

## Reflection report on the use of large language models

We used chat-GPT in a couple of different ways to help us better understand certain elements of the course and help improve answers and coding to the problems given in the assignment.

Firstly, we used it as a sort of second lecturer, where we would ask questions about the theory of the course and what certain variables really mean. Here, an example of these sort of questions:

*'Why is it logical that you do not regularize your intercept with regression'*

This leads to a detailed explanation of how regularization actually works and what the function of the intercept is. Using chat-GPT as a way to repeat certain explanations or just letting it elaborate on subjects we forgot or did not really understand from lectures seems very helpful to us. This definitely made us more productive as we did not need to read very difficult-to-understand Wikipedia pages about every term used in slides and felt like it was a tutor on demand. Of course, to ensure that the answer provided by chat-GPT is right we would still look at the slides or Google terms. Another related way we would use chat-GPT is by letting it explain questions to us. So what exactly do all the terms mean in the question and what would be a general strategy in tackling such problems? I found that it generally was able to answer these questions very well where we did not need any follow-up questions to understand how to continue. For example:

*'Show the effect of the regularization on the parameter estimates in terms of bias and variance. For this you can repeat the optimization 100 times using bootstrap and visualise the profile of the Lasso regression coefficient over a grid of the hyperparameter, optionally including the variability as error bars. What does this mean? No code please'.*

Secondly, we used it to help us with coding and bug fixes. Here we, however, encountered quite a few problems with regard to using chat-GPT. Often when asked to refactor code, it would leave other elements out or change something very slightly in the code where it suddenly was not doing what chat-GPT says it does. It can be a very convincing bullshitter in this way where it can go around in a loop introducing very strange ways of coding things. This is why we would often for an exercise lay a framework ourselves and then let it help us with identifying errors, plotting results etc. For instance, we would ask it what certain lines of code mean:

*'mean\_scores = -grid\_search.cv\_results\_['mean\_test\_score']*

*What does this line do?'*

*'GridSearchCV , what does this do?'*

This would help us understand how to use certain sklearn elements, which we sometimes found hard to understand while reading documentation. Productivity wise we think that using chat-GPT like this can be very beneficial but also very harmful. If you do not fully understand the changes chat-GPT makes to your code it can quickly lead to undesired behaviour, where you have to put a lot of time and effort into reverting back to your original plan/code. Asking it to explain code however and writing small changes and inspecting every line it generates to understand what it is doing can be very beneficial.

In conclusion, we think using models like these is very beneficial only if the user truly understands the answers it gives. If you are simply copying and pasting it brings great risks where it will fool you.