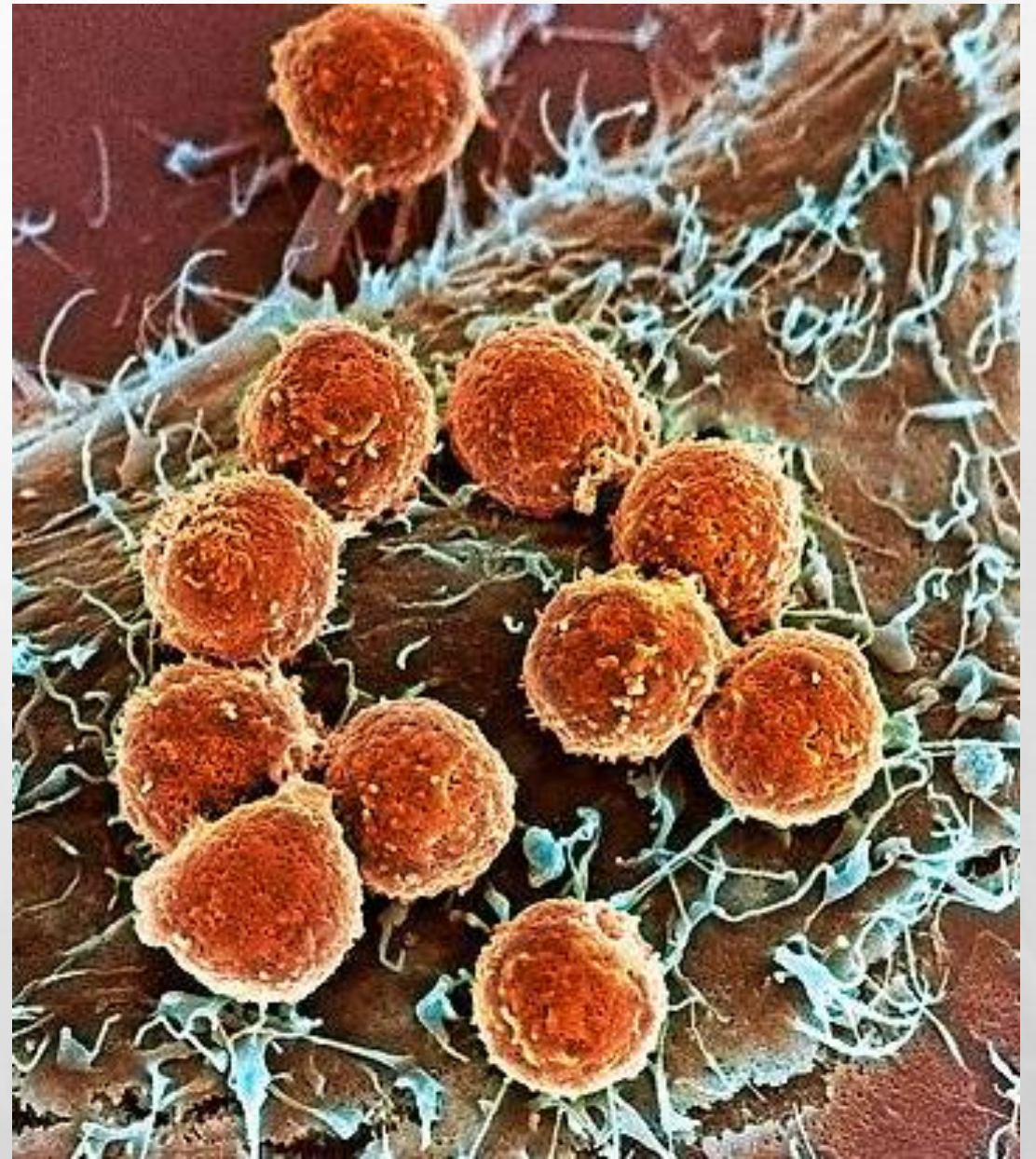
# **PREDICCIÓN DE MELANOMAS BENIGNOS Y MALIGNOS**

**POR JOSÉ EDUARDO LÓPEZ**



# ¿POR QUÉ ESTE PROYECTO?

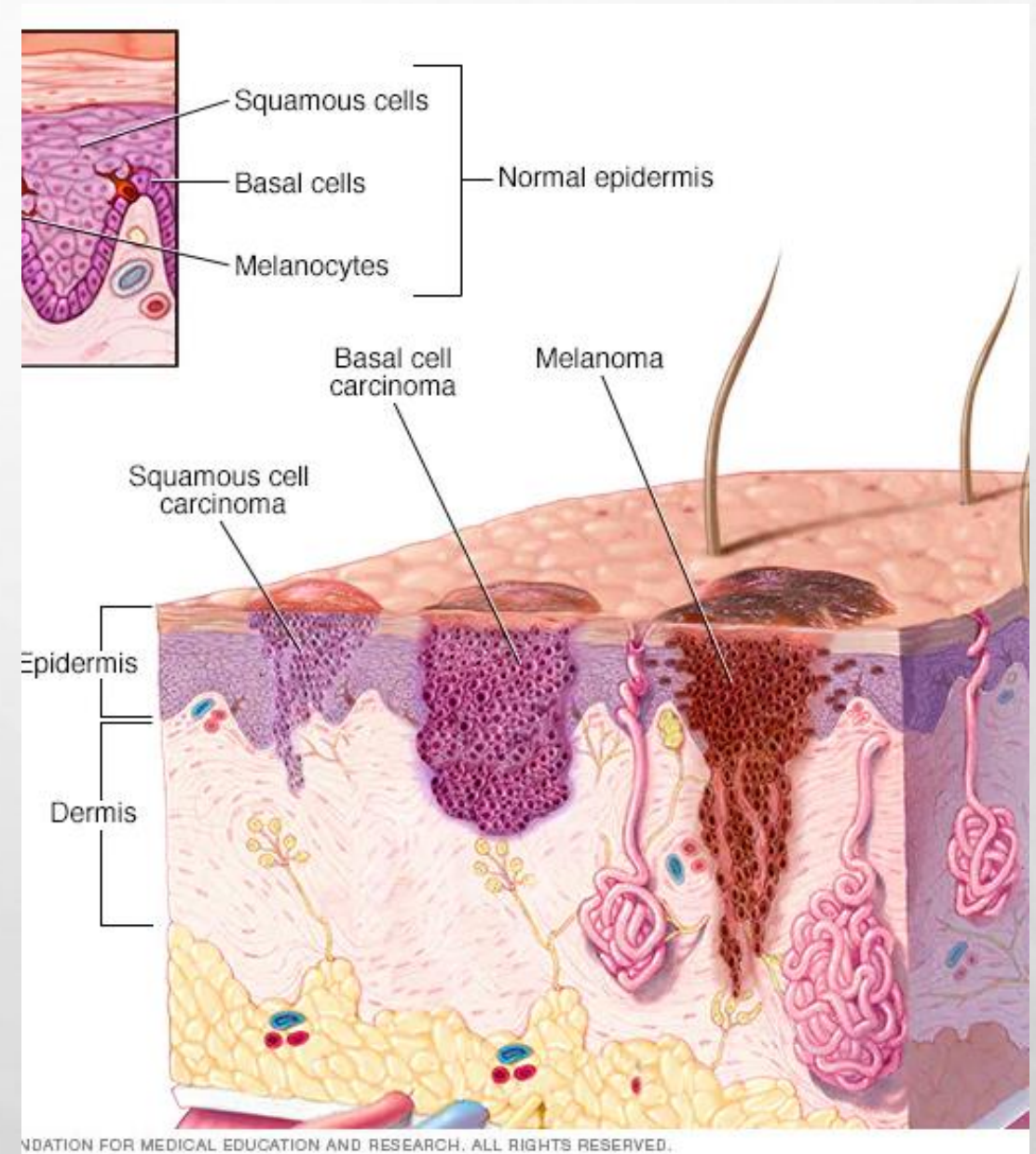
- EL CÁNCER ES UNA ENFERMEDAD INCURABLE, QUE AFECTA A UNA GRAN PARTE DE LA POBLACIÓN MUNDIAL. SOLO EN 2018 SE REGISTRARON 9.5 MILLONES DE MUERTES POR CÁNCER EN EL MUNDO.





# MELANOMA

- **LOS MELANOMAS SON EL TIPO MÁS GRAVE DE CÁNCER DE PIEL. OCURRE CUANDO EXISTE UN PROBLEMA CON LAS CÉLULAS PRODUCTORAS DE LA PROTEÍNA MELANINA (PRODUCE LA PIGMENTACIÓN DE LA PIEL).**
- **SE CREE QUE UNO DE LOS FACTORES CRUCIALES ES LA EXPOSICIÓN A LOS RAYOS UV.**

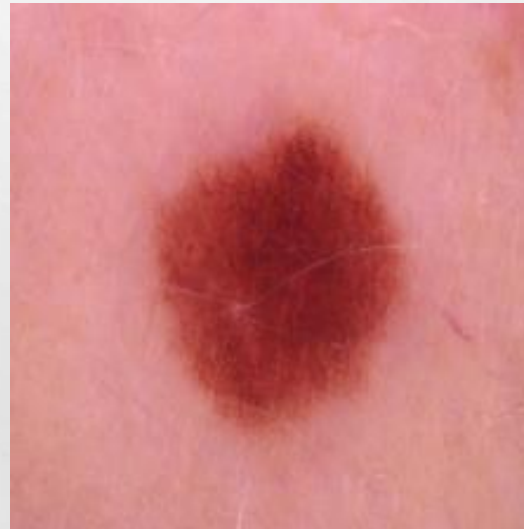


# BASE DE DATOS

**Dentro de la plataforma Kaggle se encontró una base de datos de 3,297 fotografías de melanomas (benignos y malignos). Las imágenes son de 224x224 con una escala de colores RGB.**

**La información ya venía particionada en datos de entrenamiento y datos de prueba (2637, para entrenamiento, 660 para prueba).**

**MELANOMA BENIGNO**

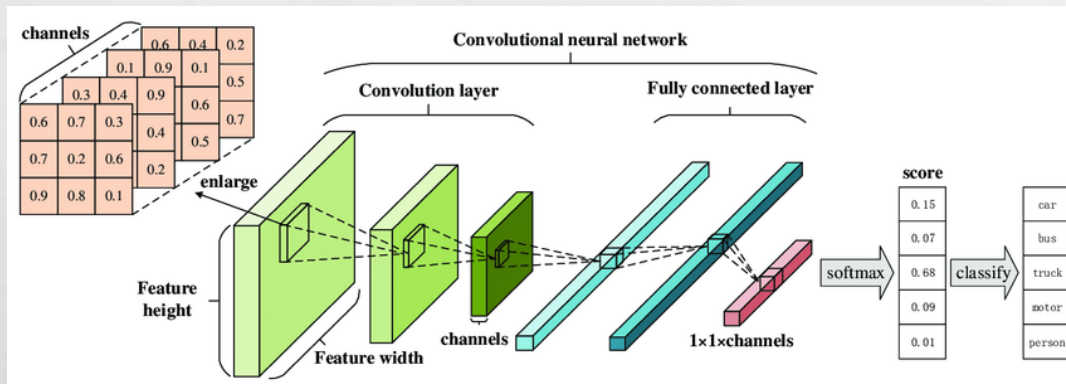


**MELANOMA MALIGNO**

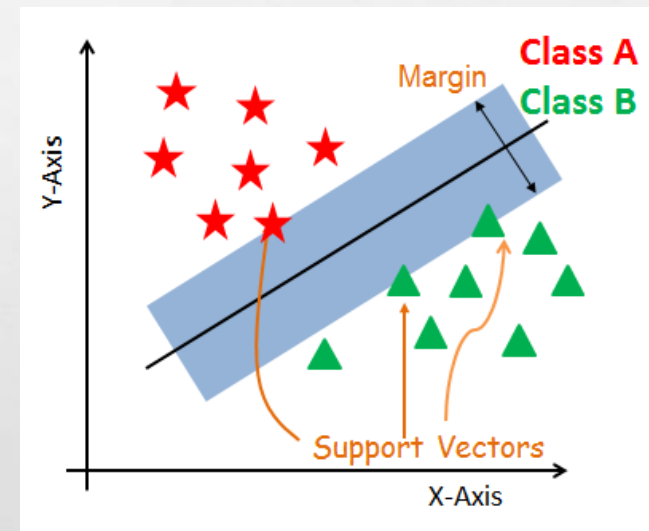


# ¿QUÉ SE TRABAJÓ?

## RED NEURONAL CONVOLUCIONAL (CNN)



## PREDICCIÓN USANDO SVM





# CNN

- SE TRABAJÓ UN RED NEURONAL CONVOLUCIONAL CON 3 BLOQUES DE CONVOLUCIÓN. LOS KERNEL FUERON DE TAMAÑO (3,3) Y LOS MAXPOOL DE (2,2).
- SE EMPLEÓ LA FUNCIÓN SIGMOIDE EN LA MAYORÍA DE CAPAS. EXCEPTO EN LA ÚLTIMA QUE SE UTILIZE UNA FUNCIÓN SOFTMAX
- SE HICIERON ALGUNAS MODIFICACIONES, PERO ESTE MODELO RESULT SER EL MAJOR.

Model: "CNN-model"

Layer (type)	Output Shape
=====	
input (InputLayer)	[(None, 224, 224, 3)]
conv1 (Conv2D)	(None, 222, 222, 32)
maxpool1 (MaxPooling2D)	(None, 111, 111, 32)
dropout1 (Dropout)	(None, 111, 111, 32)
conv2 (Conv2D)	(None, 109, 109, 64)
maxpool2 (MaxPooling2D)	(None, 54, 54, 64)
dropout2 (Dropout)	(None, 54, 54, 64)
conv3 (Conv2D)	(None, 52, 52, 64)
flatten (Flatten)	(None, 173056)
dense (Dense)	(None, 64)
classifier (Dense)	(None, 2)
=====	
Total params: 11,132,098	
Trainable params: 11,132,098	
Non-trainable params: 0	

# MODIFICACIONES

- **SE INTENTÓ REDUCIR EL TAMAÑO DE LA IMAGEN.**
- **AUMENTAR EL NÚMERO DE CAPAS DE CONVOLUCIÓN (PARA REDUCIR PARÁMETROS).**
- **SE INTENTÓ HACER UN EMBEDDING**

**Todos estos intentos por mejorar el desempeño de la red dieron resultados no satisfactorios. Al aplicar estos cambios la red se “estancaba” después de la primeras 3 o 4 épocas.**

**Con el modelo seleccionado se mantenía realizando oscilaciones en lugar de converger a un valor, pero al menos no se estancó en 32 épocas.**

**Las oscilaciones eran esperadas debido a que se está trabajando con imágenes.**

# ENTRENAMIENTO

## PRIMERAS 16 ÉPOCAS

```
Epoch 1/32
18/18 [=====] - 347s 15s/step - loss: 1.0238 - accuracy: 0.5053 - val_loss: 0.7353 - val_accuracy: 0.4727
Epoch 2/32
18/18 [=====] - 149s 8s/step - loss: 0.7253 - accuracy: 0.4718 - val_loss: 0.7062 - val_accuracy: 0.5273
Epoch 3/32
18/18 [=====] - 115s 6s/step - loss: 0.6972 - accuracy: 0.5304 - val_loss: 0.6953 - val_accuracy: 0.5273
Epoch 4/32
18/18 [=====] - 157s 8s/step - loss: 0.6870 - accuracy: 0.5627 - val_loss: 0.6920 - val_accuracy: 0.5273
Epoch 5/32
18/18 [=====] - 56s 3s/step - loss: 0.6896 - accuracy: 0.5489 - val_loss: 0.6936 - val_accuracy: 0.5273
Epoch 6/32
18/18 [=====] - 58s 3s/step - loss: 0.6897 - accuracy: 0.5485 - val_loss: 0.6924 - val_accuracy: 0.5273
Epoch 7/32
18/18 [=====] - 56s 3s/step - loss: 0.6869 - accuracy: 0.5633 - val_loss: 0.6917 - val_accuracy: 0.5273
Epoch 8/32
18/18 [=====] - 107s 6s/step - loss: 0.6885 - accuracy: 0.5547 - val_loss: 0.6940 - val_accuracy: 0.5273
Epoch 9/32
18/18 [=====] - 110s 6s/step - loss: 0.6908 - accuracy: 0.5445 - val_loss: 0.6918 - val_accuracy: 0.5273
Epoch 10/32
18/18 [=====] - 89s 5s/step - loss: 0.6920 - accuracy: 0.5153 - val_loss: 0.6942 - val_accuracy: 0.5273
Epoch 11/32
18/18 [=====] - 107s 5s/step - loss: 0.6922 - accuracy: 0.5371 - val_loss: 0.6954 - val_accuracy: 0.5273
Epoch 12/32
18/18 [=====] - 56s 3s/step - loss: 0.6888 - accuracy: 0.5432 - val_loss: 0.7035 - val_accuracy: 0.5273
Epoch 13/32
18/18 [=====] - 56s 3s/step - loss: 0.6940 - accuracy: 0.5512 - val_loss: 0.6923 - val_accuracy: 0.5273
Epoch 14/32
18/18 [=====] - 59s 3s/step - loss: 0.6900 - accuracy: 0.5443 - val_loss: 0.6953 - val_accuracy: 0.5273
Epoch 15/32
18/18 [=====] - 56s 3s/step - loss: 0.6918 - accuracy: 0.5421 - val_loss: 0.6918 - val_accuracy: 0.5273
Epoch 16/32
18/18 [=====] - 57s 3s/step - loss: 0.6901 - accuracy: 0.5524 - val_loss: 0.6917 - val_accuracy: 0.5273
```

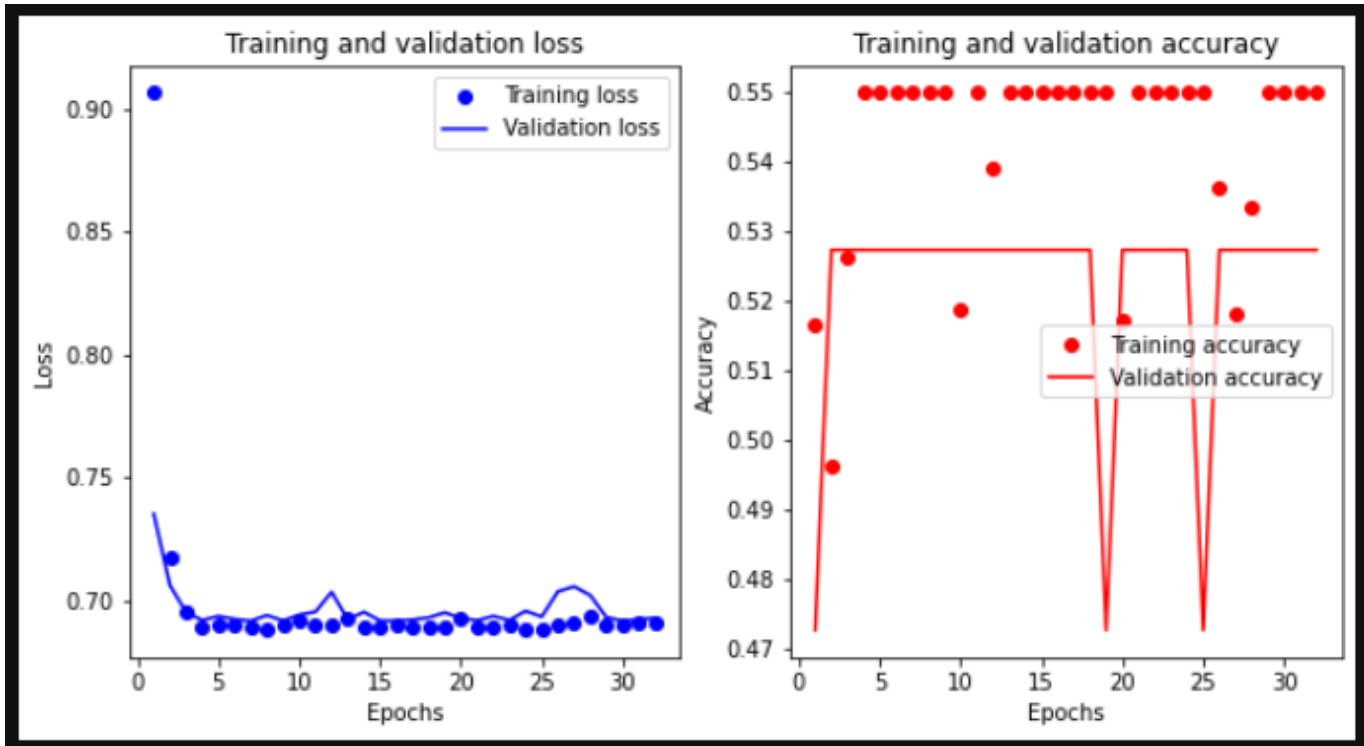
## ÚLTIMAS 16 ÉPOCAS

```
Epoch 17/32
18/18 [=====] - 56s 3s/step - loss: 0.6887 - accuracy: 0.5571 - val_loss: 0.6922 - val_accuracy: 0.5273
Epoch 18/32
18/18 [=====] - 56s 3s/step - loss: 0.6882 - accuracy: 0.5551 - val_loss: 0.6931 - val_accuracy: 0.5273
Epoch 19/32
18/18 [=====] - 56s 3s/step - loss: 0.6867 - accuracy: 0.5611 - val_loss: 0.6951 - val_accuracy: 0.4727
Epoch 20/32
18/18 [=====] - 56s 3s/step - loss: 0.6959 - accuracy: 0.4872 - val_loss: 0.6928 - val_accuracy: 0.5273
Epoch 21/32
18/18 [=====] - 58s 3s/step - loss: 0.6871 - accuracy: 0.5592 - val_loss: 0.6919 - val_accuracy: 0.5273
Epoch 22/32
18/18 [=====] - 56s 3s/step - loss: 0.6896 - accuracy: 0.5441 - val_loss: 0.6937 - val_accuracy: 0.5273
Epoch 23/32
18/18 [=====] - 57s 3s/step - loss: 0.6881 - accuracy: 0.5600 - val_loss: 0.6920 - val_accuracy: 0.5273
Epoch 24/32
18/18 [=====] - 55s 3s/step - loss: 0.6902 - accuracy: 0.5279 - val_loss: 0.6958 - val_accuracy: 0.5273
Epoch 25/32
18/18 [=====] - 57s 3s/step - loss: 0.6840 - accuracy: 0.5685 - val_loss: 0.6934 - val_accuracy: 0.4727
Epoch 26/32
18/18 [=====] - 56s 3s/step - loss: 0.6916 - accuracy: 0.5055 - val_loss: 0.7037 - val_accuracy: 0.5273
Epoch 27/32
18/18 [=====] - 55s 3s/step - loss: 0.6924 - accuracy: 0.5155 - val_loss: 0.7057 - val_accuracy: 0.5273
Epoch 28/32
18/18 [=====] - 54s 3s/step - loss: 0.6951 - accuracy: 0.5365 - val_loss: 0.7021 - val_accuracy: 0.5273
Epoch 29/32
18/18 [=====] - 54s 3s/step - loss: 0.6909 - accuracy: 0.5504 - val_loss: 0.6932 - val_accuracy: 0.5273
Epoch 30/32
18/18 [=====] - 54s 3s/step - loss: 0.6879 - accuracy: 0.5561 - val_loss: 0.6917 - val_accuracy: 0.5273
Epoch 31/32
18/18 [=====] - 55s 3s/step - loss: 0.6905 - accuracy: 0.5502 - val_loss: 0.6923 - val_accuracy: 0.5273
Epoch 32/32
18/18 [=====] - 54s 3s/step - loss: 0.6904 - accuracy: 0.5530 - val_loss: 0.6929 - val_accuracy: 0.5273
```



# UN POCO MÁS VISUAL

- **PODEMOS VER QUE LA PÉRDIDA NO CONVERGE Y SE MANTIENE OSCILANDO, AL IGUAL QUE EL ACCURACY.**



# RESULTADOS DE PRUEBA

PODEMOS NOTAR QUE EL ACCURACY ES UN POCO MAYOR A 50%. LO CUÁL PODRÍA CONSIDERARSE “DECENTE”.

	precision	recall	f1-score	support
0	0.55	1.00	0.71	360
1	0.00	0.00	0.00	300
accuracy			0.55	660
macro avg	0.27	0.50	0.35	660
weighted avg	0.30	0.55	0.39	660

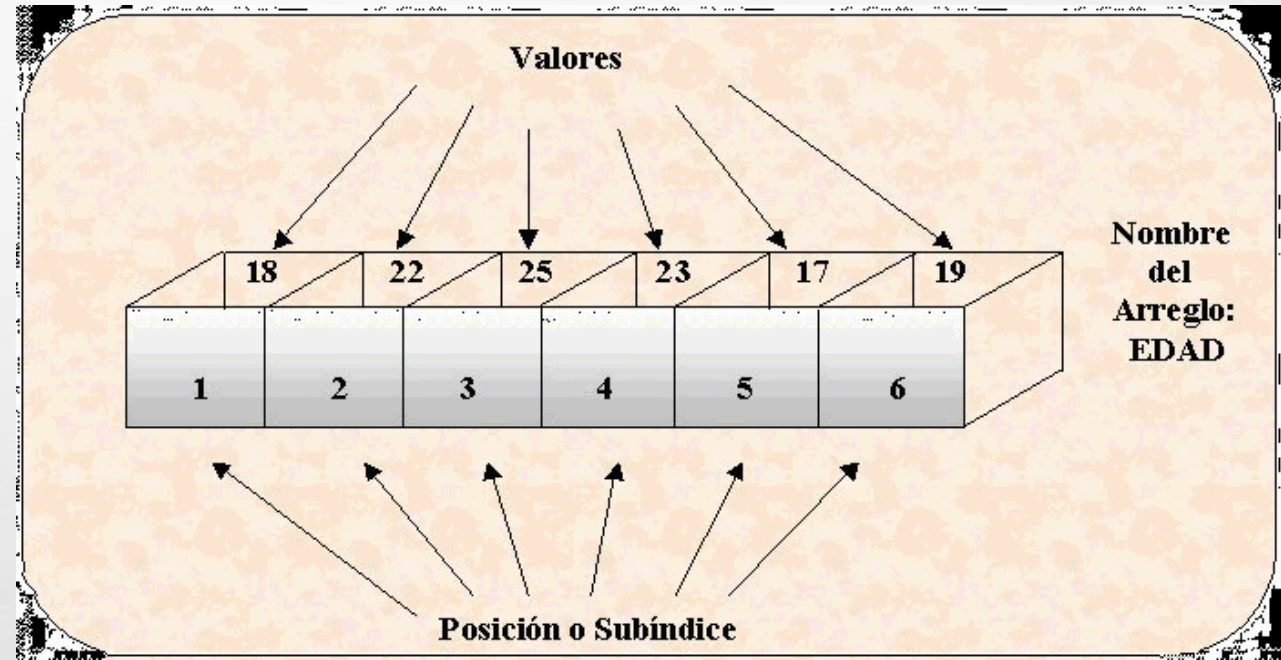
# MÁQUINA DE VECTORES DE SOPORTE





## ¿CÓMO SE TRABAJÓ?

**PRIMERO SE VECTORIZARON LAS IMÁGENES Y LUEGO CON LA AYUDA DE LA LIBRERÍA SCI-KIT LEARN SE IMPLEMENTÓ UN MODELO DE SVM.**



# RESULTADOS DE DISTINTOS SVM

## SVM

EL TIEMPO DE EJECUCIÓN DEL ENTRENAMIENTO DE ESTE MODELO FUE TARDADO PERO SE LOGRARON LOS SIGUIENTES RESULTADOS

- **ACCURACY SCORE: 83.33%**
- **F1 SCORE: 82.32%**
- **JACCARD SCORE: 69.95%**
- **AVG PRECISION SCORE: 74.51%**

## LSVC

SE EJECUTÓ CASI EN EL MISMO TIEMPO PERO LOS RESULTADOS FUERON MENOS SATISFACTORIOS:

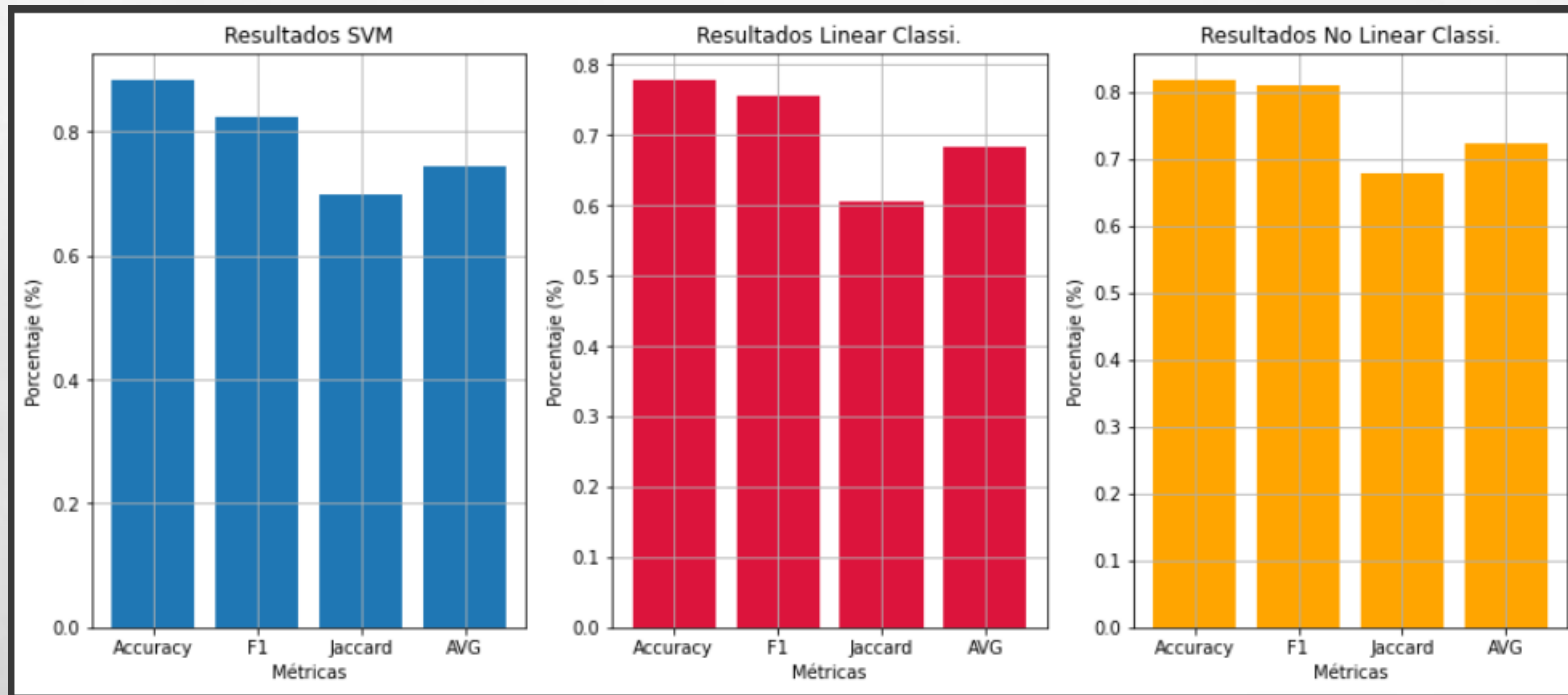
- **ACCURACY SCORE: 77.73%**
- **F1 SCORE: 75.46%**
- **JACCARD SCORE: 60.59%**
- **AVG PRECISION SCORE: 68.15%**

## NSVC

FUE EL QUE MÁS SE TARDÓ DE LOS 3 Y SUS RESULTADOS FUERON SIMILARES AL SVM

- **ACCURACY SCORE: 81.67%**
- **F1 SCORE: 80.88%**
- **JACCARD SCORE: 67.90%**
- **AVG PRECISION SCORE: 72.27%**

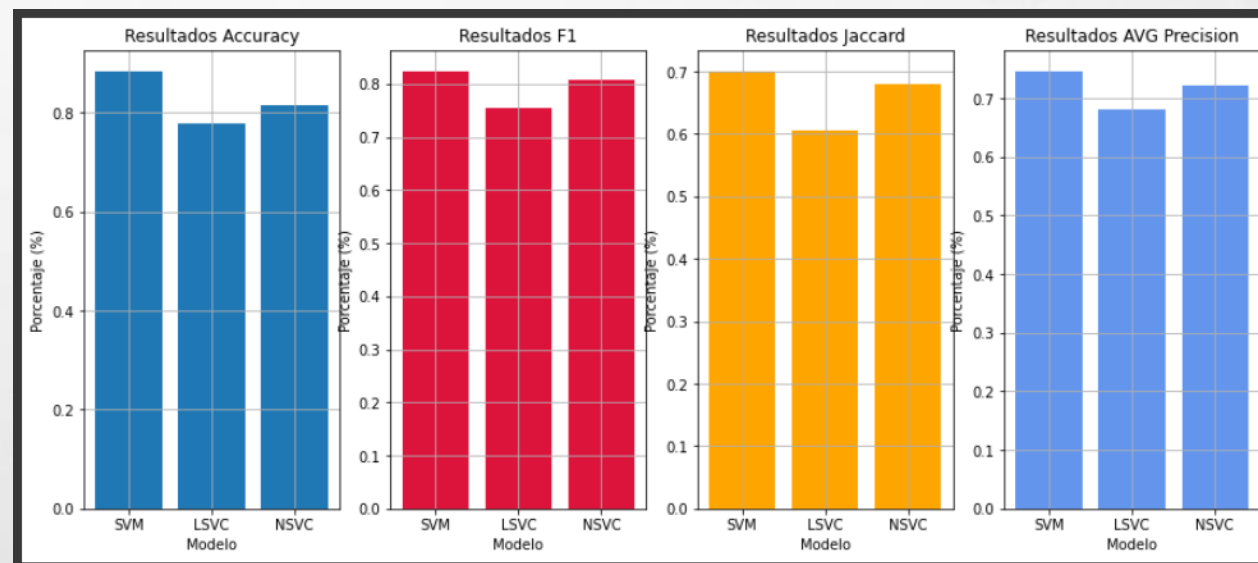
# COMPARACIÓN MÁS VISUAL





# RESULTADOS

**PODEMOS VER QUE EL MODELO QUE MEJOR  
ESTÁ PREDICIENDO LOS DATOS ES EL DE  
MÁQUINAS DE VECTORES DE SOPORTE,  
SEGUIDO DEL CLASIFICADOR NO LINEAL DE  
VECTORES DE SOPORTE Y POR ÚLTIMO  
TENEMOS EL CLASIFICADOR LINEAL.**



# CONCLUSIONES GENERALES

- **A pesar de que la red neuronal hace un buen trabajo podemos ver que los modelos de maquinas de vectores de soporte dan mejores resultados (en todos los casos).**
- **El mejor modelo fue el de máquinas de vectores de soporte, dado que tiene las métricas más altas de todos los modelos utilizados.**
- **Todos los algoritmos tienen un tiempo de ejecución bastante alto.**