How to reproduce the data coming from the report:
Trained dataset: tensorflow.csv (80:20 split)
Target/Inferred dataset: incubator-mxnet,pytorch,keras,caffe
Explaining folders and file naming:
All data produced are located in two folders:

1. base_line_tf/ :
    a. This folder holds all information for MLPBatchNorm results and MLPLayerNorm in the form of csv file
    b. In general the columns of the data across all csv files are iteration,Accuracy,Precision,Recall,F1,AUC, where there are 20 iterations each
    c. CSV file named base_data_tensorflowlayernorm and base_data_tensorflow_batchnorm holds information on the MLPBatchNorm and MLPLayerNorm key metrics performance on the validation dataset from the training dataset , iterations in the files does have a corresponding model in the models/baseline_batchnorm and models/baseline_layernorm respectively.
        i. Each model naming in the models folder follows a convention with the following "baseline_model_tensorflow_{batchnorm/{layernorm}{iteration}_iteration"
        ii. The iteration correlates to the iteration found in base_data_tensorflowlayernorm.csv or base_data_tensorflow_batchnorm.csv
    d. Results from base models inferred with other datasets follow the convention base_data_tensorflow_{batchnorm/layernorm}{targetdataset}
        i. Target dataset here points to the inferred dataset tested against the respective model
    e. To indicate result after TENT is applied the following convention is followed base_data_tensorflow_{batchnorm/layernorm}tent{targetdataset}
2. baselines_data/:
    a. This folder holds all information for NaiveBayes results and the datasets it is inferred against
    b. tensorflow_NB_detailed_baseline.csv holds the key metrics performance on the validation dataset from the training dataset
    c. The other result are located in filenames with the convention of {target_dataset }_NB_tensorflow_based_detailed_baseline

All models are saved in .pt form saved inside models directory:
All models saved are saved using pytorch, where weightsOnly=False, as such it is important if it requires you to test the model independent from the scripts provided, make sure you include the correct modeimport for simple_mlp.py

1. Baseline_batchnorm holds all models batch_norms with their iteration
2. This is true also for the baseline_layernorm folder

To reproduce all the the dataset the following must be ensured:

If you run any of the scripts mentioned above but base_line_tf and baselines_data folders are not empty , it will append values to existing csv make sure to save old csv in another folder before doing so or you could just empty these values in the csv files

Systems Tested:
1. MacOS
2. Linux (It is recommended to use Linux to ensure all commands can be executed)

Warning: Windows is not Tested as I do not have a Windows System

To automatically run training and inference for Naive Bayes Classifier, here are the requirements

Main file: br_classification_inference_on_other_datasets.py
Filepath: ISE-coursework/br_classification_inference_on_other_datasets.py
Command to run : invoke normally with python from terminal or ipynb also works (this will be true for all commands listed below
I personally run it in terminal with the command: `python br_classification_inference_on_other_datasets.py`

To automatically run training only for MLP with BatchNorm Layers with validation test results:

Main file: train_normal_NN_BatchNorm.py
Filepath: ISE-coursework/run_all_iteration_train_NN_batchnorm.py
Command to run :`python run_all_iteration_train_NN_batchnorm.py`
What it does: Spawn 4 parallel instances until 20 instances are reached to reach the 20 iterations requirement for robustness, , saves results, and model, This program runs the python script called train_normal_NN_BatchNorm.py which accepts an argument called –iteration to track iteration,you can run this python script on its own but make sure to do it 20 times. . All results will be saved in accordance with the files and folders roles explained above
Warning:

1. Only tested in Linux
2. Will cause system lag as it spawn 4 parallel instances of terminal if your system can't handle it as this was tested on 14 core system with 64 GB of RAM to speed up the process.

To automatically run training only for MLP with BatchNorm Layers with validation test results:

Main file: train_normal_NN_BatchNorm.py
Filepath: ISE-coursework/run_all_iteration_train_NN_layernorm.py
Command to run :`python run_all_iteration_train_NN_layernorm.py`
What it does: Spawn 4 parallel instances until 20 instances spawned are reached to reach the 20 iterations requirement for robustness, saves results, and model.This program runs the python script called train_normal_NN_LayerNorm.py which accepts an argument called –iteration to track iteration, you can run this python script on its own but

make sure to do it 20 times. All results will be saved in accordance with the files and folders roles explained above

Warning:

1. Only tested in Linux
2. Will cause system lag as it spawn 4 parallel instances of terminal if your system can't handle it as this was tested on 14 core system with 64 GB of RAM to speed up the process of training.

To automatically run inference only for MLP with BatchNorm Layers to target dataset(no tent):

Prerequisite:

1. You need to have run_all_iteration_train_NN_batchnorm.py already executed
2. Open classification_all.ipynb
3. Run all commands not passing over a section called "Base Line performance on MLP models" this will help you figure out which models for BatchNorm and LayerNorm are closest to the mean. Warning: This can be different on each different runs of run_all_iteration_train_NN_batchnorm.py
4. Go to the file infer_without_tent_batchnorm.py. Edit model_base variable to choose which model to run in accordance with the iteration number you received after doing step 3,

```
model_base =
torch.load('models/baseline_batchnorm/baseline_model_tensorflow_batchnorm{change the number here}_iteration.pt',map_device=device)
```

Running the file:

1. After doing the prerequisite steps run the command python infer_without_tent_batchnorm.py This would run the command to run inference on all target datasets each with 20 repetitions and save it in accordance to the file naming explanation above

To automatically run inference only for MLP with BatchNorm to target dataset(TENT):

Prerequisite:

1. You need to have run_all_iteration_train_NN_batchnorm.py already executed
2. Open classification_all.ipynb
3. Run all commands not passing over a section called "Base Line performance on MLP models" this will help you figure out which models for BatchNorm and LayerNorm are closest to the mean. Warning: This can be different on each different runs of run_all_iteration_train_NN_batchnorm.py
4. Go to the file infer_with_tent_batchnorm.py. Edit model_base variable to choose which model to run in accordance with the iteration number you received after doing step 3, refer to "**To automatically run inference only for MLP with BatchNorm Layers to**

target dataset(no tent) section" You just need to change the number on the base model basically before running the file

Running the file:

1. After doing the prerequisite steps run the command `python infer_with_tent_batchnorm.py` This would run the command to run inference on all target datasets each with 20 repetitions and save it in accordance to the file naming explanation above

To automatically run inference only for MLP with Layernorm to target dataset(No TENT):
Prerequisite:

1. You need to have `run_all_iteration_train_NN_layernorm.py` already executed
2. Open classification_all.ipynb
3. Run all commands not passing over a section called "Base Line performance on MLP models" this will help you figure out which models for BatchNorm and LayerNorm are closest to the mean. Warning: This can be different on each different runs of run_all_iteration_train_NN_layernorm.py
4. Go to file infer_with_tent_layernorm.py. Edit model_base variable to choose which model to run in accordance with the iteration number you received after doing step 3, refer to "**To automatically run inference only for MLP with BatchNorm Layers to target dataset(no tent) section"** You just need to change the number on the base model basically before running the file

Running the file:

1. After doing the prerequisite steps run the command `python infer_without_tent_layernorm.py` This would run the command to run inference on all target datasets each with 20 repetitions and save it in accordance to the file naming explanation above

To automatically run inference only for MLP with LayerNorm to target dataset (TENT):
Prerequisite:

1. You need to have `run_all_iteration_train_NN_layernorm.py` already executed
2. Open classification_all.ipynb
3. Run all commands not passing over a section called "Base Line performance on MLP models" this will help you figure out which models for BatchNorm and LayerNorm are closest to the mean. Warning: This can be different on each different runs of run_all_iteration_train_NN_layernorm.py
4. Go to line 41 of the file infer_with_tent_batchnorm.py. Edit model_base variable to choose which model to run in accordance with the iteration number you received after doing step 3, refer to "**To automatically run inference only for MLP with BatchNorm Layers to target dataset(no tent) section"** You just need to change the number on the base model basically before running the file

Running the file:

1. After doing the prerequisite steps run the command python infer_with_tent_layernorm.py This would run the command to run inference on all target datasets each with 20 repetitions and save it in accordance to the file naming explanation above

After doing all the steps above, you will have all the data required, for analysis and table generation that are found in the report run all remaining cells, in classification_all.ipynb.

Extra files:

1. br_classification_with_mlp_tent_batchnorm.ipynb: This file shows the first trial of trying all functions before automating them into script for MLP BatchNorm layers
2. br_classification_with_mlp_tent_layernorm.ipynb: This file shows the first trial of trying all functions before automating them into script for MLP LayerNorm layers
3. br_classification.py: the original file from lab1 solution
4. TextDataset.py: Custom pytorch dataset to load the chosen dataset i
5. tent.py: file for calling methods to adapt MLP models with TENTnto TF-IDF format
6. simple_mlp.py: MLP models with LayerNorm and BatchNorm1d Layer