ML Practical Design Interviews at Facebook

The ML Practical Design interview focuses on your ability to building ML systems at Facebook scale. A strong performance in this interview indicates that would be successful in most applied ML problems here at Facebook.

What we ask

Some sample questions we ask are:

* Design newsfeed ranking
* Design local search ranking
* Design evaluation framework for ads ranking

The idea is to pick any product feature and understand how to model it using machine learning. We are not looking for you to know and memorize every ML algorithm out there. You should be able to use existing toolsets to model the problem and breakdown the components involved in building a large-scale ML system.

Expectations

What we’re looking for:

* Can you visualize the entire problem and solution space?
* Are you good at feature engineering?
* Can you detect flaws in machine learning systems and suggest improvements?
* Can you design consistent evaluation and deployment techniques?
* Do you understand architecture requirements (storage, perf etc.) of your system?
* Can you model product requirements into your ML system?

A good design will touch on the following different components:

* Problem formulation
  + Optimization function
  + Supervision signal
* Feature engineering
  + Data source
  + Representation
* Model architecture
* Evaluation metrics
* Deployment (A/B testing)

How to study

To practice, take any well-known app and pick a system that can benefit from machine learning. Consider that system is built using a handful of rules for a small set of people. Now, consider you want to deprecate those rules and want to take advantage of machine learning, so you can easily extend that functionality to millions of people.  
  
Brush up basic ML theory and be comfortable with concepts like overfitting and regularization.  
  
Practice ability to convert intuitive ideas to concrete features. For example: number of likes is a good idea but a better feature would use involves normalization, smoothing and bucketing.   
  
Think about the problem end to end. What will you do after you train the model and the model does not perform well? How do you go about debugging an ML model? How do you evaluate and continuously deploy an ML model?  
  
Be ready to analyze your approach. Having a good toolset of several different algorithms and understanding the tradeoffs is helpful. For example, be able to example advantages of logistic regression compared to SVM.  
  
Work out the above problems on a paper and just think about the ways to break them down. It also helps to read up on common large-scale ML systems. Watch the public videos and learn how google search ranking works.