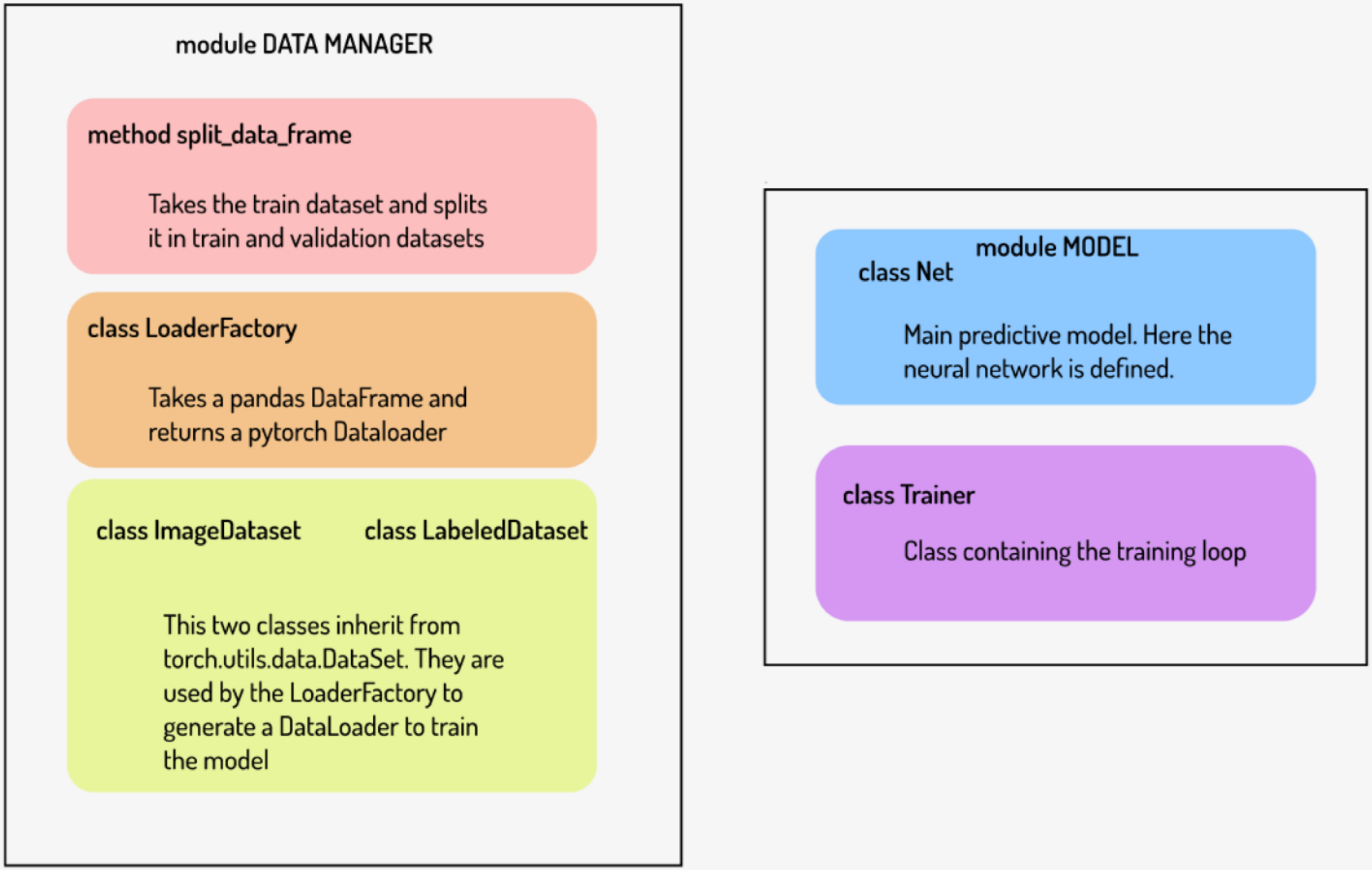
PROJECT STRUCTURE

To have the code organized, I decide to make a basic class structure. More or less, every class is responsible of one task



module DATA MANAGER

method split_data_frame

Takes the train dataset and splits it in train and validation datasets

class LoaderFactory

Takes a pandas DataFrame and returns a pytorch Dataloader

class ImageDataset

class LabeledDataset

This two classes inherit from torch.utils.data.DataSet. They are used by the LoaderFactory to generate a DataLoader to train the model

module MODEL

class Net

Main predictive model. Here the neural network is defined.

class Trainer

Class containing the training loop

MAIN WORKING LOOP

Then, in the main module of the project, I first trained the model on a split version of the train dataset.
This was to allow me to calculate the F1 score against a part of that dataset.
I kept tuning the hyperparameters and, when I was happy with the results, I retrained the model with the whole dataset. I did the predictions with that one

method split_data_frame

Takes the train dataset and splits it in train and validation datasets

class LoaderFactory

Takes a pandas DataFrame and returns a pytorch Dataloader

class ImageDataset **class LabeledDataset**

This two classes inherit from torch.utils.data.DataSet. They are used by the LoaderFactory to generate a DataLoader to train the model

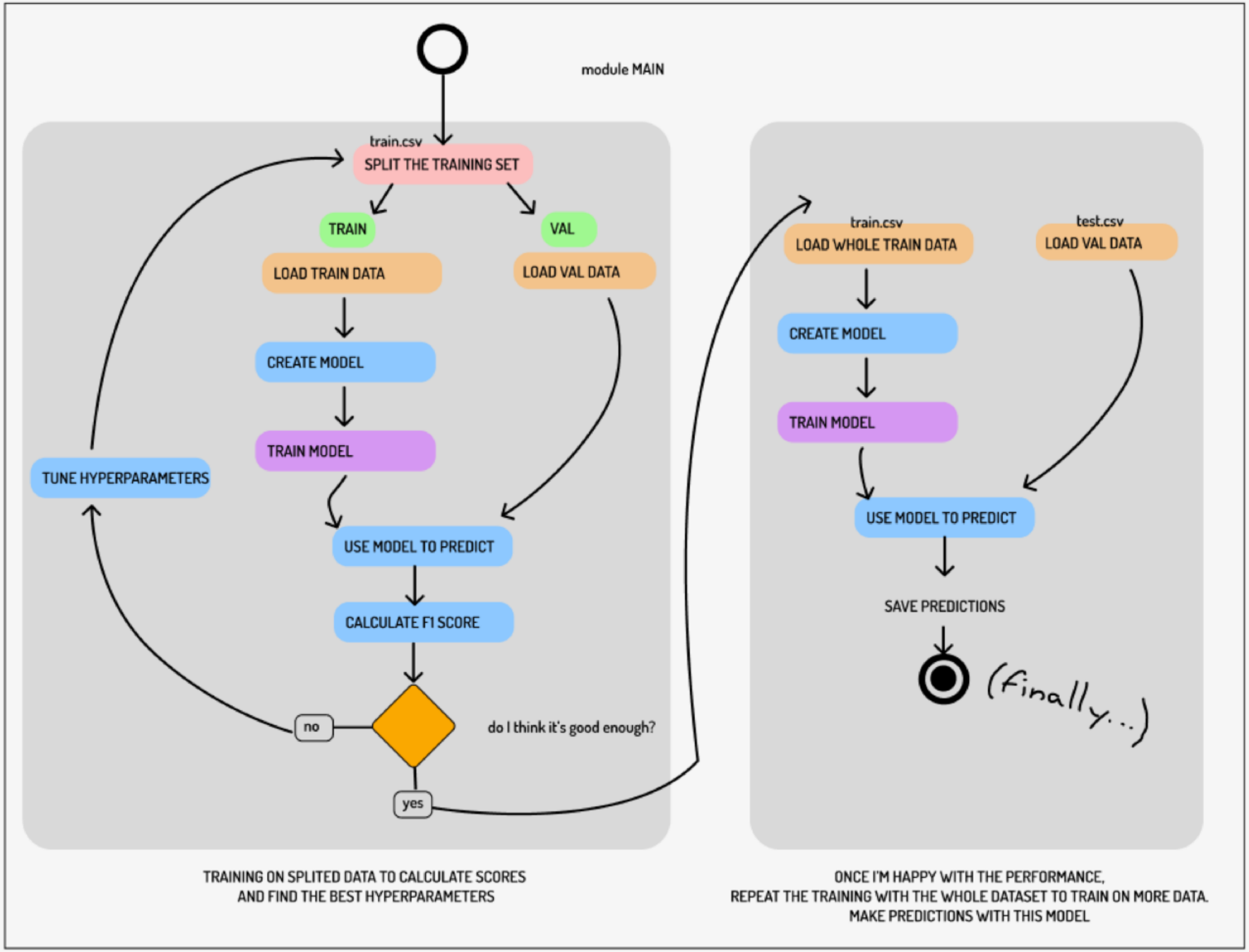
module MODEL

class Net

Main predictive model. Here the neural network is defined.

class Trainer

Class containing the training loop



module MAIN

train.csv

SPLIT THE TRAINING SET

TRAIN

VAL

LOAD TRAIN DATA

LOAD VAL DATA

CREATE MODEL

TRAIN MODEL

USE MODEL TO PREDICT

CALCULATE F1 SCORE

do I think it's good enough?

no

yes

TUNE HYPERPARAMETERS

train.csv

LOAD WHOLE TRAIN DATA

test.csv

LOAD VAL DATA

CREATE MODEL

TRAIN MODEL

USE MODEL TO PREDICT

SAVE PREDICTIONS

(Finally...)

TRAINING ON SPLITED DATA TO CALCULATE SCORES
AND FIND THE BEST HYPERPARAMETERS

ONCE I'M HAPPY WITH THE PERFORMANCE,
REPEAT THE TRAINING WITH THE WHOLE DATASET TO TRAIN ON MORE DATA.
MAKE PREDICTIONS WITH THIS MODEL

Finally, I got a score of around 0.95. I think that it’s not very good, that it may have overfitting.

I would have like to play around a bit with the network architecture, adding layers, dropping neurons… but I didn’t have time.