Labels Summary

We have run a variety of models with different arquitectures and ideas, as well as different sets of prediction labels. Before presenting the results, we show a brief summary of the labels:

- **PTB-XL :** The labels correspond to the superclases in the image to the right, but with the following modifications:

  MI : MI without STTC
  STTC : STTC without MI
  MI&STTC : MI and STTC
  OTHER : All of the rest

Superclass	Description
NORM	Normal ECG
MI	Myocardial Infarction
STTC	ST/T Change
CD	Conduction Disturbance
HYP	Hypertrophy

  The reason for doing this is that we are in general interested in detecting STEMI and NSTEMI, and separating MI from MI&STTC could give us some light into this (however, we can't tell for certain that MI&STTC = STEMI or MI = NSTEMI, since STTC could imply **any** change, not just an elevation).

- **MI or not:** If the registers contain MI, then it's 1. If not, it's 0.
- **Urgency :** Labeling according to the internal classification of 4 urgency levels.
- **Snomed :** Labeling equals the SNOMED CT diagnoses codes.

About the Results

In the next slide we present the reuslts for **PTB-XL** and **MI or not**. For each of them, we took a **Regular** and a **Threshold** approach.

- **Regular :** This one simply takes the maximum probability for each predicted vector. For example, for the predicted vector (note it does not have to sum up to 1):

  [MI, STTC, MI&STTC, OTHER] = [0.3, 0.02, 0.25, 0.8]

  we would assign class **OTHER**.

- **Threshold :** We set a threshold to detect MI more frequently than any other other class. We chose 20%, but it can be optimized according to specific needs. So, with the same vector, since 0.3 > 20%, we would choose the label **MI**.

Since PTB-XL has 4 classes, we calculate the Sensitivity and Specificity for each one, along with weighted versions of these in the Total rows.

Initial Results

- **PTB-XL :**

		Sensitivity	Specificity	Accuracy
Regular	MI	70.96 %	91.25 %	
	STTC	62.5 %	94.25 %	
	MI&STTC	8.3 %	99.51 %	
	OTHER	91.77 %	76.09 %	
	Total	**76.83 %**	**84.28 %**	**76.83 %**
MI Threshold = 20%	MI	78.49 %	86.88 %	
	STTC	59.09 %	94.25 %	
	MI&STTC	0 %	100 %	
	OTHER	88.74 %	80 %	
	Total	**75.68 %**	**85.44 %**	**75.68 %**

- **MI or not :**

Regular	**67.52 %**	**95.61 %**	**88.07 %**
MI Threshold = 20%	**82.05 %**	**89.65 %**	**87.61 %**

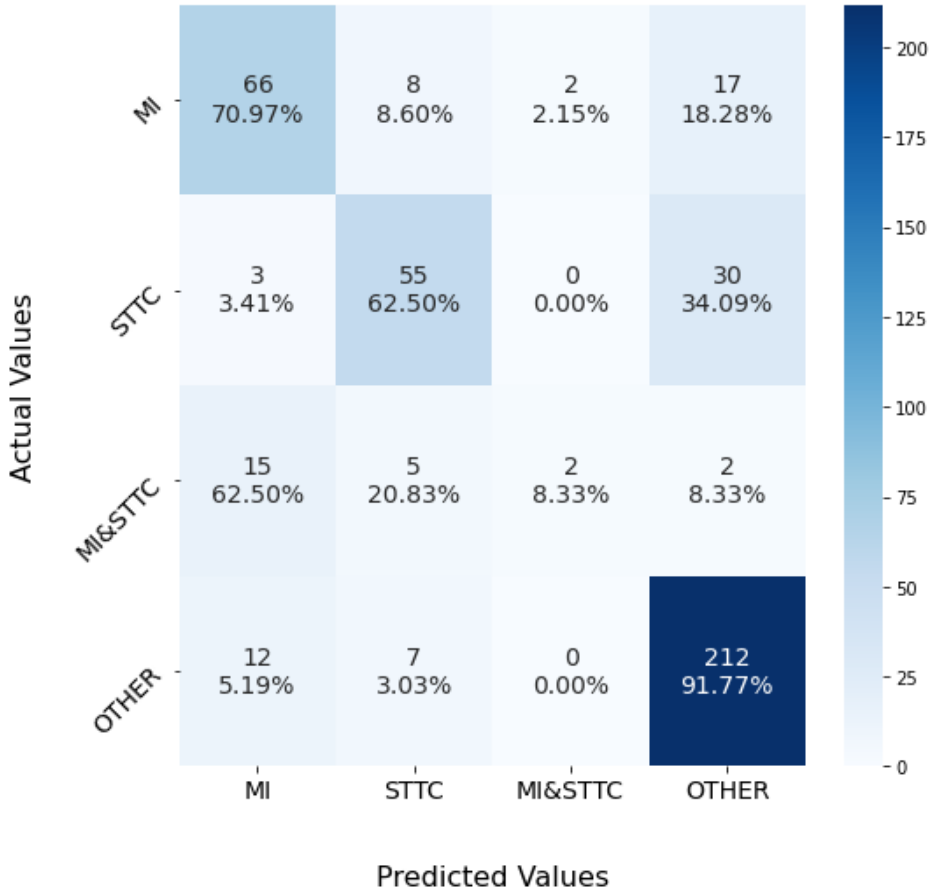
Confusion Matrix

- **PTB-XL :**

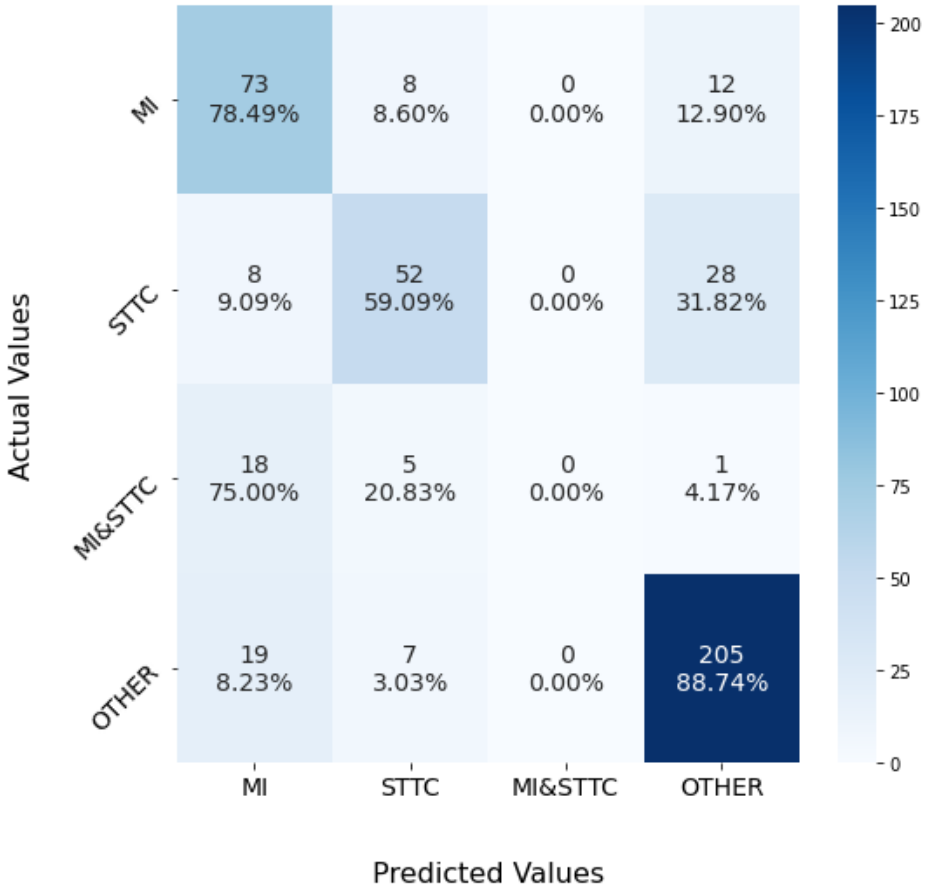
The model seems to be overfitting to OTHER. This is because that's the class with the most registers. It also seems to recover MI&STTC very poorly.

However, MI in both cases has a relatively adequate sensitivity. It can also recover a great amount of MI&STTC records (62.50% and 75%, respectively), which is desired to quickly treat the patient.

Regular



MI Threshold = 20%



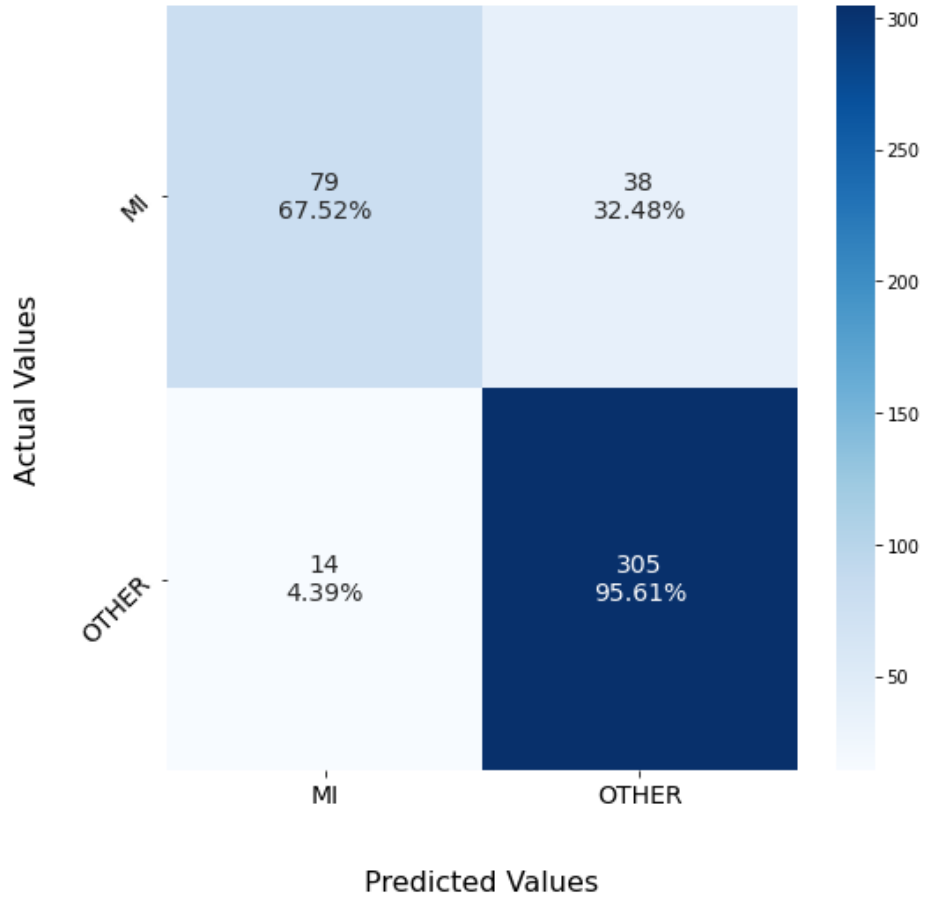
Confusion Matrix

- **MI or not :**

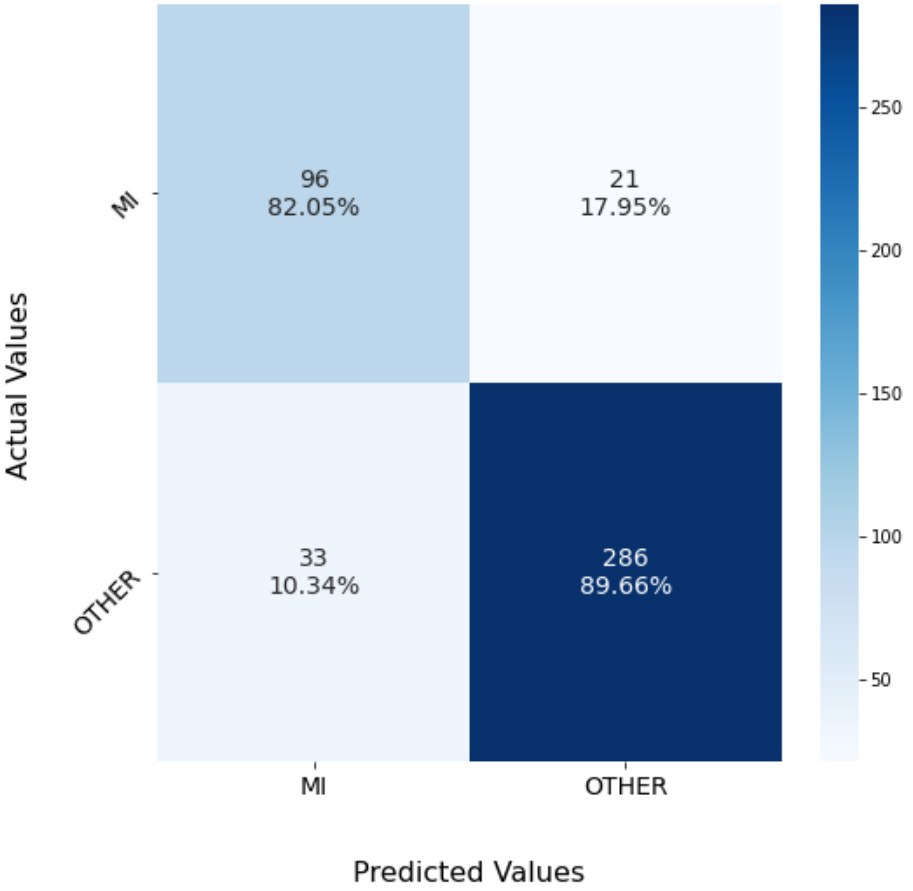
The same conclusions apply in this case: the OTHER label's great presence in the dataset seems to ponder the model to more accurately predict it.

The accuracy is around 12% better than in the PTB-XL case, but it might be more desirable in general to increase the sensitivity of MI.

Regular



MI Threshold = 20%



Conclusions

It is worth noting that it is not posible to directly compare the results from **A deep learning algorithm for detecting acute myocardial infarction [DLA]** (https://eurointervention.pcronline.com/article/a-deep-learning-algorithm-for-detecting-acute-myocardial-infarction) or the human comparison study (https://www.ahajournals.org/doi/10.1161/JAHA.113.000268#:~:text=The%20sensitivity%20to%20identify%20%E2%80%9Ctrue,1.10%2C%20P%3D0.01). with the ones presented here, since we do not know exactly how their observations or labeling look like.

Nevertheless, supposing we can compare to DLA in some way, we can see that there is a lot of room for improvement. There could be a handful of reasons for the presented results to be lower than the DLA ones:

- **Model Arquitecture:** Their model could be better in detecting and differentiating the labels. These could be given a more robust and overall bigger architecture. We are using relatively small models given our computing processing so increasing the capacity would most definitely improve our results.
- **Preprocessing:** We are still not implemeting preprocessing, like most of the best teams in the Physionet challenge did. It might be that the DLA team are indeed applying it and getting better results.
- **Data Quality:** It might be the case that our data is inheritenly not so good. We are using public databases which actually confirm that some of the labeling was done automatically by models, and also confirm that some of the data might not be correcly labeled. It is posible that the DLA used neat, correctly-labeled, noiseless data, which would certainly improve the results.
- **Class Imbalance:** It is worth noting that even though they have very few STEMI-NSTEMI registers and quite a lot of non-MI ones, they circumvented the class imbalance problem in some way. Knowing how they did it, whereas it was through weighted loss functions, data augmentation or some other way, could help us know how to avoid it as well.

Additional Observation

It is important to note that the data used by DLA was actually post-ECG validated by coronary angiograms to accurately detect STEMI and NSTEMI.

Our data was only revised through ECGs, and there is no actual differentiation labelwise as to whether an MI is STEMI or NSTEMI.

Because of this, if we use the data available at hand, we might be able to predict overall MIs but not to actually differentiate them properly. Also, knowing which is which could help the model upgrade its predictions. It would be ideal to verify the availability of their DB.