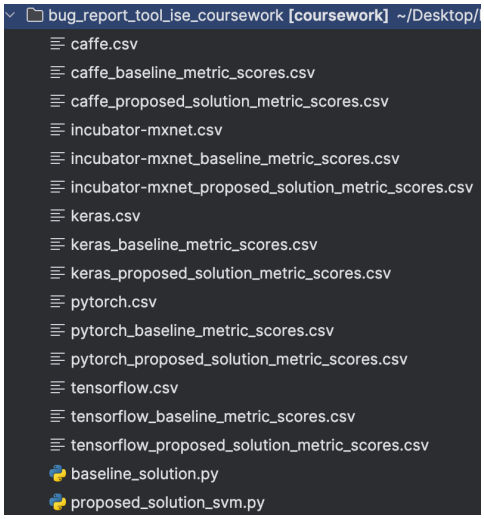
Using the Proposed Solution Tool

Please note that since we have implemented gridSearchCV to find the optimal parameter settings from a set of possible parameters. The time to execute my proposed solution, i.e. proposed_solution_svm.py, can vary from 45 seconds for the caffe (which is the smallest dataset) to 15 minutes for the tensorflow dataset (which is the largest dataset). This is due to gridSearchCV trying every possible parameter setting combination and conducting 10-fold cross validation.

The raw results which include the values of the metrics across 10 runs, the averages of these metrics, Wilcoxon rank sum test between the proposed solution and baseline solution and standard deviation results are in the file named, {project}_proposed_solution _metric_scores.csv, where project is the name of the project specified in line 60, which has been used as the dataset to train an SVM model (please see step 3 below for more details). Please follow step 1 and step 2 as described in the 'Using the Proposed Solution Tool' section above.

Step 1: Download **bug_report_tool_ise_courswork** folder in the GitHub Repository.
Step 2: Please ensure that the **datasets** (**caffe.csv, incubator-mxnet.csv, keras.csv, pytorch.csv, and tensorflow.csv**), **proposed_solution_svm.py** and **baseline_solution.py** are in the same directory as shown below: The folder will also include the metric_scores.csv which are the raw results.



Step 3: In our tool, **proposed_solution_svm.py** at **line 60,** we have set a **default dataset, 'caffe'**. Please replace this string with any other project name as appropriate to use our tool to obtain the reported experiment results on a specified dataset, the available datasets are in the folder you have downloaded (mentioned in step 2) and are csv files. E.g. if you change line 60, to indicate project = 'tensorflow', then you will be obtaining the experiment results listed in the report for the tensorflow project. The screenshots below show what strings can be stored in the project variable

E.g. If you wish to run our tool using the 'caffe.csv' dataset, then please leave it as it is and proceed to step 4. Otherwise, please the string held in the project variable to the desired dataset you wish to run with our tool before proceeding to step 4.

```
61      path = f'{project}.csv'
```

Please ensure that the specified project csv file (i.e. dataset is in the same directory as the , **proposed_solution_svm.py** file before executing.

Step 4: To run our tool, please execute the following command in the same directory as the proposed_solution.py file and the csv file stated in line 61 :

```
python proposed_solution_svm.py
```

Note: try 'python3 proposed_solution_svm.py' if the above does not work.
Please ensure you have the python 3.12 installed and the specified libraries/dependences installed before running the tool

Step 5: The tool repeats the process of fitting an SVM model 10 times for robustness. For each run, we print out on the console the evaluation metric scores and optimal parameters . These values are stored in a csv file named, {project}_proposed_solution_metric_scores.csv, where project is the name of project specified in line 60. E.g. Below shows the evaluation metric scores on the 5th iteration whilst using the caffe.csv dataset for training and testing:

```
Optimal Parameter Setting for Repeated Experiment Run 5: {'C': 10, 'gamma': 0.01, 'kernel': 'rbf'}
Repeated Experiment Run 5: Accuracy:      0.7759
Repeated Experiment Run 5: Precision:      0.6101
Repeated Experiment Run 5: Recall:      0.7276
Repeated Experiment Run 5: F1_Score:    0.6221
Repeated Experiment Run 5: AUC:    0.7276
```

Please note that optimal parameter settings varied across the 10 runs due to the randomness of the splitting the dataset into a training and testing set. We have documented this in our report as figure 5 and can be verified using the values printed on the console.

Below shows the mean results across the 10 runs for each evaluation metric for our proposed solution. These solutions were documented in our report as figure 3,4,5,6,7 and were compared with the average value across 10 runs for each evaluation metric for the baseline solution. These values are also stored in **{project}_proposed_solution_metric_scores.csv**, where project name is the name of the dataset that has been specified in line 60 of the proposed solution, and either this file or values printed on the console can be used to reproduce and verify our results stated in the report.

```
=== SVM + TF-IDF Results ===
Number of repeats:     10
Average Accuracy:      0.8017
Average Precision:     0.5990
Average Recall:        0.6350
Average F1 score:      0.5968
Average AUC:           0.6350
```

In addition to this, in the console below where the average values for each evaluation metric for the proposed solution are printed, we also print statistical test results between the proposed solution and baseline solution. Below is a screenshot of what you would expect to see after executing the proposed

solution code using the caffe dataset. We compared the baseline precision values array with the proposed solution values array. This was repeated using Recall and F1 values from the baseline and proposed solution. These values were documented as figure 13 in our report and using the output printed on the console or the {project}_proposed_solution_metric_scores.csv, you can reproduce and verify the results.

```
=== Wilcoxon Rank Sum Test Results ===
Wilcoxon Rank Sum P-Value for Precision 0.19876460637323512
Wilcoxon Rank Sum P-Value for Recall 0.6501474440948545
Wilcoxon Rank Sum P-Value for F1-Score 0.010165201891956 26
```

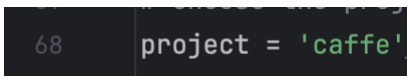
We also print out the standard deviation values in the console as shown below, these results can be help verify the results reported in the report in figures 3,4,5,6,7. The {project}_proposed_solution_metric_scores.csv also includes these values.

```
=== Standard Deviation Results ===
Standard Deviation for Accuracy:      0.0464
Standard Deviation for Precision:     0.0658
Standard Deviation for Recall:        0.0800
Standard Deviation for F1 score:      0.0655
Standard Deviation for AUC:           0.0800
```

Using the baseline code

We have included the baseline code, which is called, baseline_solution.py in our GitHub repository as we have made modifications to the one provided by the module team. This can be downloaded the same way you did for the proposed solution as it is in the same directory as the proposed solution and the relevant datasets. The raw results which include the values of the metrics across 10 runs, the averages of these metrics, and standard deviation are in the file named, {project}_baseline_metric_scores.csv, where project is the name of the project specified in line 68, which has been used to train an SVM model (please see step 3 below for more details). Please follow step 1 and step 2 as described in the 'Using the Proposed Solution Tool' section above.

Step 3: For the baseline solution named, baseline_solution.py at line 68, we have set a default dataset, 'caffe'. Please change this string to one of the following to obtain the reported experiment results on a specified dataset, the available datasets are in the folder you have downloaded and are csv files. The screenshots below show what strings can be stored in the project variable . E.g. if you change line 68, to indicate project = 'tensorflow', then you will be obtaining the experiment results listed in the report for the tensorflow



Please ensure that the specified project csv file (i.e. database) is in the same directory as the , **baseline_solution.py** file before executing.

```
69      path = f'{project}.csv'
```

Step 4: To run the baseline tool, please execute the following command in the same directory as the baseline solution.py file and the csv file stated in line 69 :

```
python baseline_solution.py
```

Note: try 'python3 baseline_solution.py' if the above does not work.

Step 5: The baseline tool repeats the process of fitting a Naïve Bayes model 10 times for robustness. For each run, we print out on the console the evaluation metric scores. E.g. Below shows the evaluation metric scores on the 5th iteration whilst using the caffe.csv dataset for training and testing the model: The array containing these values over the 10 runs are stored in a csv filed namely, **{project_name}_baseline_metric _scores.csv**, where project name is the name of the dataset that has been specified in line 68 of the baseline solution, as a reminder these files can also be found in the root directory of the GitHub repo. In addition to this, we have also stated these values in the report as well for further analysis – they can be found in figures 10,11,12,13,14 and be verified with the console output as shown below.

```
Repeated Experiment Run 5: Accuracy:      0.4655
Repeated Experiment Run 5: Precision:      0.5497
Repeated Experiment Run 5: Recall:      0.6282
Repeated Experiment Run 5: F1_Score:    0.4153
Repeated Experiment Run 5: AUC:    0.6282
```

Below shows the mean results across the 10 runs for each evaluation metric for our proposed solution. These solutions were documented in our report as figure 3,4,5,7 and were compared with the mean results across 10 runs for each evaluation metric for the proposed solution. The values in the report can be verified with the console output or the **{project_name}_baseline_metric _scores.csv**.

```
=== Naive Bayes + TF-IDF Results ===
Number of repeats:     10
Average Accuracy:      0.6000
Average Precision:     0.5608
Average Recall:        0.6532
Average F1 score:      0.4968
Average AUC:           0.6532
```

We also print out the standard deviation of the 10 values from the repeated experiment for each evaluation metric. This can then be verified with the figures shown in the report in figures 3,4,5,6,7 and with **{project_name}_baseline_metric _scores.csv**

```
=== Standard Deviation Results ===
Standard Deviation for Accuracy:      0.0854
Standard Deviation for Precision:     0.0346
Standard Deviation for Recall:        0.0868
Standard Deviation for F1 score:      0.0692
Standard Deviation for AUC:           0.0868
```