**Use Case-2**

**Skin cancer Detection**

**Problem statement:**

Skin cancer is the most diagnosed cancer all over the world, It greatly affects one's daily life. Skin cancer also causes body dysmorphia.

The objective of the skin cancer detection project is to develop a use case to analyze and assess whether the cancer is benign or malignant so that individuals can take action immediately.

**Usage domains and advantages:**

The medical industry can use this model for identifying potential patients' cancer cells stage as benign or malignant.

Individuals can check their lesion condition using this model.

By understanding which stages their lesion is one can take precautions to avoid the risk.

Skin cancer detection using machine learning is known as intelligent computational predictive systems. These are proven to be very effective in many medical industries.

**Model solution:**

The problem is about detecting a person's skin lesion stage whether it is benign or malignant.

To identify these lesion stages using images of these stages a Sequential model is considered to be more suitable.

**Dataset/Data Source:**

Dataset is from Kaggle healthcare and medical datasets

<https://www.kaggle.com/fanconic/skin-cancer-malignant-vs-benign>

Dataset is divided into test and train sets which are further divided into benign and malignant

**Preprocessing:**

Images of benign and malignant are converted to the shape of 224,224 width and height

and anti-aliasing is used.

Anti-aliasing- When an image is resized from its original state it gets aliased in other words it gets shrunk, to resolve this anti-aliasing is used to create a true image of the original image with resized shapes

Label creation:

0 represents the image of benign variant skin cancer

1 represents the images of malignant variant skin cancer

There are divided to train and test NumPy arrays

X\_train - Benign and Malignant images

Y\_train - labels

X\_test - Benign and Malignant images

Y\_test - labels

Image count of benign in training and testing datasets  
Train dataset:

Benign - 1426 | Malignant - 1380

Test dataset:

Benign - 116 | Malignant - 188

**Model Used:**

Keras - Sequential model

Model additional layers :

Convolution layer

modelF.add(Conv2D(64,(3,3),padding="same",activation="relu",kernel\_initializer="glorot\_uniform",input\_shape=(224,224,3)))

Convolution layer with a kernel size of (3,3), kernel initializer ‘glorot\_uniform’

Input size of(224,224,3) width,height,channel of the images.

The activation function used is “RELU” rectified linear unit to achieve non-linear transformation.

RELU is better than other activation functions such as logistic,tanh since it doesn’t suffer vanishing gradient

Max pooling layer

modelF.add(MaxPool2D((2,2)))

Maxpool2D layer with the size of (2,2)

Max pooling is an operation that selects the max element from the region covered by the filter.

Batch Normalization layer

This layer is used for standardizing the input to a layer for each mini-batch, By stabilizing inputs it helps in increase of learning process of the model

Dense output layer

modelF.add(Dense(2, activation='softmax'))

A fully connected layer with output activation function ‘softmax’

Output activation function ‘SOFTMAX’ makes the sum of outputs be 1 so that it will be good to interpret probabilities of different classes

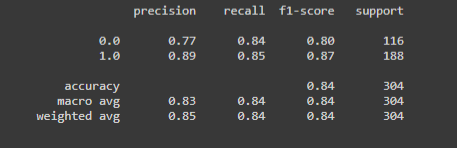
modelF.compile(optimizer = "adam" , loss = "categorical\_crossentropy", metrics=["accuracy"])

Model optimizer is ‘adam’ with metrics ‘accuracy

Model is executed with epochs 100 and batch size of 32

**Results:**

**Below are the results yielded by the model.**

****