==

Replication Guide - BERT Text Classification
==

🎯 Objective

This guide outlines the steps required to fully replicate the experiments in the project, including BERT-based classification, TF-IDF+Naive Bayes, and parameter sensitivity analysis.

🧩 Requirements

- Python version: 3.8 or later
- Install dependencies:

```
pip install -r requirements.txt
```

- Required files:

- Title+Body.csv	(dataset, must be in project root)
- BERT.py	(BERT classifier script)
- NB+TF-IDF.py	(Naive Bayes baseline)
- Sensitivity Analysis of Parameters.py	(optional analysis)
- requirements.txt	(dependency list)

📄 Dataset Format

The Title+Body.csv file must include at least two columns:

- text: A string combining the bug report title and body.
- sentiment: Integer label (e.g., 0 or 1).

Ensure no missing values — the scripts will fill empty cells with '' by default.

🚀 Replication Steps

A. Run BERT Experiment

1. Run the BERT classifier:

```
python BERT.py
```

2. The script will:
    - Perform 10 training/evaluation rounds (configurable via REPEAT in script).
    - Use bert-base-uncased via HuggingFace Transformers.

  - Compute Accuracy, Precision, Recall, F1, and AUC for each round.
  - Save averaged results in pytorch_BERT.csv.

B. Run Baseline (Naive Bayes + TF-IDF)

```
python "NB+TF-IDF.py"
```

- Perform 10 training/evaluation rounds (configurable via REPEAT in script).
- Compute Accuracy, Precision, Recall, F1, and AUC for each round.
- Save averaged results in pytorch_NB.csv.

C. Run Sensitivity Analysis

```
python "Sensitivity Analysis of Parameters.py"
```

- This script explores how changing training parameters affects the BERT model.
- Testing 'learning_rate': [2e-5, 3e-5, 5e-5], 'batch_size': [8, 16, 32], 'epochs': [2, 3, 4], 'threshold': [0.4, 0.5, 0.6]
- Save results in pytorch_BERT_sensitivity.csv'.

📁 Output Files

- pytorch_BERT.csv: Final averaged results of BERT model.
- pytorch_NB.csv: Final averaged results of NB+TF-IDF.
- pytorch_ BERT_sensitivity.csv: Sensitivity Analysis results of BERT parameters.

⚙️ Configuration Notes

- You can change model, training epochs, or evaluation strategy directly in BERT.py.
- You can also swap BERT model via:

```
model_name = 'bert-base-uncased'    # change to another like 'bert-base-cased'
```

📌 Troubleshooting

- Ensure the dataset is in the correct format and in the root directory.
- Make sure your environment matches the versions in requirements.txt.
- For large datasets or faster training, GPU is recommended.

Happy Replicating! 🔁