Software Defect Prediction:

Aidan Goodyer, Mason Azzopardi
{goodyera,azzoparm}@mcmaster.ca

1 Introduction

In software engineering static code metrics are used as a measure of 'quality' for a piece of code; think cyclomatic complexity, branch count, etc. However, investing time into hitting these metrics can be painstaking, and many of these metrics can feel like poor indicators of software correctness. Instead of trying have all these metrics hit a certain goal, why not use them as features for a binary classifier with labels contain whether or not the code has a defect. From here, instead of trying to look for the best possible classifier for the dataset, we can look for the static code metrics that actually matter. Through the application of L1 regularization, to favour sparsity, on a logistic regression classifier we hope to find which software metrics are the largest indicators of code defects.

2 Related Work

We are looking at two articles of related work one titled "Software Defect Prediction Using Support Vector Machine" (Alhija et al., 2022) and the other titled "Proposed Software Faults Detection Using Hybrid Approach" (Banga and Bansal, 2023). The first paper examines the software defect prediction performance and accuracy of a SVM over six differenct kernel funcion to find which kernel function is the best to use in this application however their results showed that no kernel function can give stable performance across different experimental settings. The second paper uses a hybrid algorithm proposed on particle swarm optimization and modified genetic algorithm for feature selection and bagging for effective classification of defective or nondefective modules in a dataset; the data being from a set NASA Metric Data Program datasets.

3 Dataset

You should write about your dataset here, following the guidelines regarding item 1. This section may be 0.5-1 pages. Depending on your specific dataset, you may want to include subsections for the preprocessing, annotation, etc.

4 Features

Describe any features you used for your model, or how your data was input to your model. Are you doing feature engineering or feature selection? Are you learning embeddings? Is it all part of one neural network? Refer to item 2. This may range from 0.25 pages to 0.5 pages.

5 Implementation

Describe your model and implementation here. Refer to item 4. This may take around a page.

6 Results and Evaluation

How are you evaluating your model? What results do you have so far? What are your baselines? Refer to item 5. This may take around 0.5 pages.

7 Feedback and Plans

Write about your plans for the remainder of the project. This should include a discussion of the feedback you received from your TA, and how you plan to improve your approach. Reflect on your implementation and areas for improvement. Refer to item 6. This may be around 0.5 pages.

8 Template Notes

You can remove this section or comment it out, as it only contains instructions for how to use this template. You may use subsections in your document as you find appropriate.

8.1 Tables and figures

See Table 1 for an example of a table and its caption. See Figure 1 for an example of a figure and its caption.

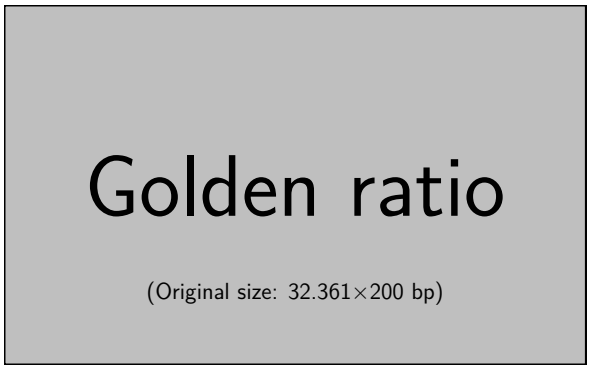

Figure 1: A figure with a caption that runs for more than one line. Example image is usually available through the `mwe` package without even mentioning it in the preamble.

8.2 Citations

Table 1 shows the syntax supported by the style files. We encourage you to use the natbib styles. You can use the command `\citet` (cite in text) to get "author (year)" citations, like this citation to a paper by Gusfield (1997). You can use the command `\citep` (cite in parentheses) to get "(author, year)" citations (Gusfield, 1997). You can use the command `\citealp` (alternative cite without parentheses) to get "author, year" citations, which is useful for using citations within parentheses (e.g. Gusfield, 1997).

8.3 References

Many websites where you can find academic papers also allow you to export a bib file for citation or bib formatted entry. Copy this into the `custom.bib` and you will be able to cite the paper in the LaTeX. You can remove the example entries.

8.4 Equations

An example equation is shown below:

$$A = \pi r^2 \tag{1}$$

Labels for equation numbers, sections, subsections, figures and tables are all defined with the `\label{label}` command and cross references to them are made with the `\ref{label}` command. This an example cross-reference to Equation 1. You can also write equations inline, like this: $A = \pi r^2$.

Team Contributions

Write in this section a few sentences describing the contributions of each team member. What did each member work on? Refer to item 7.

References

Haneen Abu Alhija, Mohammad Azzeh, and Fadi Almasalha. 2022. Software defect prediction using support vector machine. *Preprint*, arXiv:2209.14299.

Rie Kubota Ando and Tong Zhang. 2005. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 6:1817–1853.

Galen Andrew and Jianfeng Gao. 2007. Scalable training of L1-regularized log-linear models. In *Proceedings of the 24th International Conference on Machine Learning*, pages 33–40.

Manu Banga and Abhay Bansal. 2023. Proposed software faults detection using hybrid approach. *SECURITY AND PRIVACY*, 6(4):e103.

Dan Gusfield. 1997. *Algorithms on Strings, Trees and Sequences*. Cambridge University Press, Cambridge, UK.

Mohammad Sadegh Rasooli and Joel R. Tetreault. 2015. Yara parser: A fast and accurate dependency parser. *Computing Research Repository*, arXiv:1503.06733. Version 2.

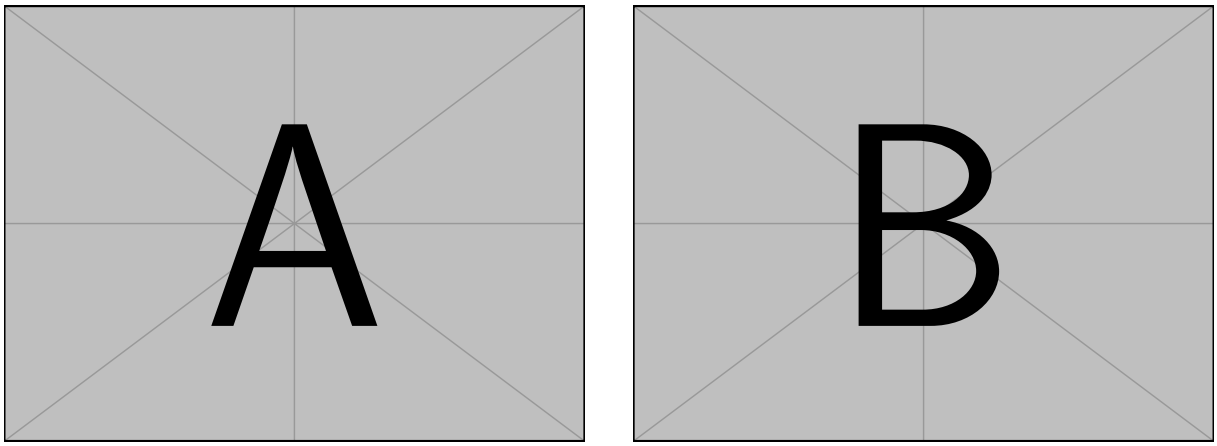

Figure 2: A minimal working example to demonstrate how to place two images side-by-side.

Output	natbib command	ACL only command
(Gusfield, 1997)	`\citep`	
Gusfield, 1997	`\citealp`	
Gusfield (1997)	`\citet`	
(1997)	`\citeyearpar`	
Gusfield's (1997)		`\citeposs`

Table 1: Citation commands supported by the style file.