

A/B Testing for AI

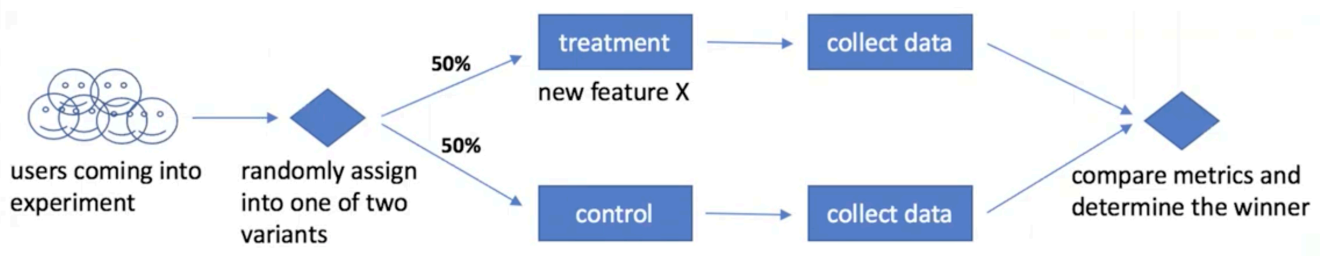
1/12/21 9:57 PM

Speaker: Pavel Dmitriev (VP of Data Science, Outreach)

Outline

- Intro to A/B testing
- Examples of real experiments
- Why A/B test AI systems
- Five pitfalls and lessons

A/B Tests in One Slide



- Can have more than two variants: A/B/C/etc. tests are common
- Must run statistical tests to confirm differences are not due to chance

A/B Tests are the best scientific way to prove causality

Examples of A/B Tests

- Different template formats for sales emails
- Search engine, (e.g. Bing.com, how many results should be displayed on one page)
- Operation system (e.g. Windows 10)

Reasons to A/B test our ML Models

1. Train/test sets get old quickly
2. Train/Test set misses entire class of examples
 - Some important features may have been neglected in model
3. Labels produced by human annotators may be inaccurate
 - E.g. In theory, users always prefer more images/videos. But in reality, more images/videos make it less convenient for users to click
4. All errors are not equal
5. UI matters
 - ML models is just one part of user experience, UI may make up certain scenarios where ML is limited
6. Model is part of a bigger system
 - E.g. output of one model is input of another. Even the model is improved, the downstream model might degrade
7. Model implementation has a bug

Five Pitfalls and Lessons for A/B Testing AI

Case: AI at Outreach - Personalized Customer Experience at Scale

- Outreach enables a sales rep to perform one-on-one personalized interactions with hundreds of prospective customers at the same time
- We use AI to optimize the scales process and personalize customer interactions

Lesson #1 AI vs. no-AI tests are tricky

- Introducing ML usually requires substantial changes to system architecture
 - o Example: introducing ML model to recommend sales rep best content to respond to prospect's email introduces extra backend processing and slows down other functionality
- Many factors other than ML model accuracy can impact the outcome of the test
- Solutions:
 - o A/B/C test:
 - A: No AI system without ML model
 - B: Hidden AI ML model runs in the backend, but results aren't used changes made to user experience
 - C: Full AI ML model runs in the backend and impacts user experience
 - o A vs. B measures performance impact, B vs. C measures impact on user experience

Lesson #2 Ensure Equal Learning Opportunity

- For models that learn and adapt on the fly, their quality depends on the amount of data they see
 - o Example: retraining rep content recommendation model daily based on user feedback
- The variant that has larger fraction of users gets more data for model training
- Solution:
 - o Ensure treatment and control are exposed to the same fraction of users
 - o If the test is not 50/50, ensure the control and "default" populations do not share data

Lesson #3 Zero in on the Target Scenario

- Often an ML model targets only a specific narrow user scenario
 - o Example: a rep content recommendation model may target only a specific type of prospect objection
 - o E.g. cost objection
- Only a subset of deals faces cost objection. When looking at "all up" results the signal may be lost in the noise
- Solution:
 - o Triggering - restricting experiment analysis to only the affected population
 - o Requires counterfactual logging - the cost objection scenarios need to be marked in the same way in both treatment and control variants
 - This may require running the model in control too, and logging the situations where it fired, even though model results are not used: A/B/C test design from lesson #1

Lesson #4 Beware of Side Effects

- While an improvement to an ML model may target a specific user scenario, it may inadvertently negatively (or positively) impact other scenarios
 - o E.g. an improvement to rep content recommendation model may have targeted not the right time scenario, but since it's a single model used for all scenarios it may have degraded accuracy for cost objection scenario
- Solution:
 - o While triggering is the key to understanding the impact in the targeted scenario, an all-up analysis should be performed to detect side effects
 - o Use a large comprehensive set of metrics, including even the metrics you don't expect to impact
 - o Use segments, such as objection scenario, persona, industry
 - Use higher thresholds when analyzing segments to avoid multiple testing pitfall (False Positive problem with too many statistical tests)

Lesson #5 Measure the full ML pipeline

- Often ML output is the result of multiple ML models stringed together
 - o Example: reg content recommendation model may consist of
 - NER model to detect if a known entity, such as competitor product, is mentioned in email
 - Scenario detection model which determines the type of objection the sales rep is facing
 - Recommendation model which recommends an email template to respond with
- To maximize learning need to understand how each part impacts the result
- Solution:
 - o Log for each part its result and confidence level
 - o Introduce segments based on confidence and prediction class of each part
 - Note: assessment of confidence needs to remain the same in both variants