Machine Learning Project Checklist

The items on this checklist come from various sources, such as Machine Learning Yearning, Full Stack Deep Learning, Building Machine Learning Powered Applications, and also from my own personal experience. This is work in progress, and contributions are welcome. If you have any additions, please submit a PR to this repo.

Before modelling

Project

- [ ] The project has a clear, codified business goal/metric.
- [ ] There is a person who is ultimately responsible for the success/failure of the project.
- [ ] We have a plan for how to reach a first deployed end product as fast as possible.
- [ ] We have decided on how and when to keep the team in sync (daily/weekly standups, retrospectives, planning meetings, etc)
- [ ] We have assessed how the product will impact stakeholders (e.g. people, society, world)
- [ ] We have identified relevant regulation and translated it to requirements
- [ ] We have identified requirements related to Fairness, Accountability, Transparency

Problem understanding

- [ ] We have decided on one single metric on which to rank my models.
- [ ] We have clarified the costs of the different kinds of erroneous predictions.
- [ ] We have an understanding of how good performance is "good enough"
- [ ] We know the constraints in serving time w.r.t. memory usage.
- [ ] We know the constraints in serving time w.r.t. latency.
- [ ] We know the constraints in serving time w.r.t. throughput.
- [ ] We know if we’re doing streaming- or batch prediction.
- [ ] We understand the current state of ML applied to the problem we’re trying to solve.
- [ ] We have an idea of how important freshness is. How often will we need to change the

model?

☐ We have domain experts who can help us understand the problem and error modes.

☐ We know where the model will be deployed (server / client, browser / on device)

Data

☐ I have selected a dev- and test set that are reflective of the real task I'm trying to solve.

☐ My dev- and test sets are from the same distribution.

☐ My dev set is large enough, so that I can detect improvements to the desired accuracy.

☐ We understand how to split the data into train/val/test to avoid data leakage.

☐ If we need to collect data, we know how difficult and costly it will be to collect and annotate.

☐ We have a plan for how to store and version our data, dataset splits, models, and change in annotations.

☐ I get a reasonable "ML Test Score", table 1.

Modelling

☐ I have one or several well thought out baselines in place. These are not good enough, so there's an actual need to use ML.

☐ There's a metrics webpage where I can compare runs and the url is ____________________.

☐ We can (approximately) reproduce a model if needed.

☐ I get a reasonable "ML Test Score", table 2.

Deployment

☐ We have CI in place.

☐ We have tests for the full training pipeline.

☐ We have validation tests.

☐ We have functionality tests.

- [ ] We have unit tests.
- [ ] We have CD in place.
- [ ] We have CT in place.
- [ ] Blue/green deployment in place.
- [ ] We can deploy a model in shadow mode.
- [ ] Monitoring in place for memory consumption.
- [ ] Monitoring in place for CPU consumption.
- [ ] Monitoring in place for latency.
- [ ] Monitoring in place for downtime.
- [ ] Monitoring in place for requests per second.
- [ ] Monitoring in place for prediction confidence over time.
- [ ] We have a way of detecting if a model will fail on a given datapoint, and a corresponding fallback.
- [ ] I get a reasonable “ML Test Score”, table 3-4.