MEASURING PRODUCT LAUNCH IMPACT THROUGH CAUSAL INFERENCE

Shafi Bashar, Machine Learning Engineer, Mixpanel, Inc.

INTRODUCTION

Mixpanel is a user analytics platform that enables our customers to analyze, learn and act on user behavior across their sites and apps. To help our customers better understand their users, we recently introduced the Impact Report. The Impact Report measures the effects of product or marketing launches on any KPI by leveraging causal inference methodologies.

The best way to understand how the Impact Report uses causal inference is to start with a concrete example. Suppose you are an early stage e-commerce website, and your front-end team recently launched a search bar on the product page of your site. You would assume that including a search bar will help your users find products faster - and therefore will improve your KPI metric - in this case, whether or not users added items to their carts. To measure the impact of this new feature launch, you can run an A/B experiment or a randomized control trial (RCT) [1] where your users are randomly allocated either to the old product page without the search bar (known as the control group) or to the new product page with the search bar (known as the treatment group). A/B experiments are considered the gold standard for measuring the causal impact of such launch events on KPI metrics. The randomized allocation of the treatment assignment - whether a user lands on the new or old product page with or without the search bar - ensures that the assignments are not confounded by any other factors. In this example, we can measure the causal impact of introducing the search bar simply by taking a difference in the number of items added to the cart between the treatment and the control groups.

In reality, it is not always possible to run A/B experiments. More often than not, an e-commerce website would launch several features at the same time. Running A/B experiments for each feature launch may be cost-prohibitive. Other times, they might encounter situations where it may not be possible to run A/B experiments at all. Situations like this occur whenever a new version of an app is released. A major UI revamp or a launch event with lots of marketing around it such that we can't show the change to only half of our user base can also create a similar problem. Situations like these, where we need to rely on retrospective observational data to measure the impact of a feature launch, are known as observational studies [2]. Unlike A/B experiments, treatment assignment in the case of observational studies is usually not random. When using observational data, the simple difference of a KPI metric between the users who used the new feature and the users who didn't may not be enough to measure launch impact.

In observational studies typical to an e-commerce setting, treatment assignments happen usually by means of a self-selection process in which users choose treatment for themselves. Imagine you released a new and improved product page with a search bar. Some of your users used the search bar - let's call them adopters. A lot of your users didn't use the search bar - let's call them non-adopters. Unlike the control and treatment groups in A/B experiments, the adopters and non-adopters in observational studies suffer from self-selection bias. In our e-commerce example, power users of the site who land on the product page more frequently are more likely to encounter and use the new feature compared to less engaged users. Power users in general also exhibit different KPI behavior patterns than non-power users. As a result, taking the KPI difference between adopters and non-adopters of the new feature will give you a biased estimate of the actual impact.

To obtain a more accurate estimate of the causal impact in observational studies, we first need to address the systematic difference between the characteristics of the adopters and the non-adopters. In academia, the use of propensity scores to reduce or eliminate self-selection bias in observational studies is widely adopted. To provide our customers with a more accurate and unbiased measure of the causal impact, Mixpanel's Impact Report utilizes a propensity score subclassification approach under the hood. In this whitepaper, we aim to provide readers with an overview of the propensity score subclassification approach used in causal inference from observational data and how we built it in the Impact Report.

The rest of this paper is organized as follows:

- In Section 1, we introduce the Neyman-Rubin counterfactual framework - the fundamental building block for many causal inference techniques.
- In Section 2, we provide an overview of the assumptions required to make causal inference in the framework introduced in Section 1.
- In Section 3, we discuss how to calculate average causal impact conditioned on the assumption from Section 2.
- In Section 4, we introduce the concept of propensity scores and the subclassification method.
- Finally in Section 5, we take a closer look into how we incorporate these methodologies into Mixpanel's Impact Report to provide our customers with an unbiased estimate of the impact of their feature launches on KPIs.

1. NEYMAN-RUBIN COUNTERFACTUAL FRAMEWORK

Originally proposed by Jerzy Neyman [3] and later generalized by Donald Rubin [4], the Neyman-Rubin counterfactual framework is based on the idea of potential outcomes, also known as counterfactual outcomes. This framework is the basic building block for any propensity score based causal inference technique.

For user-i, we denote $W_i \in \{0,1\}$ as the treatment assignment indicator. In the Impact Report, $W_i = 1$, when user-i is an adopter (performed the launch event), and $W_i = 0$ when user-i is a non-adopter (didn't perform the launch event). $X_i \in \mathbb{R}^d$ is the d-dimensional covariates or feature vectors for user-i. In the Impact Report, X_i correspond to users' past action data or events leading up to the new feature launch event. Y_i^1 is the counterfactual outcome when user-i is assigned the treatment, whereas Y_i^0 is the counterfactual outcome whenever user-i is not assigned to the treatment. In the Impact Report, these correspond to a user's potential metric event outcome depending on whether the user is an adopter or a non-adopter respectively.

Individual Treatment Effect (ITE)

$$\text{ITE}_i = Y_i^1 - Y_i^0 \quad \dots \quad (1)$$

Conditional Average Treatment Effect (CATE)

$$\tau(x) = \mathbb{E}[Y_i^1 \,|\, X_i = x] - \mathbb{E}[Y_i^0 \,|\, X_i = x] \quad \dots \quad (2)$$

Average Treatment Effect (ATE)

$$\text{ATE} = \mathbb{E}[Y_i^1] - \mathbb{E}[Y_i^0] \quad \dots \quad (3)$$

$$= \mathbb{E}_x[\tau(x)] \quad \dots \quad (4)$$

With these notations in place, in the Neyman-Rubin counterfactual framework, we define the treatment to have a causal effect on user-i if the counterfactual outcomes of the users differ, ie. $Y_i^1 \neq Y_i^0$. The Individual Treatment Effect (ITE) is defined as the difference between these two counterfactuals (Equation 1). The average causal effect, usually referred to as the Average Treatment Effect (ATE) is the average over all individuals' treatment effects (Equation 3). Conditional Average Treatment Effect (CATE) is the conditional ATE conditioned on covariates X_i (Equation 2). In practice, ATE is derived from CATE by marginalizing over the distribution of X_i (Equation 4).

Table 1 presents a sample dataset for our fictional e-commerce site. The table demonstrates the fundamental problem of causal inference. A user can either be an adopter or a non-adopter; consequently, we can only observe one of the two potential outcomes. This is unfortunate because to calculate ITE or individual users' causal effects (Equation 1), we need both the counterfactuals. The good news is, even though we cannot calculate ITE, under certain assumptions, we can calculate the average causal effect or ATE. In the next section, we will describe these assumptions.

2. CAUSAL ASSUMPTIONS

User i	Used search bar? W_i	Added item to the cart? Y_i	Power user? X_i	Added item to the cart after using the search bar? Y_i^1	Added item to the cart but didn't use the search bar? Y_i^0
	treatment	observed outcome	covariate	adopter counterfactual	non-adopter counterfactual
1	1	1	1	1	n/a
2	0	1	1	n/a	1
3	1	0	0	0	n/a
4	0	0	0	n/a	0
...					
99	1	1	1	1	n/a
100	0	0	1	n/a	0

Table 1

A user can either be an adopter or a non-adopter. As a result only one of the counterfactual outcome is available for each user here, making it impossible to calculate individual causal effect or ITE. This is the fundamental problem of Causal Inference.

Interested readers can find more details about these assumptions in [5], [6]. To provide our customers with an accurate estimate of the causal impact on KPIs as a result of their new feature launches, Mixpanel's Impact Report also relies on these assumptions. Any violations of these assumptions will result in a biased estimate.

Assumption 1 SUTVA

SUTVA or Stable Unit of Treatment Value Assumption actually consists of two assumptions - no interference among the users, and consistency of treatment.

No Interference Among the Users

The treatment assignment of one user should not interfere with the outcome of another user. In our e-commerce example, whether user-1 uses the search bar or not will only affect the KPI outcome of user-1 and not of user-2 or any other users for that matter. While the data sets of most Mixpanel customers comply with this assumption, there may be cases especially with our social network clients where this assumption may be violated. In social network sites, the behavior of one user can have a likely and measurable impact on the other users connected to their social or professional network [7]. For example, suppose a social network company like Twitter or Facebook decided to test whether recommending cat videos to users will increase their engagement with the site. In this case, a control user who was not recommended a cat video can still see the video if shared by another treatment user on their social network who was recommended the video.

Consistency

Consistency assumes that there is only a single version of the treatment. Going back to our e-commerce example, suppose both user-1 and user-2 used the new search algorithm. However, in case of user-1, the placement of the search box was at the top of the page, whereas for user-2, the placement was at the bottom of the page and the user needed to scroll down to get to the search box. Even though both users used the search box, because of the inconsistent placement, the consistency assumption is violated.

Assumption 2 Ignorability

The ignorability assumption is also known as the unconfoundedness assumption. We can assume the treatment assignments as random when conditioned on the covariates or feature vectors $X_i \in \mathbb{R}^d$. This enables us to control for self-selection bias and treat data collected from observational studies as if the data is from a hypothetical randomized A/B experiment, within the subset of users with the same value of the covariates.

In our e-commerce site example, power users of the site are more likely to be adopters than non-power users. In Table 1, we have a single covariate X_i indicating whether a user is a power user or not. When including both power and non-power users, we cannot infer causal impact simply by taking the KPI difference between the adopters and the non-adopters. However, if we only include the subset of power users (or the subset of non-power users), ie. we condition on the covariate X_i, we satisfy the ignorability assumption. We can then compare the KPI difference between adopter and non-adopter power users to infer the causal impact among the subset of power users. The same holds when we only include non-power users. In Mixpanel's Impact Report, we satisfy this assumption by conditioning on past user event data.

Assumption 3 Overlap

Also known as positivity, this assumes that any user has a positive probability of receiving all values of the treatment. In a randomized A/B test, positivity holds because we assign users at random with equal probability between the treatment and the control group. However, due to self-selection bias, this assumption can be violated in observational studies.

In our e-commerce example, suppose all of the power users are adopters of the new feature. In that case, we won't be able to infer the causal impact among the power users - simply because we don't have any corresponding non-adopter data.

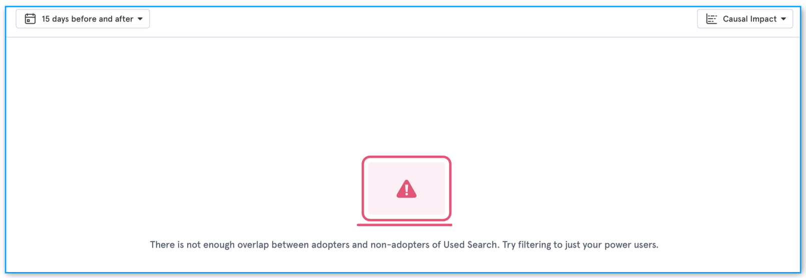


Figure 1

In Mixpanel's Impact Report, if the overlap assumption is violated, instead of providing a biased estimate, we warn our customers with the error message shown above.

The ignorability and overlap assumptions together are known as strong ignorability assumption [8] and are central to any propensity score based causal inference method.

3. ATE CALCULATION

The causal assumptions described in the previous section enable us to reformulate CATE in Equation 5 (see appendix for details). The two terms on the right hand side of Equation 5 are the mean observed outcomes for adopters and non-adopters respectively. We can get these values from the observed outcome column in Table 1.

Conditional Average Treatment Effect (CATE)

$$\tau(x) = \mathbb{E}[Y_i | X_i = x, W_i = 1] - \mathbb{E}[Y_i | X_i = x, W_i = 0] \quad \dots \quad (5)$$

This is great news, because we can now calculate CATE from Table 1. In contrast, in the original definition in Equation 3, we require the mean of the adopter counterfactual and the non-adopter counterfactual columns from Table 1. Not all of the counterfactuals are defined, making it impossible to calculate CATE.

For power users, CATE is the difference of the means of the outcome column between the adopters (ie. rows with treatment $W_i = 1$ and covariate $X_i = 1$) and the non-adopters (ie. rows with treatment $W_i = 0$ and covariate $X_i = 1$). We can similarly calculate CATE for non-power users from the rows with covariate $X_i = 0$. By averaging over the marginal distribution of X_i, we can calculate ATE from these two CATE.

4. PROPENSITY SCORE

Propensity scores are a solution to the curse of dimensionality problem which arises from too many covariates or confounding factors in the calculation of CATE. In our example from the last section, we only have one covariate, namely whether or not user-i is a power user. In practice, many participants may have many factors that influence their likelihood of using a new feature. As the number of covariates increases, the curse of dimensionality kicks in and it becomes prohibitively expensive to compare all covariates in the CATE calculation using the above mentioned method [9]. As an example, with p binary covariates, there will be 2^p possible combinations to compare - that's over a million possible values for when $p = 20$. Propensity score simplifies this problem by calculating a one dimensional score ranging from [0, 1] for each user indicating their propensity to use the new feature from all of the covariates. We can then group both adopters and non-adopters into buckets with similar scores, and calculate the ATE from adoption comparing only users that are similar to each other.

Introduced in [8], the propensity score for user-i is the conditional probability of assignment to treatment $W_i = 1$ given a vector of observed covariates X_i (Equation 6). In [8], it is also proved that, under strong ignorability conditions, the difference between the mean observed outcome for the adopters and the non-adopters for a given value of propensity score is equal to ATE at that value (Equation 7). Similar to Equation 4, we can derive ATE from Equation 7 by marginalizing over the distribution of $e(x_i)$.

Propensity Score

$$e(x_i) = P(W_i = 1 \mid X_i = x) \quad \dots \quad (6)$$

ATE conditioned on Propensity Score

$$= \mathbb{E}[Y_i \mid e(x_i), W_i = 1] - \mathbb{E}[Y_i \mid e(x_i), W_i = 0] \quad \dots \quad (7)$$

To estimate propensity scores, we first fit a binary logistic regression model [10] with the treatment indicator W_i used as the label data, and the vector of observed covariates X_i used as feature data. The model estimates a vector of regression parameters β_i. The estimated propensity score for user-i is then the output score of this model when covariates X_i is used as input.

Propensity Score Estimation from Logistic Regression Model

$$\tilde{e}(x_i) = \frac{1}{1 + \exp(-x_i^T \beta_i)} \quad \dots \quad (8)$$

Once we estimate the propensity score using the logistic regression model fit, we can create subclasses or strata of users based on their propensity scores. In Figure 2, we present a typical overlap plot of the distribution of estimated propensity scores between adopters (top) and non-adopters (bottom). We can group adopters and non-adopters with similar propensity scores into bins, with each bin as a subclass in the propensity score subclassification method (bins interval are of equal size here; in the subclassification method however, the interval ranges of the bins are calculated based on the quantiles of propensity scores). Within each bin, the propensity scores between the adopters and the non-adopters are very similar - in other words both adopters and non-adopters in the bin have comparable self-selection bias and we can assume strong ignorability and use Equation 7 to determine the conditional causal effect or CATE for the bin.

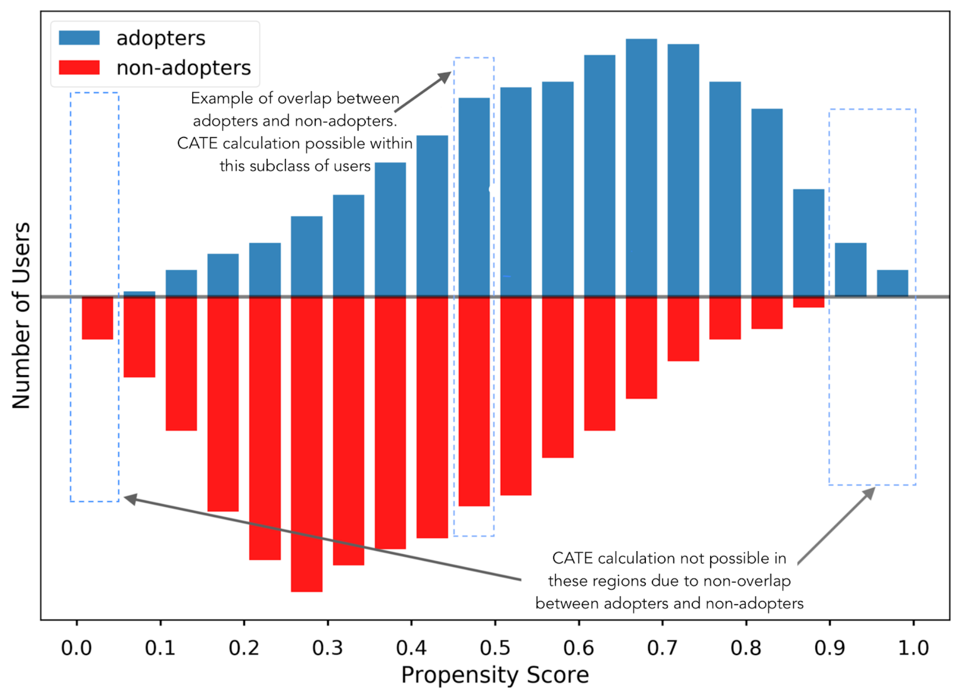


Figure 2

Estimated propensity score overlap plot between the adopters and non-adopters.
The propensity scores for the adopters and the non-adopters within the same bin are similar and therefore, ignorability assumption holds within each bin enabling us to calculate CATE within each bin.

Few things are of interest here. First, the distribution of adopters is right skewed towards higher values of propensity scores and the distribution of non-adopters is left skewed towards lower values. This is typical and indicates the self-selection bias present in observational studies. Second, the extreme left bins and the extreme right bins may only contain non-adopters and adopters respectively, thereby violating the overlap assumption which will result in an incorrect estimate. In Mixpanel's Impact Report, we address this problem in two different ways - first, we used the methodologies described in [11] and only consider the set of users with propensity scores within the range [0.1, 0.9]. Second, if the violation is severe, rather than showing our customers incorrect estimates, we throw the error condition shown in Figure 1.

Once we calculate the CATE for each bin, the average causal effect can be estimated by taking the weighted average across all subclasses. Rosenbaum, Rubin [8] proved that if the propensity scores of the users within each subclass are very close to each other, the weighted average across the subclasses will provide an unbiased estimate of the ATE. The interested reader can find a step-by-step breakdown of the propensity score subclassification approach in the appendix.

5. MIXPANEL IMPACT REPORT

Readers can find instructions on how to use Mixpanel's Impact Report in our help documentation [12]. QBQ - our community page also contains a best practice guide [13]. In this section, we will focus on the machine learning specific implementation details of the report.

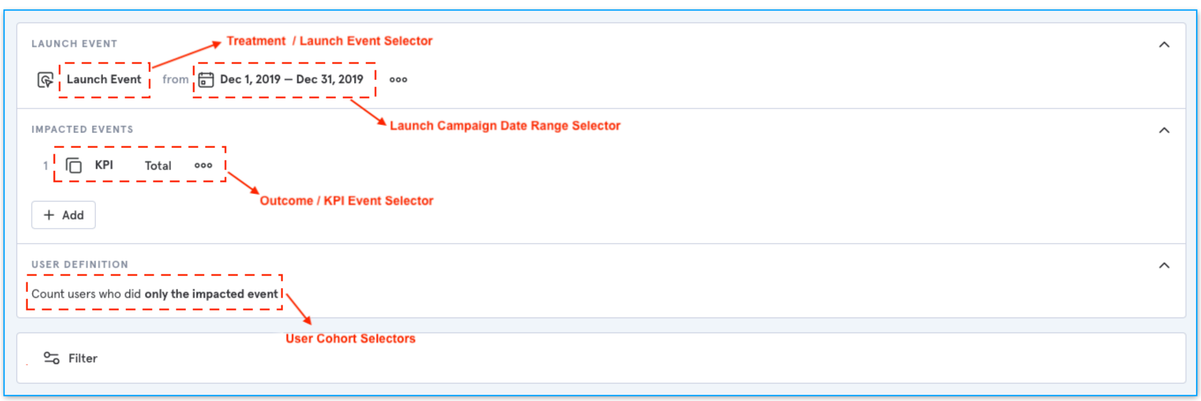


Figure 3

Customer can select the Launch Event, KPI and User Cohort using the appropriate selectors in the Impact Report.

Customers can select their desired launch event, KPI, date range and user cohorts using the appropriate selectors in the report as shown in Figure 3. Every time a customer uses the Impact Report, we train a logistic regression model under the hood to calculate the estimated propensity scores for all the users in the user cohort. To train the logistic regression model, we first generate labels and features for the model training from the user events stored in Mixpanel's database.

The user timeline diagram showed in Figure 4 will help us demonstrate the label and and feature data generation process. Here, Day 0 is the release date - the first day when the new feature is launched. To collect the label data, we look within a launch window - [day 0, day 15] is the default launch window in the Impact Report. The report also allows our customers to adjust the window to their desired range using the date range selector. In this example, user A, B and C performed the launch event one or multiple times within this launch window whereas user D didn't perform the launch event. In our ML model, we assign label-1 to users A, B and C; we assign label-0 to user D.

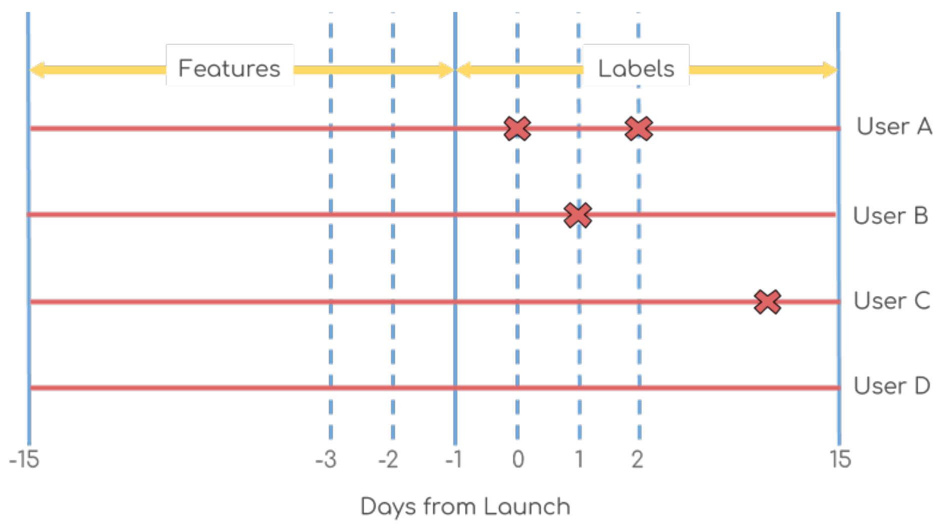
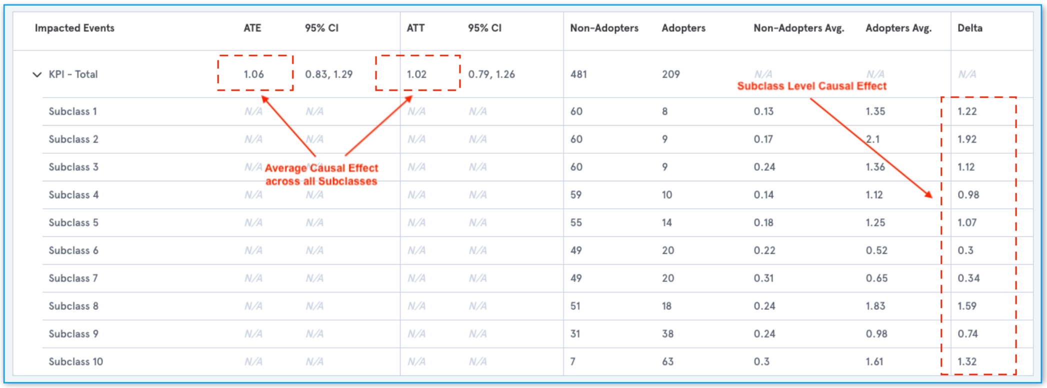


Figure 4

User timeline diagram above indicates two disjoint time ranges used to generate label and feature data. Whether a user performed the launch event or not within [day 0, day 15] is used to generate label data. All of users activities leading up to day 0 within [day-15, day-1] are used to generate feature data.

To calculate the feature data for our model training, we use past activities of users leading up to day 0 of the launch event. We collect features and labels data from disjoint time windows to prevent any potential label leakage. At Mixpanel, we track a substantial amount of events or user action data for our customers. The recency and frequency of these events performed by the users leading up to the launch event can be a good indicator of users' propensity to perform the launch event. All of these events are used as features in our ML model. We use a statistical significance based feature selection stage before the model training stage in order to reduce the number features for model training. To measure the goodness of fit of the trained model, we perform a 5-fold cross validation.

After model fit, we calculate the estimated propensity scores, perform subclassification on the propensity scores and estimate causal effect within each subclass. In the report, in addition to the average causal effect, we also present the subclass level causal effect information to provide our customers with more insight into different subgroups of their user-base based on the propensity to perform the launch event (Figure 5).



Impacted Events	ATE	95% CI	ATT	95% CI	Non-Adopters	Adopters	Non-Adopters Avg.	Adopters Avg.	Delta
KPI - Total	1.06	0.83, 1.29	1.02	0.79, 1.26	481	209	N/A	N/A	N/A
Subclass 1	N/A	N/A	N/A	N/A	60	8	0.13	1.35	1.22
Subclass 2	N/A	N/A	N/A	N/A	60	9	0.17	2.1	1.92
Subclass 3	N/A	N/A	N/A	N/A	60	9	0.24	1.36	1.12
Subclass 4	N/A	N/A	N/A	N/A	59	10	0.14	1.12	0.98
Subclass 5	N/A	N/A	N/A	N/A	55	14	0.18	1.25	1.07
Subclass 6	N/A	N/A	N/A	N/A	49	20	0.22	0.52	0.3
Subclass 7	N/A	N/A	N/A	N/A	49	20	0.31	0.65	0.34
Subclass 8	N/A	N/A	N/A	N/A	51	18	0.24	1.83	1.59
Subclass 9	N/A	N/A	N/A	N/A	31	38	0.24	0.98	0.74
Subclass 10	N/A	N/A	N/A	N/A	7	63	0.3	1.61	1.32

Figure 5

Mixpanel's Impact Report provides both subclass level and average causal effect.

For average causal effect, we provide our customers with two sets of metrics - ATE and ATT. In the previous sections, we discussed the concept of ATE in detail. ATT (Average Treatment Effect on the Treated) indicates causal impact of performing the launch event on the KPI metric only among the adopters. In a randomized A/B experiment, ATE and ATT are generally equal. However, in observational studies, these two metrics can be different due to self-selection bias.

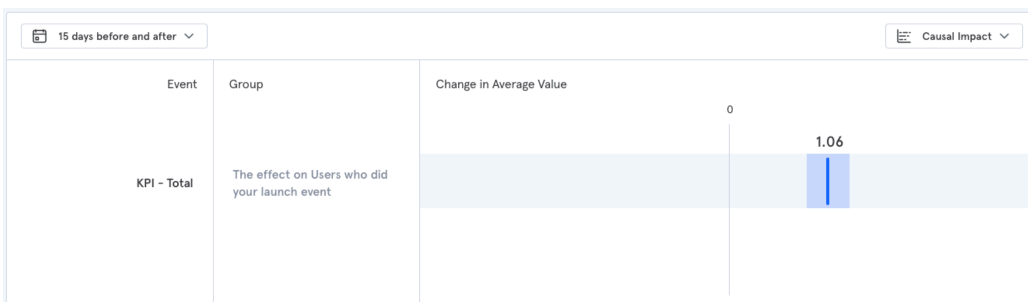


Figure 6

Error bar indicating the ATE estimate and 95% confidence interval of the estimate.
If the bar does not contain zero, the result is considered to be statistically significant.

ATT can provide additional insights especially in an e-commerce setting, where power users tend to be the adopters. Power users often generate more revenue and may be worth more from an LTV perspective. Product managers are often interested in understanding how a new feature launch affects their product's power users. ATT can be a good proxy to measure that in this setting.

In addition to the point estimate of average causal effect, we also provide the 95% confidence interval of the estimate using an error bar plot (Figure 6). Customers can infer statistical significance of the estimate from this plot by observing whether the interval crosses zero or not and how wide or narrow the interval is.

CONCLUSION

This whitepaper provides readers with an overview of how to measure the impact of a product launch on KPIs from observational data using causal inference. We discussed how matching adopters and non-adopters based on their propensity scores for performing a launch event can reduce confounding and self-selection bias inherent in observational data and can therefore provide an unbiased estimate of the average causal effect. We also provided an under the hood look of how Mixpanel's Impact Report uses the propensity score subclassification approach to calculate impact.

The inclusion of the propensity score method in the Impact Report makes it a powerful tool that enables our customer to assess the impact of both their new and past product launches on important KPIs with ease. If you haven't already, try the new Impact Report and let us know what you think!

APPENDIX

Average Causal Effect Equations

Using the causal assumptions from Section 2, we can rewrite Conditional Average Treatment Effect (CATE) from Equation 2 as follows:

$$
\begin{aligned}
\text{CATE } \tau(x) &= \mathbb{E}[Y_i^1 \mid X_i = x] - \mathbb{E}[Y_i^0 \mid X_i = x] \quad \dots \quad (i) \\
&= \mathbb{E}[Y_i^1 \mid X_i = x,\ W_i = 1] - \mathbb{E}[Y_i^0 \mid X_i = x,\ W_i = 0] \quad \dots \quad (ii) \\
&= \mathbb{E}[Y_i \mid X_i = x,\ W_i = 1] - \mathbb{E}[Y_i \mid X_i = x,\ W_i = 0] \quad \dots \quad (iii)
\end{aligned}
$$

Above, the ignorability assumption enables us to go from Equation (i) to Equation (ii). We made use of the consistency assumption to get Equation (iii) from Equation (ii). The overlap assumption ensures that both the terms of Equation (iii) is defined for all values of covariate X and treatment W.

Propensity Score Subclassification Method

The propensity score subclassification approach can be broken down into the following steps [6]:

Step 1:

Estimate propensity scores $e(x_i)$ from the trained logistic regression model.

Step 2:

Using quantiles of the estimated propensity scores, divide the users into K subclasses. In the Impact report, we use K=10.
We denote $0 = c_0 < c_1 < \dots < c_{10} = 1$ as the boundary values of the subclasses; Then user-i belongs to subclass k if $c_{k-1} < e(x_i) < c_k$

Step 3:

Within each subclass, calculate Δ_k- the difference between the mean adopters KPI value and the mean non-adopters KPI value.

Step 4:

Estimate average causal effect from the weighted average over Δ_k. Denote n_k as the total number of users (adopter and non-adopters) in subclass k and N as the total number of users across all subclasses

$$\text{ATE} = \sum_{k=1}^{K} \frac{n_k}{N} \Delta_k$$

REFERENCE

[1] "Randomized controlled trial", Wikimedia Foundation, Inc. [Online]. Available: https://en.wikipedia.org/wiki/Randomized_controlled_trial

[2] "Observational study", Wikimedia Foundation, Inc. [Online]. Available: https://en.wikipedia.org/wiki/Observational_study

[3] J. Splawa-Neyman, D. M. Dabrowska, and T. P. Speed, "On the Application of Probability Theory to Agricultural Experiments. Essay on Principles. Section 9", Statistical Science, vol. 5, no. 4. pp. 465–472, 1990.

[4] D. B. Rubin, "Estimating causal effects of treatments in randomized and nonrandomized studies," Journal of Educational Psychology, vol. 66, no. 5. pp. 688–701, 1974.

[5] "coursera.org" [Online]. Available: https://www.coursera.org/lecture/crash-course-in-causality/causal-assumptions-f5LPB

[6] S. Guo and M. W. Fraser, Propensity Score Analysis: Statistical Methods and Applications. SAGE Publications, 2014.

[7] James D. Wilson and David T. Uminsky, "The power of A/B testing under interference", arXiv 1710.03855, 2017.

[8] P. R. Rosenbaum and D. B. Rubin, "The Central Role of the Propensity Score in Observational Studies for Causal Effects" 1981.

[9] P. R. Rosenbaum, "Design of Observational Studies", Springer Series in Statistics. 2010

[10] "Logistic regression", Wikimedia Foundation, Inc. [Online]. Available: https://en.wikipedia.org/wiki/Logistic_regression

[11] R. K. Crump, V. J. Hotz, G. W. Imbens, and O. A. Mitnik, "Dealing with limited overlap in estimation of average treatment effects", Biometrika, vol. 96, no. 1. pp. 187–199, 2009.

[12] "Impact Report," Mixpanel Help Center. [Online]. Available: http://help.mixpanel.com/hc/en-us/articles/360034129112-Impact-Report

[13] "Impact Report Best Practices | Mixpanel Community." [Online]. Available: https://community.mixpanel.com/announcements-6/impact-report-best-practices-3887

ACKNOWLEDGEMENT

The author would like to thank the following for their feedback and review:
Zach Wener-Fligner, Brandon Skerda, Wonja Fairbrother, Iris McLeary, Sam Chow, Krishna Vuppala, Tiffany Qi, Adam Kinney and Harsh Patel.