Online Experimentation

Irrespective of the problem you're working on, model experimentation and evaluation flow are always critical. In this lesson, we will go over the key steps and concepts in model experimentation and evaluation.

We'll cover the following

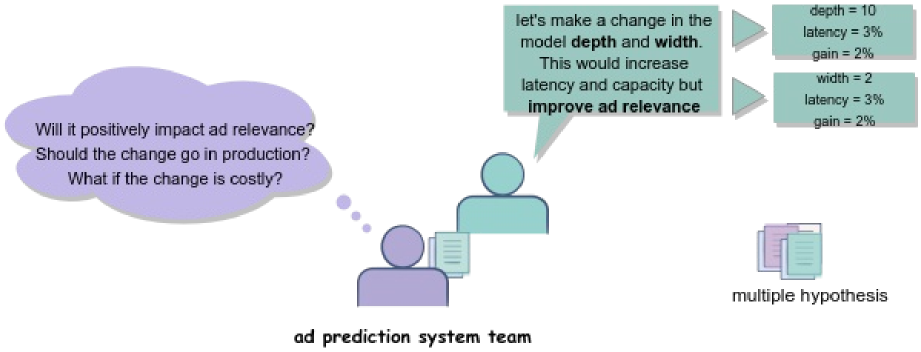
A successful machine learning system should be able to gauge its performance by testing different scenarios. This can lead to more innovations in the model design. For an ML system, “success” can be measured in numerous ways. Let’s take an example of an advertising platform that uses a machine-learning algorithm to display relevant ads to the user. The success of this system can be measured using the users’ engagement rate with the advertisement and the overall revenue generated by the system. Similarly, a search ranking system might take into account correctly ranked search results on SERP as a metric to claim to be a successful search engine. Let’s assume that the first version of the system (v0.1) has been created and deployed.

The initial version of the system is created

Hypothesis and metrics intuition

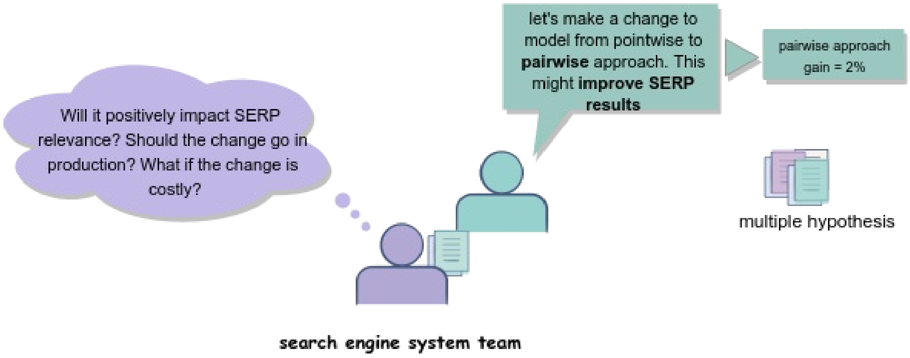
At any point in time, the team can have multiple hypotheses that need to be validated via experimentation.

Imagine, for instance, that the team designing an ad prediction system wants to test the hypothesis that increase in the neural network model depth (increase in hidden layers) or width (increase in activation units) will increase latency and capacity but will still have an overall positive effect on user engagement and net ad revenue.


Will it positively impact ad relevance? Should the change go in production? What if the change is costly?
let's make a change in the model **depth** and **width**. This would increase latency and capacity but **improve ad relevance**
depth = 10
latency = 3%
gain = 2%
width = 2
latency = 3%
gain = 2%
multiple hypothesis
ad prediction system team

Team desires to test multiple hypotheses to see the impact on the system

Similarly, a team working on designing a search engine wants to test the hypothesis that the pointwise algorithm instead of the pairwise algorithm would positively impact search relevance.


Will it positively impact SERP relevance? Should the change go in production? What if the change is costly?
let's make a change to model from pointwise to **pairwise** approach. This might **improve SERP results**
pairwise approach
gain = 2%
multiple hypothesis
search engine system team

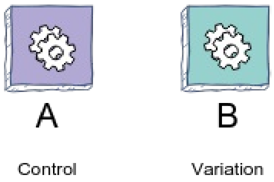
Team desires to test the hypotheses to see the impact on the system

So, to test the hypotheses, should the ML system v0.2 be created and deployed in the production environment? What if the hypothesis intuition is wrong and the mistake becomes costly?

This is where online experimentation comes in handy. It allows us to conduct controlled experiments that provide a valuable way to assess the impact of new features on customer behavior.

Running an online experiment

A/B testing is very beneficial for gauging the impact of new features or changes in the system on the user experience. It is a method of comparing two versions of a webpage or app against each other simultaneously to determine which one performs better. In an A/B experiment, a webpage or app screen is modified to create a second version of the same page. The original version of the page is known as the control and the modified version of the page is known as the variation.

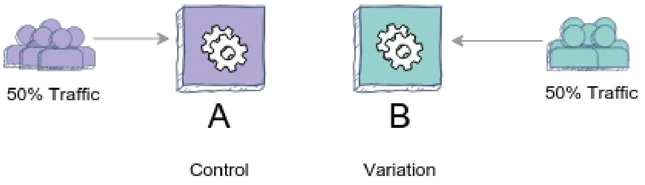


A/B testing

We can formulate the following two hypotheses for the A/B test:

- **The null hypothesis**, H0 is when the design change will not have an effect on variation. If we fail to reject the null hypothesis, we should not launch the new feature.
- **The alternative hypothesis**, H1 is alternate to the null hypothesis whereby the design change will have an effect on the variation. If the null hypothesis is rejected, then we accept the alternative hypothesis and we should launch the new feature. Simply put, the variation will go in production.

Now the task is to *determine if the number of successes in the variant is significantly better from the control*, i.e., if the conversion caused a positive impact on the system performance. This requires confidently making statements (using statistical analysis) about the difference in the variant sample, even if that difference is small. Before statistically analyzing the results, a power analysis test (https://en.wikipedia.org/wiki/Power_of_a_test) is conducted to determine how much overall traffic should be given to the system, i.e., the minimum sample size required to see the impact of conversion. Half of the traffic is sent to the control, and the other half is diverted towards the variation.



Traffic split evenly between version A and B

Measuring results

As visitors are served with either the control or variation/test version of the app, and their engagement with each experience is measured and analyzed through statistical analysis testing. Note that unless the tests are statistically significant, we cannot back up the claims of one version winning over another.

Computing statistical significance

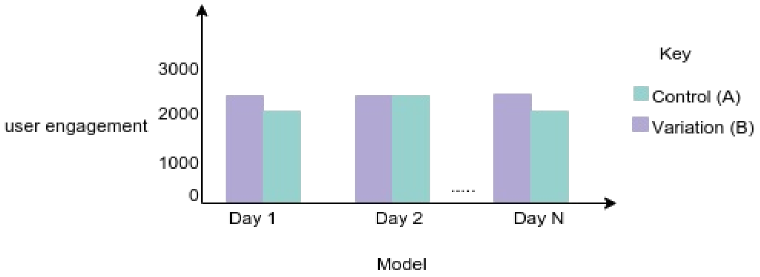
P-value (https://en.wikipedia.org/wiki/P-value) is used to help determine the statistical significance of the results. In interpreting the p-value of a significance test, a significance level (alpha) must be specified.

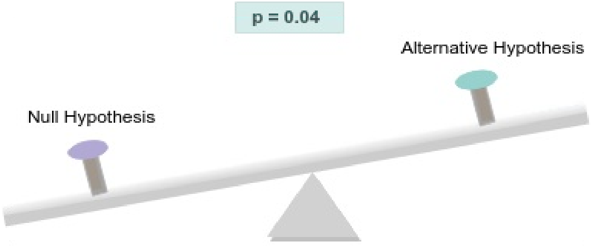
> The significance level is a boundary for specifying a statistically significant finding when interpreting the p-value. A commonly used value for the significance level is 5% written as 0.05.

The result of a significance test is claimed to be “statistically significant” if the p-value is less than the significance level.

- p <= alpha: reject H0 - launch the new feature
- p > alpha: fail to reject H0 - do not launch the new feature

If an A/B test is run with the outcome of a significance level of 95% (p-value ≤ 0.05), there is a 5% probability that the variation that we see is by chance.





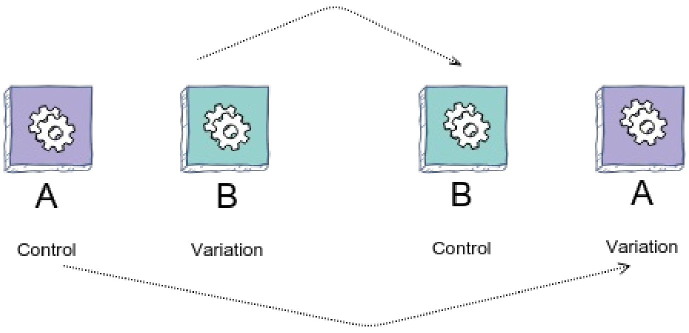
Statistical test analysis shows system B(variation) outperforms system A(control).

Measuring long term effects

In some cases, we need to be more confident about the result of an A/B experiment when it is overly optimistic.

Back Testing

Let's assume that variation improved the overall system performance by 5% when the expected gain was 2%. In the case of the ads prediction system, we can say that the rate of user engagement with the ad increased by 5% in variation (system B). This surprising change puts forth a question. Is the result overly optimistic? To confirm the hypothesis and be more confident about the results, we can perform a backtest (https://en.wikipedia.org/wiki/Backtesting). Now we change criteria, system A is the previous system B, and vice versa.



Backtest

We will check all potential scenarios while backtesting:

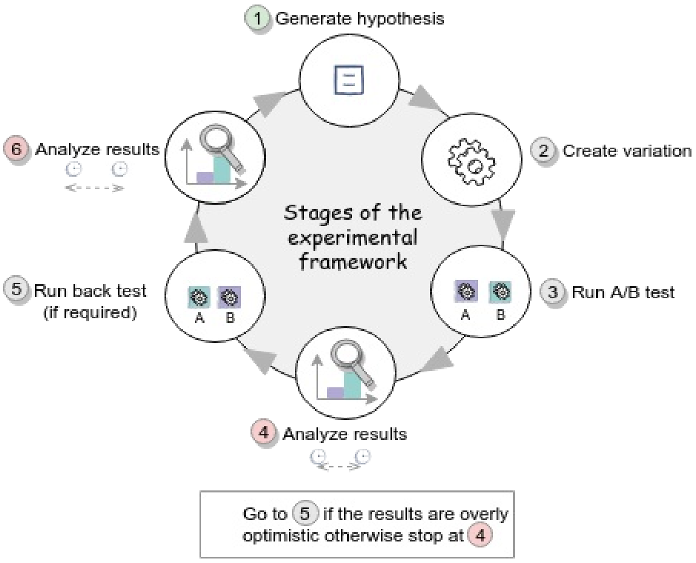
Do we lose gains? Is the gain caused by an A/B experiment equal to the loss by B/A experiment? Assume that the A/B experiment gave a gain of 5% and B/A experiment gave a loss of 5%. This will ensure that the changes made in the system improved performance.

Long-running A/B tests

In a few experiments, one key concern could be that the experiment can have a negative long term impact since we do A/B testing for only a short period of time. Will any negative effects start to appear if we do a long term assessment of the system subject to variation?

For example, suppose that for the ad prediction system, the revenue went up by 5% when we started showing more ads to users but this had no effect on user retention. Will users start leaving the platform if we show them significantly more ads over a longer period of time? To answer this question, we might want to have a long-running A/B experiment to understand the impact.

The long-running experiment, which measures long-term behaviors, can also be done via a backtest. We can launch the experiment based on initial positive results while continuing to run a long-running backtest to measure any potential long term effects. If we can notice any significant negative behavior, we can revert the changes from the launched experiment.



1 Generate hypothesis
2 Create variation
3 Run A/B test
A B
4 Analyze results
5 Run back test (if required)
A B
6 Analyze results
Stages of the experimental framework
Go to 5 if the results are overly optimistic otherwise stop at 4

Experimental framework stages

← Back

Training Data Collection Strategies

Next →

Embeddings

✓ Mark as Completed

Report an Issue

Ask a Question (https://discuss.educative.io/tag/online-experimentation__practical-ml-techniquesconcepts__grokking-the-machine-learning-interview)