Muneeb Parwaz
Student ID: 2448869

Replication of Results Reported

Table 1:

Experiment	Accuracy	Precision	Recall	F1-score	AUC
TF-IDF + NB (baseline)	*0.625827815*	*0.609021933*	*0.747575464*	*0.555144187*	*0.747575464*
TF-IDF + NB without stop-word removal	0.609933775	0.606261803	0.739621698	0.543441129	0.739621698
TF-IDF baseline with Comments included from bug reports	0.565562914	0.597830512	0.72101522	0.511415018	0.72101522
TF-IDF baseline with Labels included	0.709271523	0.625938841	0.768447891	0.613627993	0.768447891
TF-IDF baseline with Code Snippets included	0.631788079	0.608497595	0.743277987	0.558226446	0.743277987
TF-IDF baseline with Labels AND Code Snippets incl.	0.669536424	0.617454989	0.75834931	0.585876292	0.75834931

How to replicate results from table 1:

- Experiment 1: TF-IDF + NB (baseline) can be replicated by running the tool with the baseline configs:

```
project = 'pytorch'
REPEAT_TIMES = [10]
IncludeCommentsFromBugReports = False
IncludeCodeSnippetsAndErrorLogsFromBugReports = False
IncludeLabelsFromBugReports = False
Method = 'TFIDFNaiveBayes'
RemoveHTMLTags = True
RemoveEmoji = True
RemoveStopWords = True
CleanString = True
UseGridSearchCVForBERTLR = False
```

  - *Output file is called pytorch_NB.csv*
- For experiment 2: set “RemoveStopWords” to False and then save and re-run the tool.
  - *Output file is called pytorch_NB_KeepStopwords.csv*
- For experiment 3: set “IncludeCommentsFromBugReports” to True and then re-run the tool.
  - *Output file is called pytorch_NB_CommentsIncluded.csv*
- For experiment 4: set “IncludeLabelsFromBugReports” to True and then re-run the tool.
  - *Output file is called pytorch_NB_LabelsIncluded.csv*
- For experiment 5: set “IncludeCodeSnippetsAndErrorLogsFromBugReports” to True and then re-run the tool.
  - *Output file is called pytorch_NB_CodeSnippetsAndErrorLogsIncluded.csv*
- For experiment 5: set both “IncludeLabelsFromBugReports” and “IncludeCodeSnippetsAndErrorLogsFromBugReports” to True and then re-run the tool.
  - *Output file is pytorch_NB_CodeSnippetsAndErrorLogsIncluded_LabelsIncluded.csv*

Table 2:

Model	Accuracy	Precision	Recall	F1-score	AUC
TF-IDF + NB *(baseline)*	*0.625827815*	*0.609021933*	*0.747575464*	*0.555144187*	*0.747575464*
BERT + LR	0.871523179	0.699985972	0.632523249	0.652598514	0.632523249

How to replicate results from Table 2:

- for TF-IDF + NB (baseline), run the tool with the following configs:

```
project = 'pytorch'
REPEAT_TIMES = [10]
IncludeCommentsFromBugReports = False
IncludeCodeSnippetsAndErrorLogsFromBugReports = False
IncludeLabelsFromBugReports = False
Method = 'TFIDFNaiveBayes'
RemoveHTMLTags = True
RemoveEmoji = True
RemoveStopWords = True
CleanString = True
UseGridSearchCVForBERTLR = False
```

  - *Output file is called pytorch_NB.csv*
- for BERT + LR, set "Method" to "BERTLogisticRegression", and then save and rerun the tool.
  - *Output file is called pytorch_BERT-LR.csv*

Table 3:

Dataset	Model	Accuracy	Precision	Recall	F1-score	AUC
PyTorch	TF-IDF + NB	*0.625827815*	*0.609021933*	*0.747575464*	*0.555144187*	*0.747575464*
	BERT + LR	0.871523179	0.699985972	0.632523249	0.652598514	0.632523249
TensorFlow	TF-IDF + NB	*0.561744966*	*0.635961066*	*0.72267484*	*0.539710278*	*0.72267484*
	BERT + LR	0.863758389	0.773706983	0.728266	0.745852715	0.728266
Keras	TF-IDF + NB	0.560447761	0.629413581	0.698389094	0.540072792	0.698389094
	BERT + LR	0.829850746	0.726213108	0.687601091	0.701687124	0.687601091
MXNet	TF-IDF + NB	*0.606730769*	*0.613445133*	*0.749989724*	*0.547106142*	*0.749989724*
	BERT + LR	0.888461538	0.761363159	0.679620009	0.703560512	0.679620009
Caffe	TF-IDF + NB	*0.522413793*	*0.55724276*	*0.623421921*	*0.442683432*	*0.623421921*
	BERT + LR	0.875862069	0.570779282	0.533520415	0.535300613	0.533520415

How to replicate results from Table 3:

- The first pair of rows (under PyTorch) come from table 2, so can be obtained in the same way as described on the previous page.

- To get the remaining 8 rows, change the "project" variable from 'pytorch' to the 4 other projects names ('tensorflow', 'keras', 'incubator-mxnet' and 'caffe').
- Then for each of these projects, run the tool twice. Once with "Method" set to "TFIDFNaiveBayes" and the second time with Method="BERTLogisticRegression".
  - *The output files (in order in which they appear in Table 3) are called:*
    - *pytorch_NB.csv*
    - *pytorch_BERT-LR.csv*
    - *tensorflow_NB.csv*
    - *tensorflow_BERT-LR.csv*
    - *keras_NB.csv*
    - *keras_BERT-LR.csv*
    - *incubator-mxnet_NB.csv*
    - *incubator-mxnet_BERT-LR.csv*
    - *caffe_NB.csv*
    - *caffe_BERT-LR.csv*

Table 4:

Model	Accuracy	Precision	Recall	F1-score	AUC
BERT + LR with stop-word removal *(new baseline)*	*0.871523179*	*0.699985972*	*0.632523249*	*0.652598514*	*0.632523249*
BERT + LR without stop-word removal	0.886092715	0.756221254	0.650873254	0.678438532	0.650873254
BERT + LR with Comments included	0.873509934	0.700764192	0.620798251	0.63706784	0.620798251
BERT + LR with Labels included	0.869536424	0.687540901	0.621969034	0.641102365	0.621969034
BERT + LR with Code Snippets included	0.878145695	0.730814982	0.632696434	0.655390815	0.632696434

How to replicate results from Table 4:

- For row 1, run the tool with the following configurations:

```
project = 'pytorch'
REPEAT_TIMES = [10]
IncludeCommentsFromBugReports = False
IncludeCodeSnippetsAndErrorLogsFromBugReports = False
IncludeLabelsFromBugReports = False
Method = 'BERTLogisticRegression'
RemoveHTMLTags = True
RemoveEmoji = True
RemoveStopWords = True
CleanString = True
UseGridSearchCVForBERTLR = False
```

  - *Output file is called pytorch_BERT-LR.csv*
- For experiment 2, set "RemoveStopWords" to False, and then save and re-run the tool.
  - *Output file is called pytorch_BERT-LR_KeepStopwords.csv*
- For experiment 3, set "IncludeCommentsFromBugReports" to True, and then save and re-run the tool.
  - *Output file is called pytorch_BERT-LR_CommentsIncluded.csv*
- For experiment 4, set "IncludeLabelsFromBugReports" to True, and then save and re-run the tool.
  - *Output file is called pytorch_BERT-LR_LabelsIncluded.csv*
- For experiment 5, set "IncludeCodeSnippetsAndErrorLogsFromBugReports" to True, and then save and re-run the tool.
  - *Output file is called pytorch_BERT-LR_CodeSnippetsAndErrorLogsIncluded.csv*

How to replicate statistical test results highlighted in each table above, and reported in the table below:

Metric	Test Statistic	P-value	Significantly different? (p < 0.05)	Interpretation
AUC	0.0	1.862645149230957e-09	Yes	BERT + LR is significantly better
Accuracy	0.0	1.7245993818153558e-06	Yes	BERT + LR is significantly better
Precision	6.0	2.60770320892334e-08	Yes	BERT + LR is significantly worse
Recall	0.0	1.862645149230957e-09	Yes	BERT + LR is significantly better
F1 Score	4.0	1.30385160446167e-08	Yes	BERT + LR is significantly worse

1. After running two different models in main.py, open the csv output file generated for each model.
2. Open the statisticalTest.py file
3. Copy the list of Auc, Accuracy, Precision, Recall and F1 scores from the output file of the model (found in the last 5 columns of the CSV)
4. Paste them into lines 5-9 for the first model, and onto 12-16 for the second model (into their corresponding variables)
5. Save and run the file by running the command: *python statisticalTest.py*
6. View the results of the statistical test for each evaluation metrics in the output:

```
Test Results:
Test Statistic: 23.0000
P-value: 0.6953
🔴 No statistically significant difference in Auc (p >= 0.05)
The difference could be due to random chance.
Test Statistic: 0.0000
P-value: 0.0020
🟢 The difference in Accuracy is statistically significant (p < 0.05)
The new model significantly outperforms the Baseline
Test Statistic: 0.0000
P-value: 0.0020
🟢 The difference in Precision is statistically significant (p < 0.05)
The new model significantly outperforms the Baseline
Test Statistic: 23.0000
P-value: 0.6953
🔴 No statistically significant difference in Recall (p >= 0.05)
The difference could be due to random chance.
Test Statistic: 0.0000
P-value: 0.0020
🟢 The difference in F1 is statistically significant (p < 0.05)
The new model significantly outperforms the Baseline
```

Note: the highlighted statistical comparisons in tables 1-4 used the list of values from the output csv files from the row where repeated_times=10, whereas the table in Appendix 3 – which compares values from the output files *"pytorch_NB.csv"* and *"pytorch_BERT-LR_KeepStopwords.csv"* – uses the row where repeated_times=30.