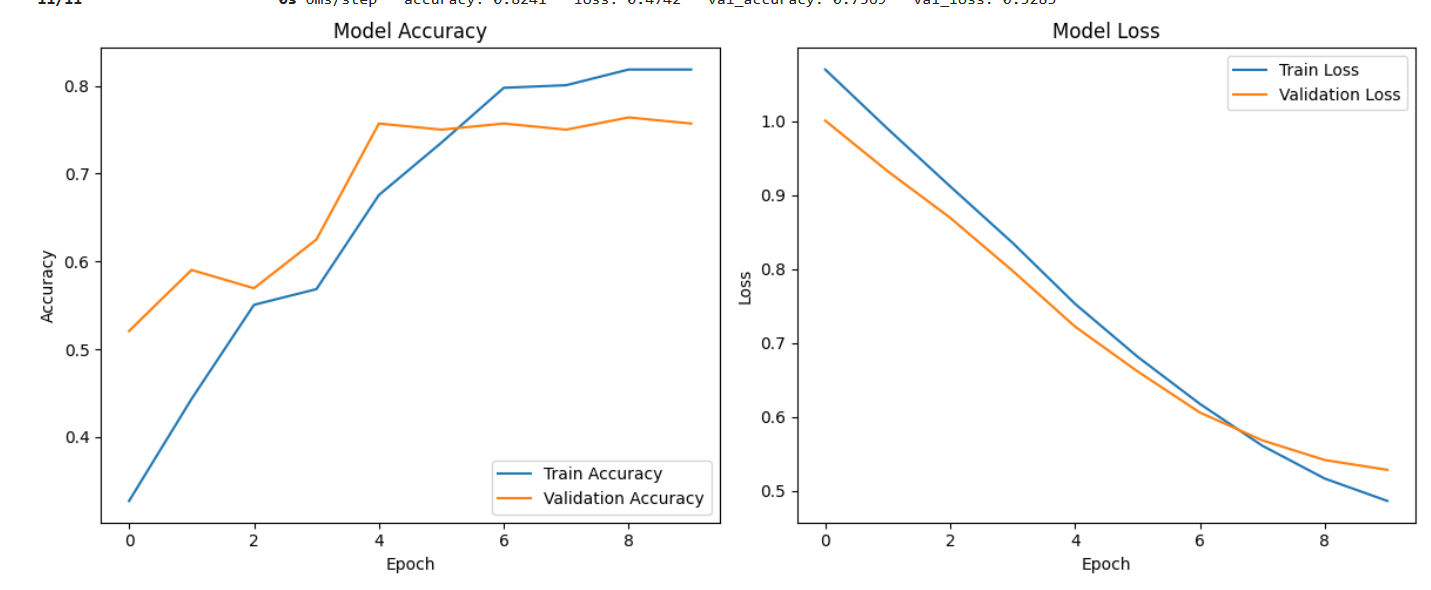
Report

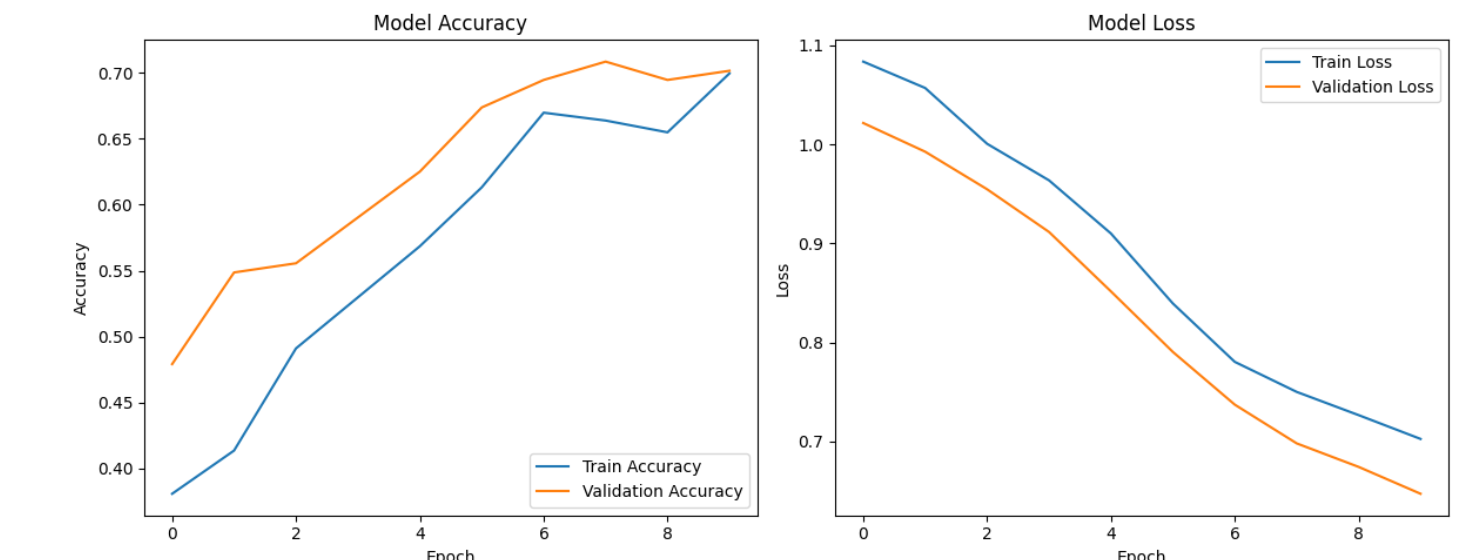
## Summarize

First, I analyze the dataset then check if there are null values and perform encoding for the categorical features since we can only pass a vector to the model. Then I separate the dataset into two parts (x, y). X represents the column without the results and Y represents the classes. Then I split X and Y as X\_train, X\_test, y\_train, and y\_test. (training set=70%) Then I scale those 4 parts using min-max scaling. Then I created the model and compiled it. Then run the training and testing. After that, I visualize the loss, accuracy, precision, recall, and F1-Score. After observing the initial results, I modified the model by adding a Dropout layer to improve its performance. Following these modifications, I achieved better loss and accuracy graphs compared to the previous result.

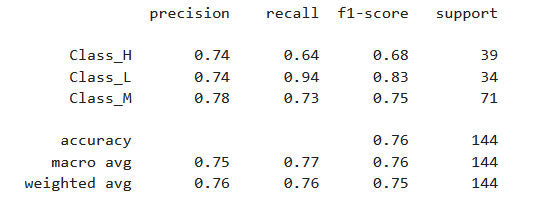
Previous result



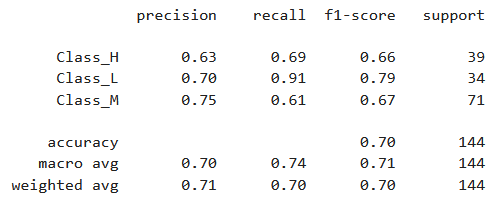
After the modification result



Previous result



After the modification result



# Challenges faced

I was unable to get the perfect diagram and a good accuracy at the same time.

Overfitting happens when I run the model fitting several times.

# Lessons learned

After a one model fitting restart the kernel.

If y\_train is one-hot encoded (meaning it has 3 columns for 3 classes), you need to use categorical\_crossentropy

Apply early stopping to prevent overfitting