# CSC413 Final Project Rubric

The project in this course is an opportunity to develop a sequence-based deep learning application in an area of your own choosing. It also provides the chance to complete a deep learning project that is much closer to a real-world application.

The project github repository is due at the end of term April 7, 10pm.

The repository should be either public, or privately viewable to the instructors and TAs. We recommend a public repository to showcase the work that you are able to do!

## Submission

Please submit a file called `github.txt` containing a link to the github repository. If your repository will be private, please email the instructors by April 7, 10pm so that TAs and instructors can be added—even if you use tokens.

## Repository Content

The repository should contain:

(1) **The code you used to pre-process the data, but not the data itself**. It is generally a bad idea to include data in your github repository, since git is great for lots of small files, but a poor choice for sharing large files. Moreover, most groups are using data collected by other people. While you should share the source of your data, you should generally not share a copy of the data.

(2) **The code you used to train your model**. You *may* opt to share model weights, or not.

(3) **A README file with the following component**:

- Introduction that states the deep learning model that you are building
- Model:
  - A figure/diagram of the model architecture that demonstrates understanding of the steps involved in computing the forward pass
  - Count the number of parameters in the model, and a description of where the parameters come from
  - Examples of how the model performs on two actual examples from the test set: one successful and one unsuccessful
- Data:
  - Describe the source of your data
  - Provide summary statistics of your data to help interpret your results (similar to in the proposal)
  - Describe how you transformed the data (e.g. any data augmentation techniques)
  - If appropriate to your project, describe how the train/validation/test set was split. (Note that splitting the training/validation/test set is not always straightforward!)
- Training:
  - The training curve of your final model
  - A description how you tuned hyper-parameters
- Results:
  - Describe the quantitative measure that you are using to evaluate your result
  - Describe the quantitative and qualitative results
  - A justification that your implemented method performed reasonably, given the difficulty of the problem—or a hypothesis for why it doesn't (this is extremely important)
- Ethical Consideration:
  - Description of a use of the system that could give rise to ethical issues. Are there limitations of your model? Your training data?
- Authors
  - A description of how the work was split—i.e. who did what in this project.

## Marking Scheme

Here is the marking scheme that we will use. Note that you model must be able to make reasonable predictions for your project to receive a passing project grade. In particular, without a reasonable model, you won't be able to earn credit for Model Examples, Training Curve, Hyperparameter Tuning, Qualitative/Quantitative Results, etc.

**README/Writeup (70 points)**

- **Introduction (4 points)**: What deep learning model are you building? We are looking for a clear and concise description that uses standard deep learning terminology. Clearly describe the type of task that you are solving, and what your input/outputs are.
- **Model Figure (4 points)**: A figure/diagram of the model architecture that demonstrates understanding of the steps involved in computing the forward pass. We are looking to see if you understand the steps involved in the model computation (i.e. are you treating the model as a black box or do you understand what it's doing?)
- **Model Parameters(4 points)**: Count the number of parameters in the model, and a description of where the parameters come from. Again, we are looking to see if you understand what the model is doing, and what parameters are being tuned.
- **Model Examples (4 points)**: Examples of how the model performs on two actual examples from the test set: one successful and one unsuccessful.
- **Data Source (1 point)**: Describe the source of your data.
- **Data Summary (4 points)**: Provide summary statistics of your data to help interpret your results, similar to in the proposal. Please review the feedback provided in the proposal for some guidance on what information is helpful for interpreting your model behaviour.
- **Data Transformation (3 points)**: Describe how you transformed the data, i.e. the steps you took to turn the data from what you downloaded, to something that a neural network can use as input. We are looking for a concise description that has just enough information for another person to replicate your process.
- **Data Split (2 points)**: If appropriate to your project, describe how the train/validation/test set was split. Note that splitting strategy is not always straightforward, so we are looking to see a split that can be justified.
- **Training Curve (4 points)**: The training curve of your final model. We are looking for a curve that shows both training and validation performance (if applicable). Your training curve should look reasonable for the problem that you are solving.
- **Hyperparamter Tuning (4 points)**: A description how you tuned hyper-parameters. We are looking for hyperparameter choices that makes sense.
- **Quantitative Measures (2 points)**: A description and justification of the quantitative measure that you are using to evaluate your results. For some problems this will be straightforward. For others, please justify the measure that you chose.
- **Quantitative and Qualitative Results (8 points)**: Describe the quantitative and qualitative results. You may choose to use a table or figure to aid in your description. We are looking for both a clear presentation, and a result that makes sense given your data summary. (As an extreme example, you should not have a result that performs worse than a model that, say, predicts the most common class.)
- **Justification of Results (20 points)**: A justification that your implemented method performed reasonably, given the difficulty of the problem—or a hypothesis for why it doesn't. This is extremely important. We are looking for an interpretation of the result. You may want to refer to your data summary and hyperparameter choices to make your argument.
- **Ethical Consideration (4 points)**: Description of a use of the system that could give rise to ethical issues. Are there limitations of your model? Your training data? Please review the feedback provided in the proposal for some guidance on how to think deeply about these issues.
- **Authors (2 points)**: A description of how the work was split—i.e. who did what in this project. If there are significant issues with the way that work is split, we may follow up with individual teams, and not award equal points to all team members.

**Code/Documentation (20 points)** We are looking for whether TAs can generally understand what your code does, how it is organized, and the steps that needs to be taken to replicate your model and results. Your code must be in working order (otherwise the TA will not be able to replicate your results)

**Advanced Concept (10 points)**. Your project involves at least one of the following:

- Data Augmentation *applied in a way that makes sense for your domain*
- Transformer
- Generative Model, Sequence-to-Sequence Architecture (e.g. that uses teacher-forcing)