BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Test In Production

Pravin Y Pawar

Adapted from “Designing Machine Learning Systems”
By Chip Huyen

Offline Evaluation

- An offline model evaluation happens when the model is being trained by the analyst

- The analyst tries out different
  - features,
  - models,
  - algorithms,
  - and hyperparameters

- Model training is guided in the right direction by
  - Tools like confusion matrix
  - Various performance metrics, such as precision, recall, and AUC

- Process
  - First, validation data is used to assess the chosen performance metric and compare models
  - Once the best model is identified, the test set is used, also in offline mode, to again assess the best model's performance
  - This final offline assessment guarantees post-deployment model performance

Offline Evaluation(2)

Two major test types for offline evaluation

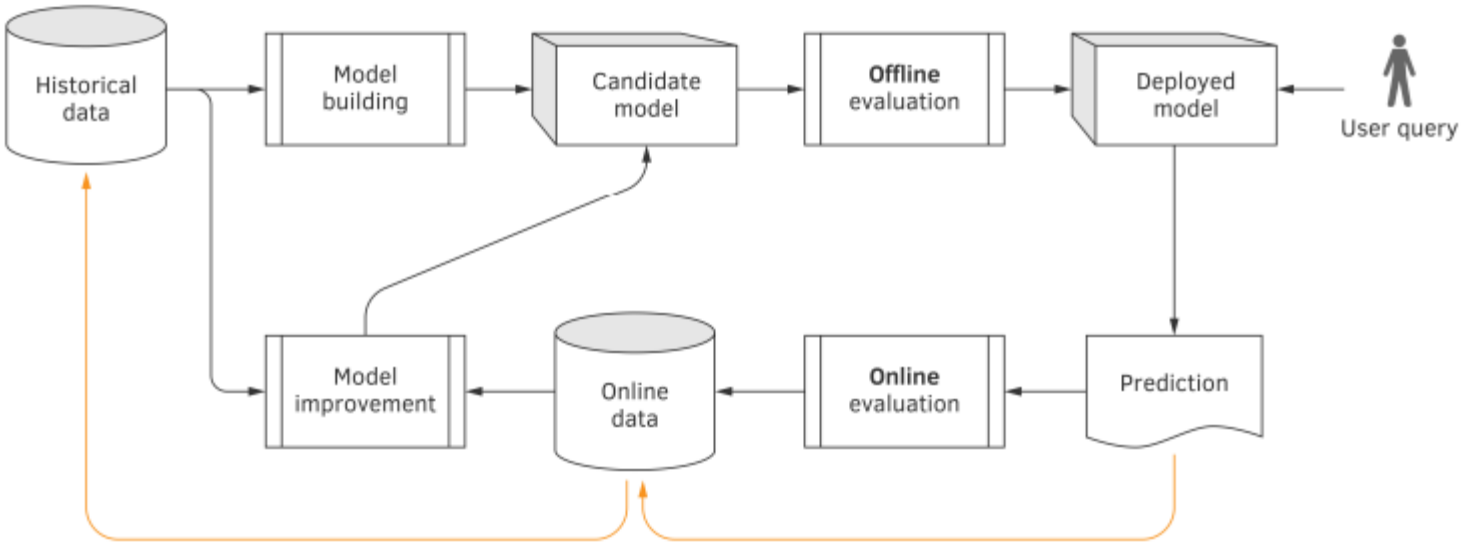
- Test splits
  - are usually static and have to be static so that you have a trusted benchmark to compare multiple models
  - will be hard to compare the test results of two models if they are tested on different test sets

- Backtests
  - method of testing a predictive model on data from a specific period of time in the past
  - If model is updated to adapt to a new data distribution, it's not sufficient to evaluate this new model on test splits from the old distribution
  - one idea is to test model on the most recent data that you have access to - after updated model on the data from the last day, might want to test this model on the data from the last hour

- The question is whether backtests are sufficient to replace static test splits. Not quite!
  - If something went wrong with data pipeline and some data from the last hour is corrupted, evaluating model solely on this recent data isn't sufficient.
  - With backtests, should still evaluate model on a static test set that have been extensively studied and (mostly) trust as a form of sanity check.

Offline and Online Evaluation

- Historical data is first used to train a deployment candidate
  - Then it is evaluated offline
  - If the result is satisfactory, deployment candidate becomes the deployed model, and starts accepting user queries
- User queries and the model predictions are used for an online evaluation of the model
  - The online data is then used to improve the model
  - To close the loop, the online data is permanently copied to the offline data repository



The placement of offline and online model evaluations in a machine learning system.

Source: Machine Learning Engineering by Burkov

Why do we evaluate both offline and online?

- The offline model evaluation reflects how well the analyst succeeded
  - in finding the right features, learning algorithm, model, and values of hyperparameters
  - reflects how good the model is from an engineering standpoint

- Online evaluation, focuses on measuring business outcomes,
  - such as customer satisfaction, average online time, open rate, and click-through rate
  - may not be reflected in historical data, but it's what the business really cares about

- Offline evaluation doesn't allow us to test the model in some conditions that can be observed online,
  - such as connection and data loss, and call delays

Test in production

Aka Online Evaluation

- The only way to know whether a model will do well in production is to deploy it
  - led to one seemingly terrifying but necessary concept

- Techniques to help to evaluate models in production (mostly) safely
  - Shadow deployment
  - A/B testing
  - Canary analysis
  - Interleaving experiments
  - Bandits

- To sufficiently evaluate models, first need a mixture of offline evaluation and online evaluation!

Shadow Deployment

Might be the safest way to deploy model or any software update

- Works as follows
  1. Deploy the candidate model in parallel with the existing model
  2. For each incoming request, route it to both models to make predictions,
     but only serve the existing model's prediction to the user
  3. Log the predictions from the new model for analysis purposes.

- Only when found that the new model's predictions are satisfactory do replace the existing model with the new model!

- The risk of this new model doing something funky is low
  - as don't serve the new model's predictions to users until made sure that the model's predictions are satisfactory

- Always not favorable because it's expensive
  - doubles the number of predictions system has to generate, which generally means doubling inference compute cost

A/B Testing

- A way to compare two variants of an object, typically by testing responses to these two variants, determining which of the two variants is more effective
  - the existing model as one variant, and the candidate model (the recently updated model) as another variant

- Works as follows:
  1. Deploy the candidate model alongside the existing model
  2. A percentage of traffic is routed to the new model for predictions; the rest is routed to the existing model for predictions.
  3. Monitor and analyze the predictions and user feedback, if any, from both models to determine whether the difference in the two models' performance is statistically significant.

- Two important things:
  - First, A/B testing consists of a randomized experiment: the traffic routed to each model has to be truly random.
    - If not, the test result will be invalid
  - Second, A/B test should be run on a sufficient number of samples to gain enough confidence about the outcome.
    - How to calculate the number of samples needed for an A/B test is a simple question with a very complicated answer.

- Often, in production, you don't have just one candidate but multiple candidate models
  - possible to do A/B testing with more than two variants, which means can have A/B/C testing or even A/B/C/D testing

Canary Release

- A technique to reduce the risk of introducing a new software version in production by slowly rolling out the change to a small subset of users
  - before rolling it out to the entire infrastructure and making it available to everybody

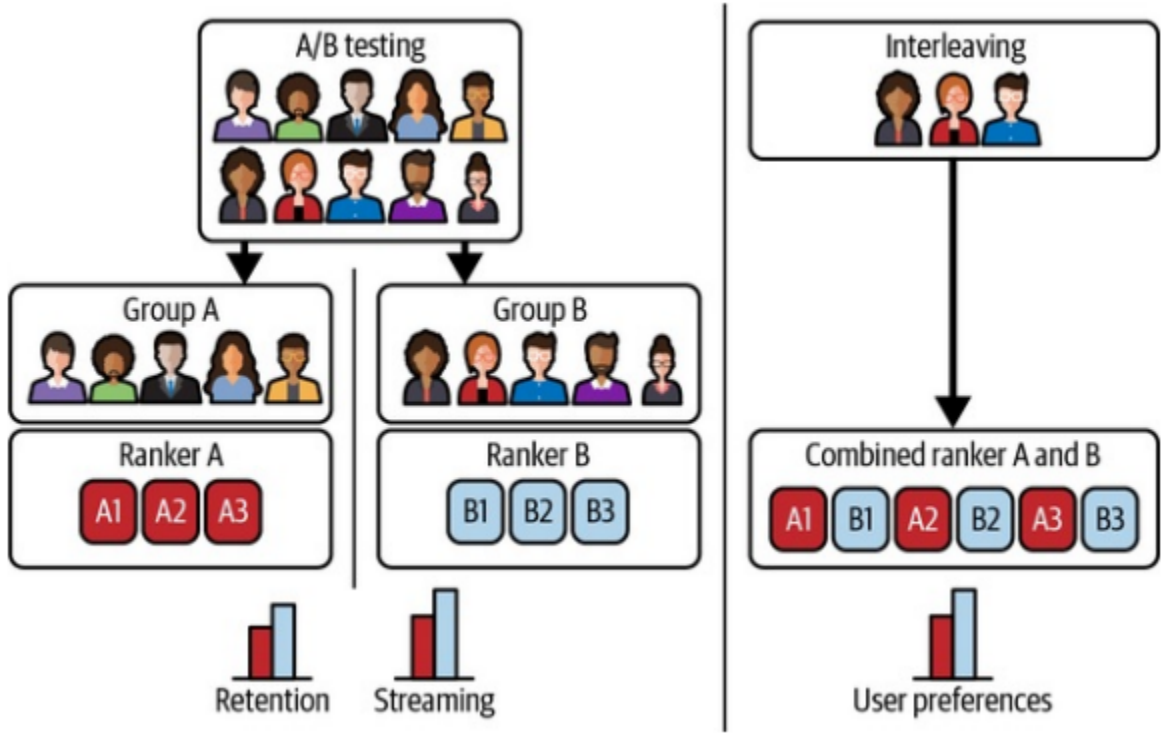
- Works as follows:
  1. Deploy the candidate model alongside the existing model. The candidate model is called the canary
  2. A portion of the traffic is routed to the candidate model.
  3. If its performance is satisfactory, increase the traffic to the candidate model.
     If not, abort the canary and route all the traffic back to the existing model.
  1. Stop when either the canary serves all the traffic (the candidate model has replaced the existing model) or when the canary is aborted

- The candidate model's performance is measured against the existing model's performance according to the metrics you care about
  - If the candidate model's key metrics degrade significantly, the canary is aborted and all the traffic will be routed to the existing model

- Canary releases can be used to implement A/B testing due to the similarities in their setups
  - don't have to randomize the traffic to route to each model
  - first roll out the candidate model to a less critical market before rolling out to everybody

Interleaving Experiments

- Imagine two recommender systems, A and B, and want to evaluate which one is better
  - Each time, a model recommends 10 items users might like
- With A/B testing, divide users into two groups: one group is exposed to A and the other group is exposed to B.
  - Each user will be exposed to the recommendations made by one model
- What if instead of exposing a user to recommendations from a model, we expose that user to recommendations from both models and see which model's recommendations they will click on?



An illustration of interleaving versus A/B testing. Source: Adapted from an image by Parks

Interleaving Experiments(2)

- In A/B testing, core metrics like retention and streaming are measured and compared between two groups
- In interleaving, two algorithms can be compared by measuring user preferences,
  - no guarantee that user preference will lead to better core metrics

- Netflix found that interleaving “reliably identifies the best algorithms with considerably smaller sample size compared to traditional A/B testing.”

- When recommendations from multiple models are shown to users,
  - the position of a recommendation influences how likely a user will click on it
  - users are much more likely to click on the top recommendation than the bottom recommendation

- For interleaving to yield valid results, must ensure that at any given position,
  - a recommendation is equally likely to be generated by A or B.

Bandits

- Bandit algorithms originated in gambling!

- A slot machine is also known as a one-armed bandit
  - A casino has multiple slot machines with different payouts
  - don’t know which slot machine gives the highest payout
  - Can experiment over time to find out which slot machine is the best while maximizing payout

- Multi-armed bandits are algorithms
  - allow you to balance between
    - exploitation (choosing the slot machine that has paid the most in the past)
    - and exploration (choosing other slot machines that may pay off even more)

- When multiple models needs to be evaluated, each model can be considered a slot machine whose payout (i.e., prediction accuracy) is unknown
- Bandits allow to determine how to route traffic to each model for prediction to determine the best model while maximizing prediction accuracy for users

Bandits(2)

- Bandit is stateful: before routing a request to a model, need to calculate all models' current performance

- Requires three things:
  - Model must be able to make online predictions
  - Preferably short feedback loops
  - A mechanism to collect feedback, calculate and keep track of each model's performance, and route prediction requests to different models based on their current performance

- Bandits are well-studied in academia and have been shown to be a lot more data efficient than A/B testing
  - require less data to determine which model is the best
  - at the same time, reduce opportunity cost as they route traffic to the better model more quickly
  - But a lot more difficult to implement than A/B testing because it requires computing and keeping track of models' payoffs

Thank You!

In our next session: