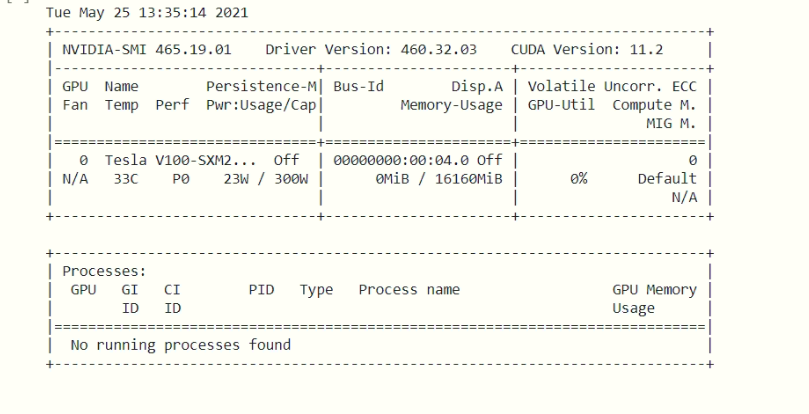
**DataFiles used in the competition**

1. We used only the datasets given by Zindi
2. We saved them in a folder and named it as – “AI4D Baamtu Datamation - Automatic Speech Recognition in WOLOF.zip”
3. The folder was zipped and loaded in google drive
4. The zipped folder was unzipped in the code for usage. This involved unzipping of clips as well

**Environment used:-**

1. Colab Pro was used with the GPU configurations mentioned in the notebook. Also with GPU we selected the “High RAM” option.
2. **Run the code only on Google Chrome (if you are running with Colab Pro)** else we found different results when we ran the same code from Firefox with same GPU configurations and same input files. This happens even after setting the seed for the model as well as for the entire torch environment.   
   This is a little bit surprising for us as well because we don’t know why this is happening.   
   Running the code on Google Chrome and colab pro with the said configurations resulted in reproducible results.
3. **GPU Configurations**- v100-sxm2-16gb GPU used on colab pro which gave around 12 hrs of training time. (The P100 will give you about 18 hours of training time)



**Approach used:-**

1. **Augmentation used:-**The observations having frequency of transcriptions less than 8 were augmented by selecting random samples from among themselves to ensure that their frequency reaches at least 8.   
   For example, If frequency of a transcription is 5, we will select 3 random samples from 5 and add it to the given data set to make sure the frequency of the transcription is 8
2. **Train Test Split-**We create a train test split with 90% of augmented data to be used for training and only 10% for validation. Stratification was applied to make sure that the data selected for validation covers almost all the transcriptions.
3. **Vocabulary creation-**The character by character vocabulary was created after removal of special characters from the text and we had 39 characters observed which were saved in a json file for tokenization is later steps
4. **Feature extraction and tokenization**The feature audio files were converted to a float array format and that will serve as a feature for the model to be used. Also tokenization was done using Wav2vec2tokenizer from HuggingFace.  
   While preparing model ready data, all columns apart from transcription one were dropped. The columns like Downvotes, Gender etc were dropped from the data.
5. **Model training**We used facebook/wav2vec2-large-xlsr-53 and the parameters were finetuned using Weights and biases API and the results attained are as a result of the finetuned parameters.   
   The training process took almost like 12 hours on the given GPU on google colab (and on google Chrome)
6. **Test results and finetuning the performance.**The model gave a word error rate of almost 4% on the test dataset but we did finetune the results using fuzzy-wuzzy module.  
   We used semantic similarity measure to find the similarity between the predicted text from the model and the existing transcription from the train data.   
   We replaced the model results with the similarity results only if the similarity was above 45%