Machine Learning Project
Task 1 Report Template

Philipp Liznerski
liznerski@cs.uni-kl.de

Tobias Michels
t_michels15@cs.uni-kl.de

Saurabh Varshneya
varshneya@cs.uni-kl.de

21st October 2021

1 HOW-TO

The source code in LaTex for this template is located in the file named "report.tex" in the project repository. This template aims at less experienced LaTeXusers and shows how to achieve some common tasks that might come up as you are writing your report. We don't require you to use the template; if you want, you can simply delete it and start your report from scratch or use your own template files.

1.1 Equations

We can easily typeset equations over multiple lines using the `align` environment, however this will add a number to the end of every single line. The `align*` environment on the other hand does not add numbers to any of the equations, but we can use the `\numberthis` command at the end of a line to give only that line a number:

$$
\begin{aligned}
2 + 2 &= 4 \\
4 - 1 &= 3 \qquad (1)
\end{aligned}
$$

Now we can reference the second line as eq. (1) using the `\cref` command. Note that the `cleveref` package automatically determined that the type of object we wanted to reference is an equation and it therefore added the corresponding text. Equation (1) can also be referenced at the beginning of a sentence if we use the `\Cref` command instead.

1.2 Ordered Lists

The `enumitem` packages provides great flexibility in handling ordered lists, specifically `enumerate` environments. Usually, items would be labelled by Arabic numbers and a period:

1. This

2. is

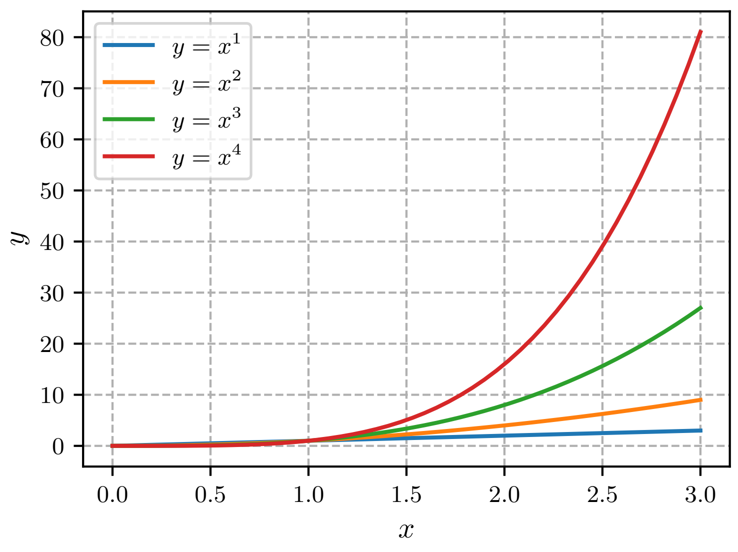

Figure 1: A figure that shows the relationship between y and x.

3. a test.

Perhaps you would like to have letters instead to replicate the task sheet and not quite as much empty space between the items? Not a problem:

a) This

b) is

c) a test.

Spacing can also be adjusted globally, using the command `\setlist{itemsep=<spacing>}`.

1.3 Figures & Code

At some point, you might want to include plots or other graphics in your report. Usually, this is done by defining a figure that does not necessarily appear at the point in the document where it was defined; see fig. 1 for example. The python code that generated the figure is also included with the report template, please have a look at the file `figures/make_figures.py`.

But what if you want to include and reference parts of your code in the report? In that case, you can use the `lstlisting` environment, which enables you to typeset code directly in LaTeX. The header file comes with an updated definition of the python language, which supports syntax highlighting for most parts of the language. Here is a small example:

```python
def count_ones(bits: int) -> int:
```

Listing 1: Source code that generated fig. 1.

```python
def make_example_plot(path, n=4, left=0, right=3, res=50):
    width = TEXT_WIDTH * TEXT_WIDTH_MUL
    fig = plt.figure(figsize=(width, width*ASPECT))
    ax = plt.gca()
    ax.set_xlabel('$x$')
    ax.set_ylabel('$y$')
    ax.grid(linestyle='dashed')

    x = np.linspace(left, right, num=res)
    for i in range(1, n+1):
        y = x ** i
        ax.plot(x, y, label=f'$y = x^{i}$')

    ax.legend()
    fig.tight_layout()
    plt.savefig(os.path.join(path, 'example_plot.pdf'))
    plt.close(fig)
```

```python
    # count number of ones in the binary representation
    # of the signed integer bits
    c = 0
    v = bits
    while v != 0:
        c += 1
        v &= v - 1  # This clears the least significant 1 bit

    return c
```

Instead of writing the code directly in the LaTeXfile, you can also “import” it from the original source file with the `\lstinputlisting` command. For example, listing 1 shows an excerpt from the source code file `figures/make_figures.py` that was used to generate fig. 1.

Note how the code about counting bits was split up between two separate pages. This is expected behaviour, since the normal text mode works like a paragraph of text, which can obviously be split. If you don’t want that behaviour, use the optional `float` parameter that turns the listing into a floating object, similar to a figure. For example, listing 1 uses this parameter and is not split between two pages, but at the same time the source code does not appear in the exact position where it was included in the LaTeXfile.

1.4 TODO Notes

TODO: Remove this TODO note

Notes and reminders like the one above can by typeset with the `\todo[inline]{<text>}` command. This is especially helpful if you are collaborating with others, since you can

Mention that notes can also be added next to the text area use this to remind everyone of work that still needs to be done.

1.5 Citing References

If you want to use material created by someone else in your report, it is mandatory that you cite it properly. This is even more important if you want to refer to somebody else's ideas. Luckily, LaTeXand the internet can help you with that, making your life a lot easier in that regard. Ideally, you would only want to cite papers that were published either in a peer-reviewed journal or a conference's proceedings. For example, you might have read the OpenAI blog post about *adversarial examples*[1] and you want to pick up some of the ideas presented there. Since the blog post lists the relevant papers, you can search for them in an engine like *Google Scholar* that lets you export citations as *BibTeX* entries. After pasting the entry in your references file, you can cite them in your text by simply using the `\autocite` command: [1].

Of course, finding a peer-reviewed paper for the ideas in a blog post might not always be possible. In such a case, you can also cite the blog post directly using an `@online` entry in the bib file [2].

1.6 Tables

The following webpage is an excellent resource to learn to make well-formatted tables in LaTeX using the *booktabs* package `https://jdhao.github.io/2019/08/27/latex_table_with_booktabs/`.

References

[1] Nicolas Papernot et al. "Practical Black-Box Attacks against Machine Learning". In: *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*. ASIA CCS '17. Abu Dhabi, United Arab Emirates: Association for Computing Machinery, 2017, pp. 506–519. ISBN: 9781450349444. DOI: `10.1145/3052973.3053009`.

[2] Ian Goodfellow et al. *Attacking Machine Learning with Adversarial Examples*. 24th Feb. 2017. URL: `https://openai.com/blog/adversarial-example-research/` (visited on 02/11/2020).

[1]`https://openai.com/blog/adversarial-example-research/`