1. What are 3 advantages of deploying using Model Serving Vs. deploying on GH Pages or HF for free?

Deploying a model using a server has the following advantages:

* Security – Model serving frameworks provide security like different kinds of authentication (AWS asks users to download Google apps that help with authentication).
* Each Model Serving has its own ways of providing libraries, resources (GPUs etc.), and other infrastructure requirements. Thus, it is possible to fine tune one’s requirements according to what the Model Server allows.
* AWS, Google Cloud, Azure etc., allow for scalability for deployment.

Deploying a model using a GitHub pages or HF has the following advantages:

* We used HF to deploy our protein language model and it was much easier to do to than the AWS set-up. It was free as well. AWS costs money. Although we didn’t pay any $ during our HW, it’s clear the cost of is in thousands of dollars when ML models go into production. For small ML model deployments, where scaling is not an issue but cost is, HF is probably better.

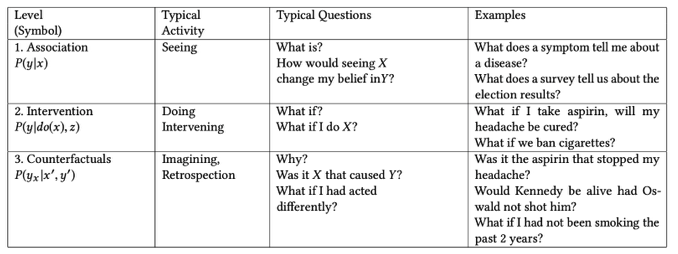
1. What is ML model deployment?

* ML model deployment means we make an ML model operational. The ML model that has been trained and tested is put into a formal production environment. It means we make available our ML model (and its endpoint) to outside users where it is used the way it was meant to be used.
* For example, if an ML model has been created to detect fraud in credit card transactions, it must work in real time so it can actually prevent fraud. Therefore the ML model must be deployed in the real world or what is called a ‘production’ environment so it can be used for its true purpose.
* The server on which we deploy our ML model can be AWS or GC or Azure or other sites like HF. Before deployment, there are requirements that must be satisfied. For example, there should be [1] processing/cleaning data [2] having a good infrastructure that can be scaled to high volumes of data [3] model must be packaged in a container. And once deployed, the ML models must be monitored and maintained.

1. What is Causal Inference and How Does It Work?

* ML model serving is to do “inference”. When we train the ML model and test it, we are doing inference albeit in a safe, ‘testing’ environment. Once the ML model is working well, we deploy the model into the real world and let it go ‘live’. Now real, new data points are inputted into the model and the model makes predictions and this called machine learning inference.
* Causal inference is however, a whole new can of worms. Inferring causation is not simple. One of the best summaries I have seen on this is given by Pearl et al. 2016. This is referred to as the three levels of causal hierarchy. The first level is association, the second is intervention, and the third is the counterfactual. The reason this table is important to me is because of how it lays out the three increasing levels of hierarchy in inferring causality. In Level 1 (i.e., association) the examples make it clear that association is about observation and relationships defined by data on the superficial level. The next Level 2 (intervention) begins to get at causation by asking the question – if I take an aspirin, will the tablet fix my headache? Level 3 (counterfactual) is the most complex of all. This asks whether it was the aspirin that fixed the headache (aspirin caused it to go away and nothing else) or what it something else (like fresh air or a nap?). This is getting at “cause”.

Inferring causality in other words, is complicated. Association is definitely not causation. Doing an intervention (for e.g., a randomized clinical trials where we test experimental drugs against a gold standard) one can infer causation as randomization takes care of confounders, but a randomized clinical trial is expensive and difficult to do, therefore causal inference is not easy to do/answer.



* Now in an ML set-up: In a ML task, we do prediction. When we do prediction, we use a ML model that is trained on a dataset is asked to predict on new data it has NOT seen before. However, it is a lot more difficult to do good causal inference when the data is new. Consider the following –
  + Imagine we train a model on predicting whether or not a person has heart-disease based on certain symptoms in US, but then we want to use this same model in Japan. Now, rate of heart disease in Japan is different than the US. The probability of having heart disease (what is called the marginal probability) is lower in Japan thus, we will need to fix our ML model, because the fraction of people with heart disease is much, much lower.
  + Imagine the same problem, but with an added complication. In Japan, people don’t have the same symptoms as people in US when they have heart disease. In US, people may be more prone to chest pain and breathlessness, while in Japan, people may have no chest pain or breathlessness but numbness in the left side. If the symptoms of heart disease are different in Japan than in US, the ML models must be revised to incorporate them.

1. What is server-less deployment and how its compared with deployment on server?

* Serverless deployment means that we used a third-party service to deploy our ML model. The cloud provider we choose to host and deploy our model, which provides us with a runtime environment, whether it is Amazon Web Services or Microsoft Azure or Google Cloud is in charge of assigning computing resources for our code to run. Each of the above organizations has a different method of organizing and implementing deployment, libraries, resourcing availability.
* In a server, one pays the cost for the server. In a serverless deployment, one only pays for the time that the application is running. In a server, the OS etc. are updated and maintained by the providers. For our own server, we have to do the above. Although one can customize a server to one’s needs, it seems to be an easier task to use a cloud than use a server.

References:

Judea Pearl, Madelyn Glymour, Nicholas P. Jewell. Causal Inference in Statistics – A Primer. Wiley, 2016.