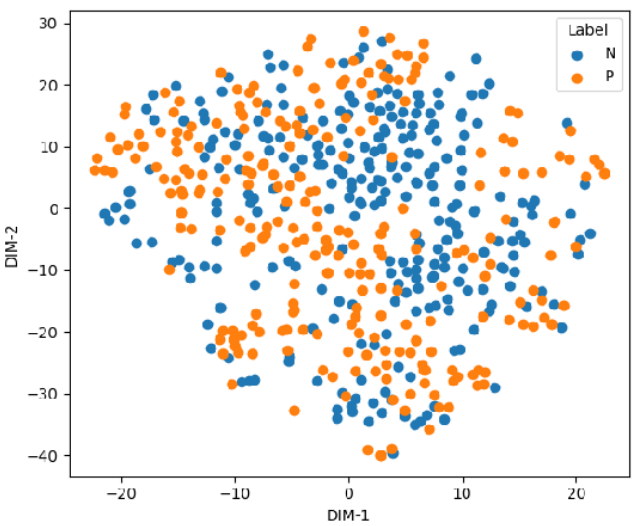
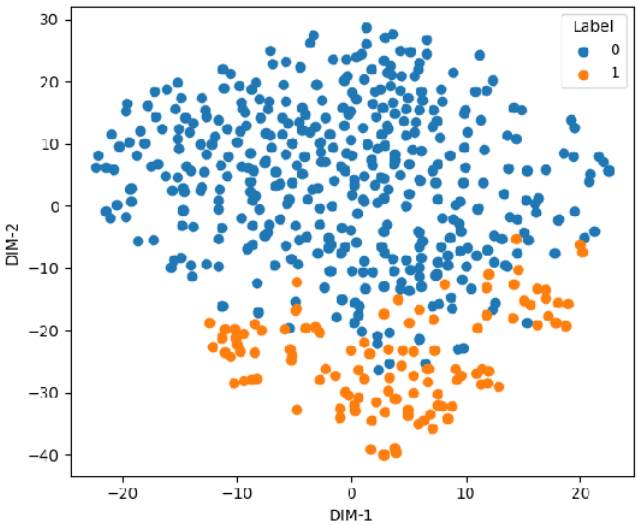
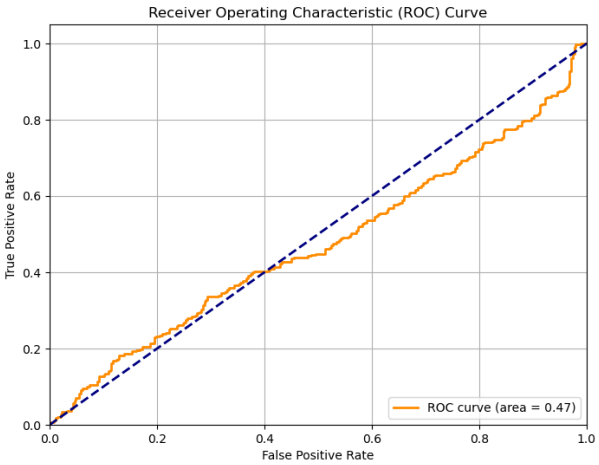
1. Changing the embedding layer affected the AUC quite a lot. Some layers (9/33) had good results (AUC 0.9/0.82) while others had worse results (AUC ~0.7)
2. A larger model did not lead to better results, however the smaller models also gave lower AUC.
3. Computing the distances to the training sample classes is a good baseline way to differentiate between classes. If the sample is close to the positive samples and far from the negative samples we'll get a large value and in the opposite case we'll get a small value.
   The log shrinks large distances avoiding cases where just being far from one group will classify you in the other (and in general more numerically stable).
4. On layer 9: batch_size=64, epochs=50, lr=0.01, hidden_dim=256, dropout=0.3 -> **AUC 0.927**
   (on layer 33 batch_size=32, epochs=50, lr=0.01, hidden_dim=128, dropout=0.2 -> **AUC=0.9091**)
5.
   a. Things like structural information, or additional metadata about the protein could be helpful.
   b. Encoding the information in a graph neural network, running a simple tree gradient boosting (XGBOOST). Also, things like RNN/transformers on the embedding of each amino acid instead of mean pooling can be helpful.
6.

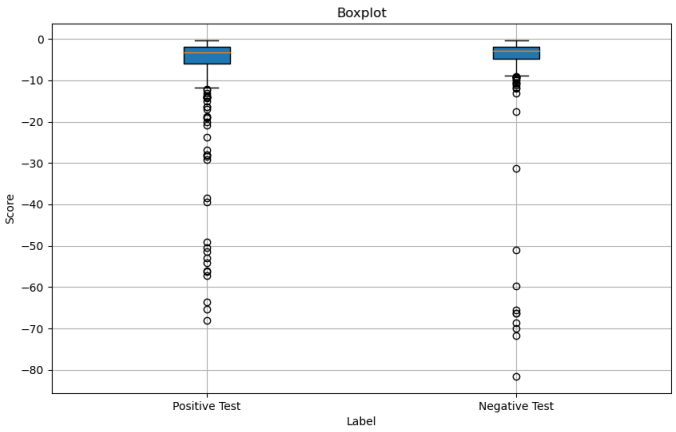




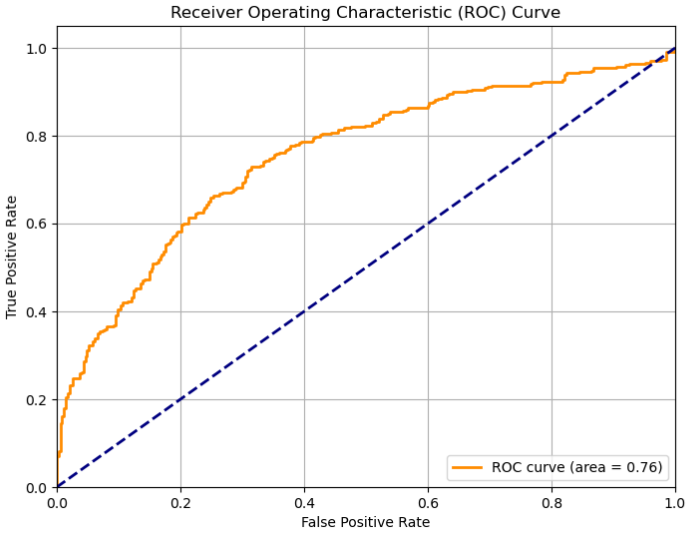
There doesn't seem to be a good separation between the data points in 2d and K means does not create meaningful clusters.

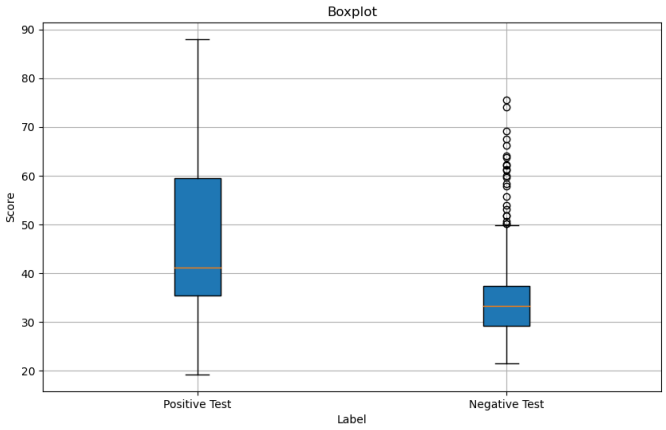
7. COM:





PLDDT:

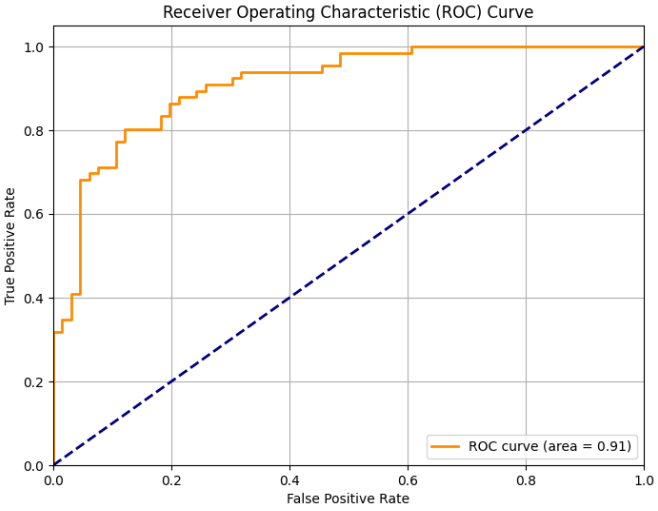


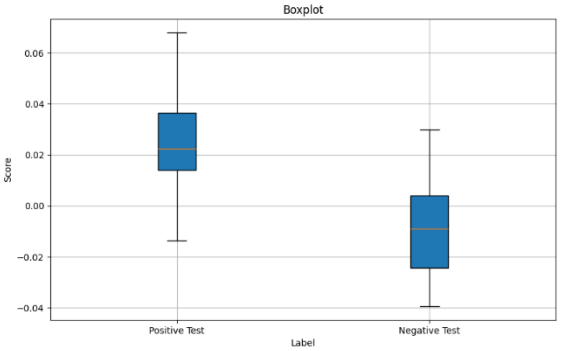


Plddt seems to work decently better than center of mass.

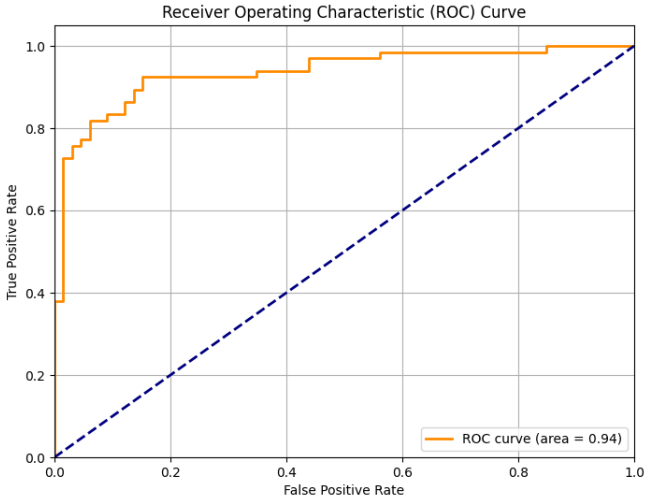
Other plots:

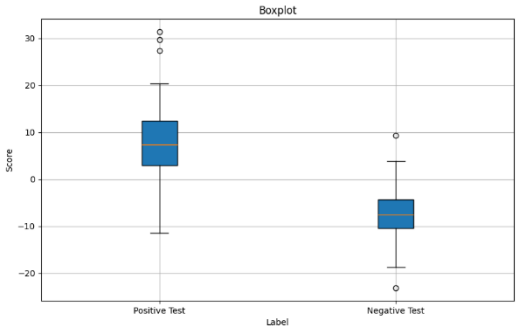
Baseline:





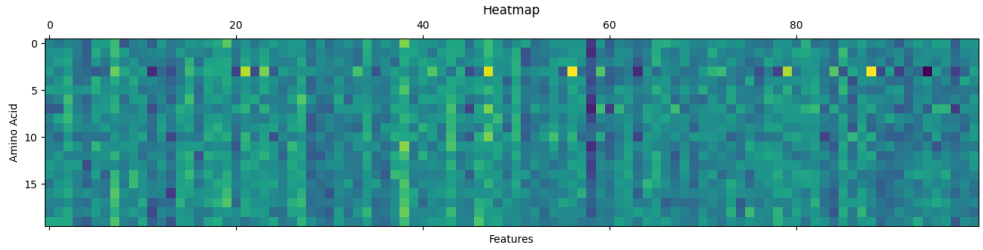
Network:





Heatmaps:

Positive:



Negative: