Intelligent Bug Report Classification: A Hybrid Approach Combining Tf-IDF, GloVe, and Meta Features

Manual

1. Ensure all dependencies listed in *requirements.pdf* are installed.
2. Make sure glove.6B.100d.txt is in the same directory as the python scripts (download it if you don't have it by following *requirements.pdf*).
3. Make sure all the CSV datasets are in the same directory as the python scripts (it should already be).
4. Open your terminal and run:
   a. *Python improved_br_classification.py*

The GUI window should launch automatically.
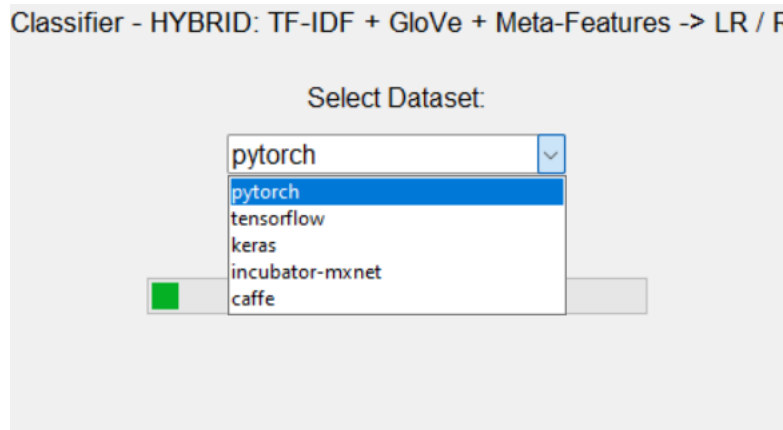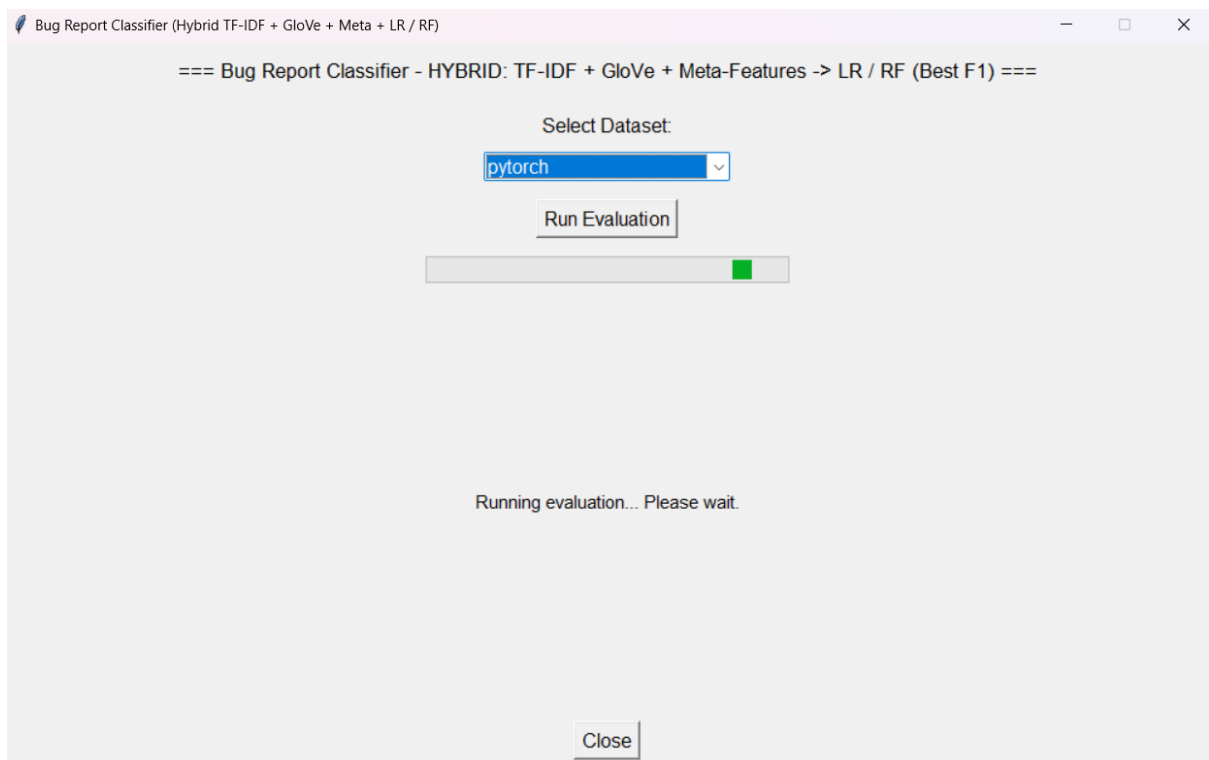
Using the GUI

Once launched, you should see the GUI:
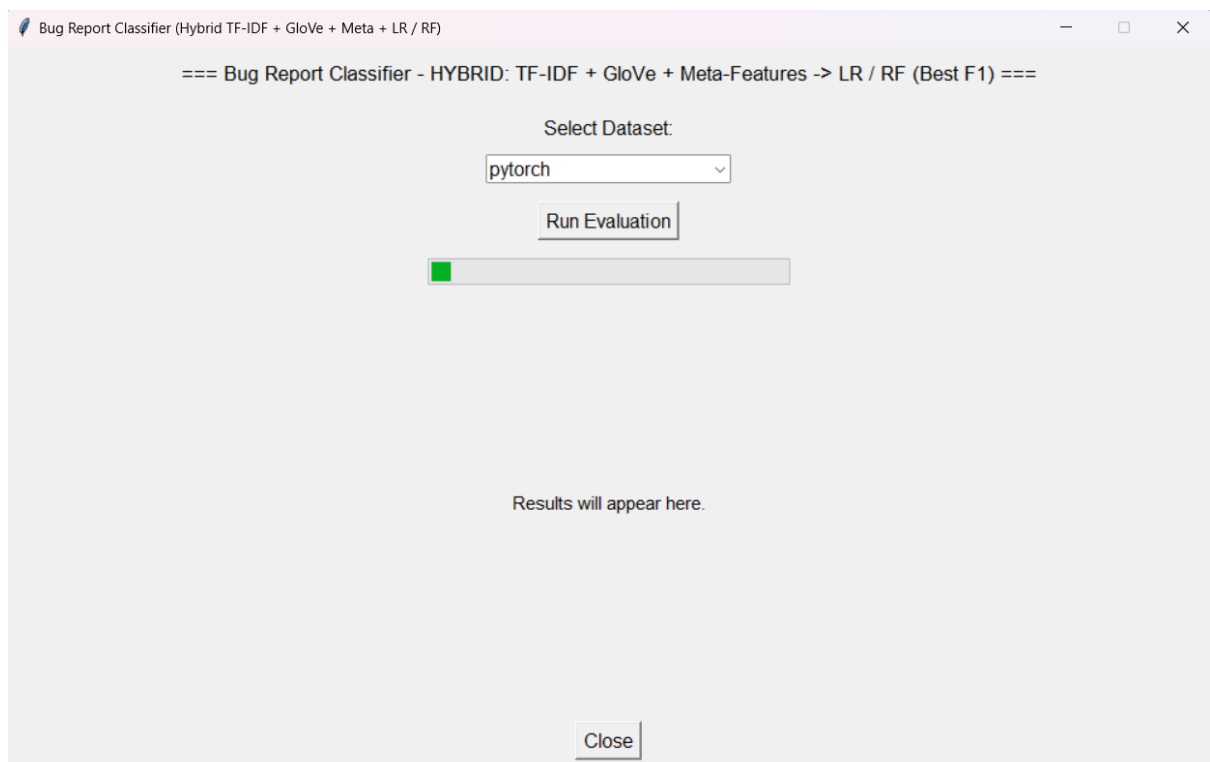
1. Select the dataset you want to evaluate from the options in the dropdown menu:
   a. pytorch
   b. tensorflow
   c. keras
   d. incubator-mxnet
   e. caffe
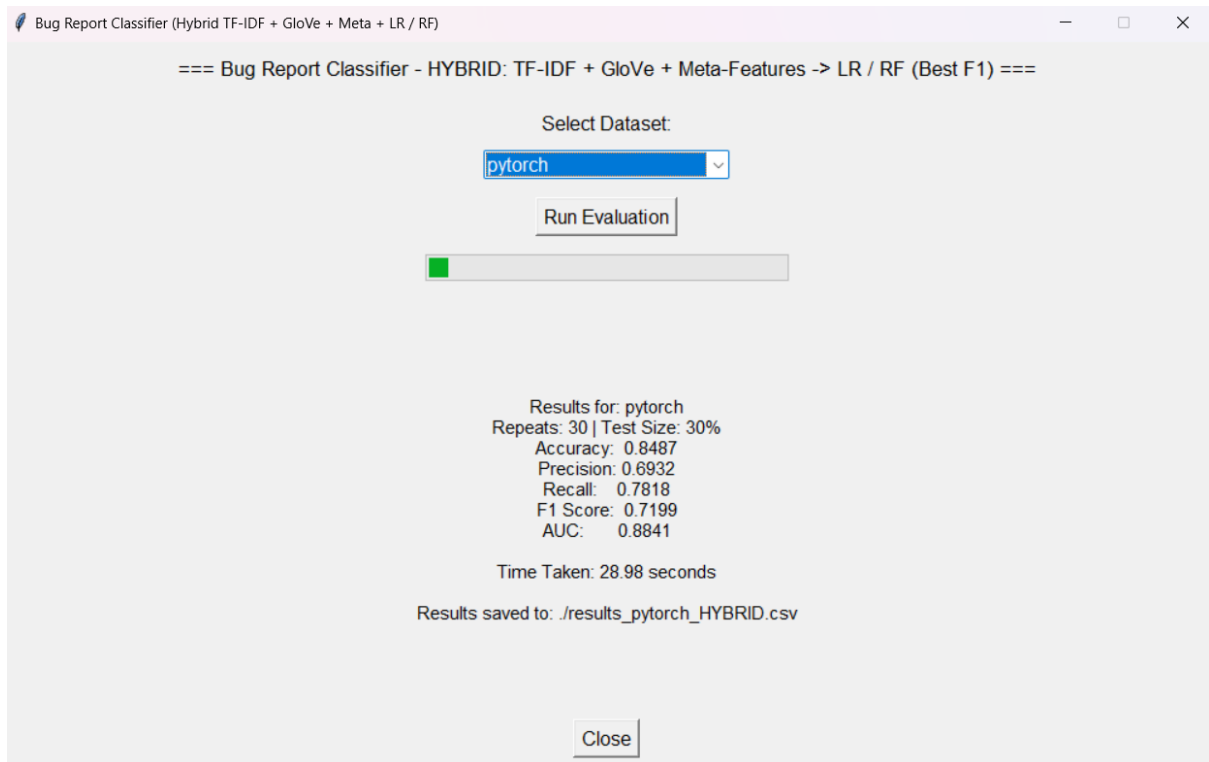
Classifier - HYBRID: TF-IDF + GloVe + Meta-Features -> LR / F

Select Dataset:

pytorch ⌄

pytorch
tensorflow
keras
incubator-mxnet
caffe

2. Click the "Run Evaluation" button.
   a. You should see a message "*Running Evaluation... Please wait*"
   b. A useless progress bar will also animate to keep you company while you wait!

Bug Report Classifier (Hybrid TF-IDF + GloVe + Meta + LR / RF) — □ ×

=== Bug Report Classifier - HYBRID: TF-IDF + GloVe + Meta-Features -> LR / RF (Best F1) ===

Select Dataset:

pytorch ⌄

Run Evaluation

Running evaluation... Please wait.

Close

3. View Results:
   a. After a short delay of around 20-60 seconds depending on the dataset chosen, the results will appear.
   b. Metrics shown include:
      i. Accuracy
      ii. Precision
      iii. Recall
      iv. F1 Score
      v. AUC

vi. Time Taken

=== Bug Report Classifier - HYBRID: TF-IDF + GloVe + Meta-Features -> LR / RF (Best F1) ===

Select Dataset:
pytorch

Run Evaluation

Results for: pytorch
Repeats: 30 | Test Size: 30%
Accuracy: 0.8487
Precision: 0.6932
Recall: 0.7818
F1 Score: 0.7199
AUC: 0.8841

Time Taken: 28.98 seconds

Results saved to: ./results_pytorch_HYBRID.csv

Close

4. Close Tool:

   a. Click the "Close" button to exit the tool

5. Output files:

   a. After each run, a CSV file is generated with the evaluation logs containing the metrics.

   b. The CSV file will be saved as *results_[dataset]_HYBRID.CSV*

   c. Each contains

      i. Accuracy, Precision, Recall, F1, AUC

      ii. List of 30 AUC scores (for statistical analysis)

      iii. List of 30 F1 scores (for statistical analysis)

      iv. Evaluation Time

After opening CSV file:

A	B	C	D	E	F	G	H	I
Time_Sec	repeated_	Accuracy	Precision	Recall	F1	AUC	CV_list(AU	CV_list(F1)
33.01642	30	0.848673	0.693234	0.781755	0.719868	0.884126	[np.float6	[0.74117064
40.78875	30	0.848673	0.693234	0.781755	0.719868	0.884126	[np.float6	[0.74117064
28.97753	30	0.848673	0.693234	0.781755	0.719868	0.884126	[np.float6	[0.74117064

To run the Baseline code:

1. Open the lab1 directory
2. Open the python code file *br_classification.py*
3. Navigate the code to find "*Project =*" and enter the name of the dataset you want to evaluate. E.g, for *caffe*:

```
# Choose the project (options: 'pytorch', 'tensorflow', 'keras', 'incubator-mxnet', 'caffe')
project = 'caffe'
path = f'{project}.csv'
```

4. Open the terminal and run:
   a. *Python br_classification.py*
5. After a short delay the results will print in the terminal:

```
=== Naive Bayes + TF-IDF Results ===
Number of repeats:     30
Average Accuracy:      0.5310
Average Precision:     0.5690
Average Recall:        0.6656
Average F1 score:      0.4669
Average AUC:           0.7300
Time Taken:            7.68 seconds

Results have been saved to: ../results_caffe_NB.csv

Process finished with exit code 0
```

6. The [dataset]_NB.csv file will be saved outside the lab1 folder.