# ML project review checklist **MUST-ASK QUESTIONS IN BOLD**

## High-level questions about the project

* **What question were you trying to answer? How did you frame it as an ML task?**
* What is human-level performance on that task? What level of performance is needed?
* Is it possible to approach this problem without machine learning?
* If the analysis focused on deep learning methods, did you try shallow learning methods?
* What are the ethical and legal aspects of this project?
* **Which domain experts were involved in this analysis?**
* Which data scientists were involved in this analysis?
* Which tools or framework did you use? (How much of a known quantity is it?)
* Where is the pipeline published? (E.g. public or internal git repositories.)
* How thorough is the testing and documentation?

## Questions about the data preparation

* Where did the feature data come from?
* Where did the labels come from?
* What kind of data exploration did you do?
* How did you clean the data? How long did this take?
* **Are the classes balanced? How did the distribution change your workflow?**
* What kind of normalization or standardization did you do?
* What did you do about missing data? E.g. what kind of imputation did you do?
* What kind of data augmentation (new rows) or feature engineering (columns) did you do?
* How did you split the data into train, validate and test? Or did you use some form of CV?

## Questions about training and evaluation

* Which models did you explore and why? Did you also try the simplest models that fit the problem?
* How did you tune the hyperparameters of the model? Did you try grid search or other methods?
* **What kind of validation did you do? Did you use cross-validation? How did you choose the folds?**
* What evaluation metric are you using? Why is it the most appropriate one?
* **How do training, validation, and test metrics compare?**
* If this was a classification task, how does a dummy classifier score?
* How are errors/residuals distributed? (Ideally normally distributed and homoscedastic.)
* How interpretable is your model? That is, do the learned parameters mean anything, and can we learn from them? E.g. what is the feature importance?
* If this was a classification task, are probabilities available in your model and did you use them?
* If this was a regression task, have you checked the residuals for normality and homoscedasticity?
* Are there benchmarks for this task, and how well does your model do on them?

## Next steps for the project

* How will you improve the model?
* Would collecting more data help? Can we address the imbalance with more data?
* Are there human or computing resources you need access to?
* How will you deploy the model?