Checklist for the final project

While this is not exhaustive or mandatory, being able to answer most of these questions is a good indication of the maturity of the final project.

1. Dataset and the problem at hand
    a. Are you sure the problem is suitable to be solved through machine learning? Do you understand the use case well?
    b. Did you find an appropriate, public dataset for your project? Is the source trustworthy?
2. EDA:
    a. Did you spend time exploring the dataset, collecting relevant statistics and visualizing distributions of interest?
    b. What features did you identify as most promising for your model?
3. Data preparation / QA:
    a. Did you use your domain knowledge to filter out or flag suspicious data points?
    b. Did you address data-imbalance problems or perform data augmentation, as appropriate?
4. Code structure:
    a. Is your code well-commented and did you write a README to introduce the project?
    b. Are dependencies specified and can the code run through a clean install from a virtual environment?
    c. Are logical components isolated in Metaflow steps and important artifacts saved and versioned?
5. Training and optimization:
    a. Did you split your data in train/test without leakage?
    b. Did you pick a suitable model for the problem at hand?
    c. Did you use a validation procedure to tune hyperparameters? [If you have doubts with Metaflow you can either just use one hyper and a single foreach with some values for it, or do fancier stuff, like they do here.]
6. Tracking:
    a. Did you track your experiments with Comet?
7. Testing:
    a. Did you pick suitable metrics for your problem?
    b. Did you create qualitative checks on specific test cases to make sure the model behavior aligns with your goals?
    c. Did you check if some interesting data slice is performing better / worse than the average?
    d. Did you check for robustness, if appropriate?
8. Deployment:
    a. Basics: can you spin up a web application on your machine and show a local endpoint returning a prediction, given the inputs (i.e.

localhost:5000/predict?x=1442.0)? Did you structure the JSON response in your endpoint in a clear way, separating data and metadata?

b. BONUS: can your model be reached from any computer to produce a prediction, leveraging AWS?