Software Defect Prediction:

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

1 Introduction

In Software Engineering, static code metrics are often used as targets to define a 'quality' gate for a given module or piece of code. However, investing time into ensuring these metrics are met is painstaking, and many seem to be poor measures of the module's correctness.

A well-known critique of metric-driven evaluation comes from economist Charles Goodhart, whose law states that:

> "When a measure becomes a target, it ceases to be a good measure".

We seek to investigate the legitimacy of this statement as it pertains to static code metrics. Rather than aiming to create the best possible classifier for software defects, we instead aim to answer the question:

> "Which static software metrics *actually* matter"?

To investigate this question, we adopt an approach centered on feature selection and model interpretability. Specifically, we employ logistic regression in conjunction with L1 regularization to favour sparsity in the learned parameter, allowing us to derive an understanding of which metrics contribute meaningfully to defect prediction. By favouring an interpretable sparse model, we aim to provide analytical insight into the value of these metrics.

2 Related Work

Here, talk about the related work you encountered for your approach. Cite at least 5 references. Refer to item 2. No one has done exactly your task? Write about the most similar thing you can find. This should be around 0.25-0.5 pages.

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.

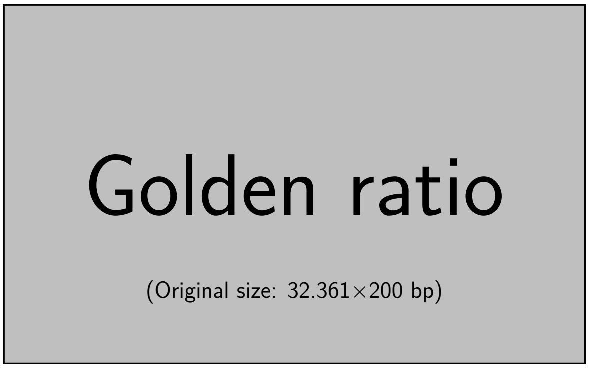

Figure 1: A figure with a caption that runs for more than one line. Example image is usually available through the `mwe` package without even mentioning it in the preamble.

8.1 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.

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

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.

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.

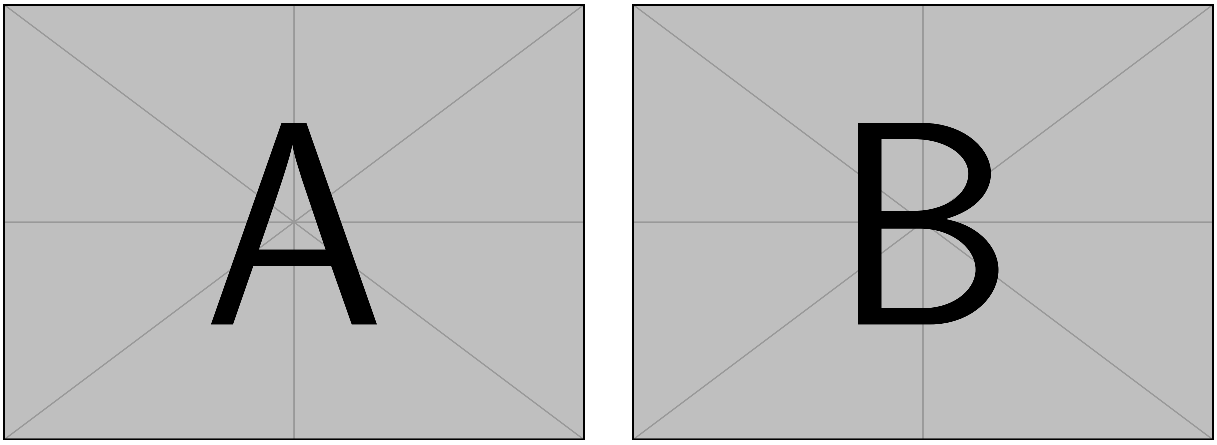

Figure 2: A minimal working example to demonstrate how to place two images side-by-side.

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.