🔁 Experiment Manual (detailed explanation in reproduce.pdf)

📁 Folder Structure and File Naming Conventions

Main Folders:

- `base_line_tf/`: Contains results for MLP models with BatchNorm and LayerNorm.
- `baselines_data/`: Contains results from Naive Bayes Classifier.
- `models/`: Stores trained `.pt` model files.

CSV Naming Conventions:

- `base_data_tensorflowlayernorm.csv` and `base_data_tensorflow_batchnorm.csv`: Baseline results (on training/validation set).
- `base_data_tensorflow_{layernorm/batchnorm}{targetdataset}.csv`: Inference results from other datasets.
- `base_data_tensorflow_{layernorm/batchnorm}_tent{targetdataset}.csv`: Inference results after TENT.
- `{target_dataset}_NB_tensorflow_based_detailed_baseline.csv`: Naive Bayes inference results.

Model File Naming:

- `baseline_model_tensorflow_{batchnorm/layernorm}{iteration}_iteration.pt`

🧠 Models and Training

MLP with BatchNorm

- Script: `run_all_iteration_train_NN_batchnorm.py`
- Command: `python run_all_iteration_train_NN_batchnorm.py`
- Spawns 4 parallel training sessions to complete 20 iterations.
- Models saved in: `models/baseline_batchnorm/`

MLP with LayerNorm

- Script: `run_all_iteration_train_NN_layernorm.py`
- Command: `python run_all_iteration_train_NN_layernorm.py`
- Spawns 4 parallel training sessions to complete 20 iterations.
- Models saved in: `models/baseline_layernorm/`

> **Warning**: Scripts are optimized for Linux systems with at least 14 cores and 64 GB RAM.

🔍 Inference Setup

Naive Bayes Classifier

- Script: `br_classification_inference_on_other_datasets.py`
- Command: `python br_classification_inference_on_other_datasets.py`
- Outputs go to: `baselines_data/`

MLP Inference (BatchNorm / LayerNorm)

Step 1: Identify the best model (closest to the mean) using `find_closest_to_mean_{layernorm/batchnorm}_model.py`

Step 2: Modify model_base variabl

```
model_base =
torch.load('models/baseline_batchnorm/baseline_model_tensorflow_batchn
orm{iteration}_iteration.pt')
```

Without TENT:

- BatchNorm: `infer_without_tent_batchnorm.py`
- LayerNorm: `infer_without_tent_layernorm.py`

With TENT:

- BatchNorm: `infer_with_tent_batchnorm.py`
- LayerNorm: `infer_with_tent_layernorm.py`
- Command (for all): `python <filename>.py`
- Repeats inference 20 times, results are saved automatically.

⚠️ Important Notes

- Always empty or back up CSV files before re-running experiments to avoid duplicated results.
- Results are deterministic per run, so reruns may yield different "closest to mean" models.
- Windows OS is untested; Linux is recommended for all operations.

📚 Additional Files

- `classification_all.ipynb`: Generates tables and performs analysis for the report.
- `br_classification_with_mlp_tent_batchnorm.ipynb` / `br_classification_with_mlp_tent_layernorm.ipynb`: Early TENT

experimentation.

- `br_classification.py`: Original classification file from lab1.
- `TextDataset.py`: Custom TF-IDF-compatible PyTorch dataset used for nlp models compplete with preprocessing pipeline.
- `tent.py`: Implements TENT-based adaptation.
- `simple_mlp.py`: Defines MLP architectures (BatchNorm and LayerNorm versions).