Team name - Machine_not_learning
Participants - Pranshu rastogi , Madhav Mathur , Suman Sahoo, Kshitij Mohan

The solution to the problem that we have providing is fairly simple and produces good score on 50% of data(i.e public LB)
Our approach is based on creating dense bert embeddings and using models over them

The embeddings are developed in 2 broad ways

1. Using pre trained Sentence bert models for better representation of the data like we used models (prominently)
    a. Paraphrase Multilingual Mpnet
    b. Paraphrase-mpnet-base-v2

These are some of the pre trained models that produces 768 dimensional embeddings

2. Fine tuning these pre models to the training data though we couldn't succeed in training longer as we don't have enough training resources we were only able to finetune it around 1 epoch which is fairly low and produces not good results

Modeling part -

Here we first used the KNN model as a very preliminary model as compatible with our resources that we could arrange. But Knn could produce very good results (64% single model) with these embeddings.

We also tried models like SVM , RF , like models but they either took too much of our gpu memory or took more than 10 hrs to produce results that are not very good as compared to the time and computation power trade off

Final ensemble
We only used a weighted ensemble of KNN prediction with embeddings from different models so indeed the more simple model the better the ensemble. reached(66.7%)

Tech stack - Pytorch for embedding generation ,Rapids(Cuml) for KNN are just used for our final solution. It could be replicated in 2-3 hrs with parallel execution.