ForTune: Running Offline Scenarios to Estimate Impact on Business Metrics

Georges Dupret
Spotify, Inc.
New York, NY, USA
gdupret@spotify.com

Konstantin Sozinov
Spotify, Inc.
New York, NY, USA

Carmen Barcena Gonzalez
Spotify, Inc.
London, UK

Ziggy Zacks
Spotify, Inc.
New York, NY, USA

Amber Yuan
Spotify, Inc.
New York, NY, USA

Ben Carterette
Spotify, Inc.
New York, NY, USA
benc@spotify.com

Manuel Mai
Spotify, Inc.
Berlin, Germany

Andrey Gatash
Spotify, Inc.
New York, NY, USA

Leo Lien
Spotify, Inc.
New York, NY, USA

Shubham Bansal
Spotify, Inc.
New York, NY, USA

Roberto Sanchis-Ojeda
Spotify, Inc.
Madrid, Spain

Mounia Lalmas-Roelleke
Spotify, Inc.
London, UK

ABSTRACT

Making ideal decisions as a product leader in a web-facing company is incredibly challenging. Beyond navigating the ambiguity of customer satisfaction and achieving business goals, leaders must also ensure their products and services remain relevant, desirable, and profitable. Data and experimentation are crucial for testing product hypotheses and informing decisions. Online controlled experiments, such as A/B testing, can provide highly reliable data to support decisions. However, these experiments can be time-consuming and costly, particularly when assessing impacts on key business metrics like retention or long-term value.

Offline experimentation allows for rapid iteration and testing but often lacks the same level of confidence and clarity regarding business metrics impact. To address this, we introduce a novel, lightweight, and flexible approach called *scenario analysis*. This method aims to support product leaders' decisions by using user data and estimates of business metrics. While it cannot fully replace online experiments, it offers valuable insights into trade-offs involved in growth or consumption shifts, estimates trends in long-term outcomes like retention, and can generate hypotheses about relationships between metrics at scale.

We implemented scenario analysis in a tool named ForTune. We conducted experiments with this tool using a publicly available dataset and reported the results of experiments carried out by Spotify, a large audio streaming service, using ForTune in production.

In both cases, the tool reasonably predicted the outcomes of controlled experiments, provided that features were carefully chosen. We demonstrate how this method was used to make strategic decisions regarding the impact of prioritizing one type of content over another at Spotify.

CCS CONCEPTS

• **Applied computing → Decision analysis**;

KEYWORDS

Scenario Analysis, Sensitivity Analysis, A/B-testing Prediction, A/B-testing Setup

1 INTRODUCTION

Product leaders in web-facing companies continually face challenging decisions. Their decisions impact customer satisfaction in unpredictable ways. Besides navigating the ambiguity of customer satisfaction and achieving business goals, leaders must also ensure that products and services to remain relevant, desirable, and profitable. However, it is not always clear how a decision today will contribute future success.

To make well-informed informed decisions, companies often conduct numerous online experiments, with some running over 200 concurrent experiments [14]. However, running large number of experiments introduces significant complications.

From an infrastructural perspective, exposing users to multiple experiments simultaneously complicates evaluation, as noise and interaction effects can bias results. Additionally, controlled experiments can test only a limited number of hypotheses, requiring teams to choose configurations wisely. Furthermore, not all hypotheses can be tested online due to technical constraints or pre-allocated traffic to other experiments.

From an outcomes and analysis perspective, online experiments might not provide a holistic picture of how a new product or service affects users. Long-term outcomes such as user satisfaction, retention and financial metrics such as revenue and gross profits are difficult to measure in real-time, especially within the confines of a short experiment.

Given these challenges, an offline method that tests hypotheses and overcomes some limitations of online experimentation would be valuable. Specifically, a method that allows testing many configurations and projects the impact on long-term key metrics.

The methodology behind ForTune, the tool we introduce in this paper, is designed to gain insight into what the results of a controlled experiment might look like before deploying one. To that aim, we rely on a "scenario" that describes, in general terms, how we think some key business metrics will be affected by the treatment. It can be understood as a kind of sensitivity analysis of the business metrics with respect to a set of control variables, but without the need of an explicit model. It is not meant to replace controlled experiments, but rather to help prioritize between several experiment candidates. It can also be used to make an educated guess when it is not possible to set up a controlled experiment.

Few flexible and lightweight methods are available to carry out such predictions. Developing a predictive model is one possibility, but it requires deep system knowledge and is typically complex and expensive in terms of computation and human effort. In contrast, the method we present is simple, flexible, and easy to implement.

For example, imagine we expect a 1% increase in podcast consumption in a online audio app, such as Spotify, following an algorithmic or interface modification. Should we release this new version? Does an increase in podcast consumption lead to increased profit? What would be the impact on user retention? Would there be a substitution effect, leading to decreased music consumption? The only definitive way to answer these questions is to run a controlled experiment, which is time consuming and expensive, especially when evaluating long term metrics.

Our proposed method predicts the impact of such a controlled experiment before deploying changes and without needing to develop and train a predictive model. The core idea is to collect past consumption data and re-weight observations to match the expected changes. For instance, returning to our example, we would expect more users characterized by a high podcast consumption after deploying the app change. If those users exhibit a higher retention rate, the test branch of the experiment would also show a higher retention rate.

We successfully verified this idea on a publicly available dataset, a large controlled experiment from an advertising company, Criteo, that provides online display advertisements. We also verified the predictions using several controlled experiments with proprietary data from Spotify. ForTune is both expressive and flexible, leading

to its adoption at various levels within Spotify to inform strategic decision-making.

Section 2 presents the method. In Section 2.1 we use an example to illustrate it. We then extend the method to more general scenarios in Section 2.2 and summarize it in Section 2.3. We discuss caveat and limitations in Section 2.4 and we review related works in Section 3. Finally, we present experiments in Section 4: those conducted on the Criteo dataset in Section 4.1, and on Spotify proprietary data in Section 4.2.

2 THE FORTUNE ALGORITHM

We describe our proposed approach, *ForTune*. We begin with a simple example to build intuition and then generalize the solution to address more complex problems.

2.1 The Intuition behind the Algorithm

Suppose you own a shoe store and launch a campaign to double the number of male customers. How will this impact the average sale price? A simple estimate can be obtained by re-weighting past male customers twice as much as female customers (or resampling accordingly) and then computing the weighted average of sale prices. An example is provided in Table 1.

gender	age	shoe size	marital status	price	weight
F	97	34	married	180	1
F	85	53	single	150	1
M	80	47	single	390	2
M	45	49	married	180	2
M	54	50	single	300	2
M	79	54	single	340	2
F	69	39	married	250	1

Table 1: Hypothetical shoe store. Before the campaign, the average shoe sale is the average of the price column, i.e. $256. After the campaign we expect twice as many male customers as before. We do not actually know the exact characteristics of these new customers, so a safe bet is to assume that they will be similar to the current male customers. Therefore, we replicate all male rows in the table. The estimated average sale price after the campaign is estimated as the average of the price column weighted by the weight column, i.e. $273.

This estimated average sale price will be accurate as long as the original male customers are representative of the new customers. However, if the campaign specifically targets males under 50 years old, and only one current customer fits this demographic, our prediction will rely exclusively on him, making it unreliable. Generally, we need enough samples to ensure our estimator controls the variance.

If we have more information about the new customers beyond gender, we can refine our predictions. For instance, if we target single male customers, a quick look at Table 1 reveals that they spend more, suggesting the average sale price should be higher after the campaign. A similar conclusion applies if we target older male customers, such as those around 65 years of age.

In the next section, we formalize these ideas and propose a method to derive a set of weights that incorporate assumptions about the campaign's impact, while making minimal assumptions about customer characteristics.

2.2 Building More Complex Scenarios

In the previous section, we discussed a simple case where the only constraint on the population was the gender mix. Often, constraints apply to a continuous variable (like age) instead of a binary one. Additionally, there may be scenarios where multiple variables need to be constrained.

Let X be a $N \times D$ feature matrix, and y an N dimensional vector representing the metric of interest. In Table 1, the features include gender, age, shoe size and marital status. The vector y represents the business metric we want to predict, which in this case is the price. The dimensions are $N = 7$ and $D = 4$.

The objective is to identify a N dimensional vector ω of weights that incorporate the expected change. In the shoe store example, this means doubling the proportion of male customers, i.e. denoting by X_n the nth observation in X and $\mathbb{1}_M(X_n)$ the indicator function for males:

$$\frac{1}{N} \sum_{n=1}^{N} \omega_n \mathbb{1}_M(X_n) = \frac{2}{N} \sum_{n=1}^{N} \mathbb{1}_M(X_n) \tag{1}$$

This constraint requires that the weights ω adjust the proportion of males in X to be twice the original proportion, i.e twice $\frac{1}{N} \sum_{n=1}^{N} \mathbb{1}_M(X_n)$. The solution to this constraint is not unique; many different vectors ω can satisfy it.

If the campaign also targets older males, an additional constraint must be enforced. For example, to ensure that the average age of male customers after the campaign is 65 years old, the following constraint would be added:

$$\frac{1}{N} \sum_{n=1}^{N} \omega_n \mathbb{1}_M(X_n) X_{n,\text{age}} = 65 \tag{2}$$

Similarly, we could opt for a more relaxed constraint by only requiring that the average male age be greater than 65 years old:

$$\frac{1}{N} \sum_{n=1}^{N} \omega_n \mathbb{1}_M(X_n) X_{n,\text{age}} \geq 65 \tag{3}$$

There are still many vectors of weights that satisfy the constraints in Equation (1) and Equation (2) or (3). Solutions that give large weights to a small number of observations are undesirable because they would rely excessively on a subset of the data, leading to high variance. Instead, it is reasonable to look for a solution that distributes importance more evenly across all observations, while remaining compatible with the constraints. This approach suggests maximizing the entropy of the weights while satisfying the constraints. The optimization problem can be formulated as follows:

$$\underset{\omega}{\arg\max} \quad \mathcal{H}(\omega) \tag{4a}$$

$$\text{subject to} \quad \frac{1}{N} \sum_{n=1}^{N} \omega_n \mathbb{1}_M(X_n) = \frac{2}{N} \sum_{n=1}^{N} \mathbb{1}_M(X_n), \tag{4b}$$

$$\frac{1}{N} \sum_{n=1}^{N} \omega_n \mathbb{1}_M(X_n) X_{n,\text{age}} = 65, \tag{4c}$$

$$\omega_n \geq 0 \, n \in \{1 \dots N\}, \tag{4d}$$

$$\sum_{n}^{N} \omega_n = 1 \, n \in \{1 \dots N\} \tag{4e}$$

We added two natural constraints: the weights must be positive, and they must sum to one, as shown in in Equations (4d) and (4e).

If no constraints are imposed beyond these natural ones, the solution to this problem is to assign equal weights to each observation, i.e. $\omega = \frac{1}{N}$. This aligns with the fact that, prior to the campaign, the target metrics are estimated as the sample average.

2.3 The Algorithm

We now formalize the algorithm we built an intuition for in the previous two sections.

The ForTune Algorithm involves solving the following convex optimization problem:[1]

$$\begin{aligned} \underset{\omega}{\arg\max} \quad & \mathcal{H}(\omega) \\ \text{subject to} \quad & g_i(x) \leq 0 \quad i = 1, \dots, m, \\ & h_j(x) = 0 \quad j = 1, \dots, p, \\ & \omega_n \geq 0 \quad n \in \{1 \dots N\}, \\ & \sum_{n}^{N} \omega_n = 1 \quad n \in \{1 \dots N\} \end{aligned} \tag{5}$$
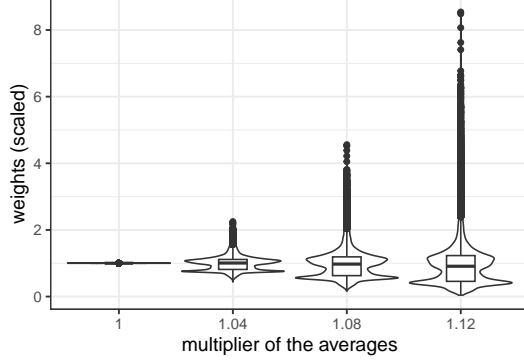
Here, the h_i are $p \geq 0$ affine functions and $g_j \leq 0$ are $m \geq 0$ convex inequalities. The last two constraints are the natural constraints. The functions g_i and h_j define the scenario associated with the problem. The problem of finding the right set of weights can be solved using an appropriate convex solver, and the solution is unique.

Once the weights ω are known, the metric of interest t is estimated as

$$\hat{t} = \sum_{n}^{N} \omega_n t_n \tag{6}$$

This algorithm provides a point estimate of the prediction under the scenario hypotheses. To obtain a distribution of point estimates and measure of uncertainty, the dataset can be divided into B distinct random subsets, and the point estimate can be independently calculated for each of subset. Another approach is bootstrapping [6], where the dataset is repeatedly sampled with replacement to create B subsets. The experiments in Section 4 will highlight the need to estimate uncertainty.

Figure 1: Box plots of the Resampling Weights for the Criteo dataset. The weights have been multiplied by the number of observations so a weight of 1 means that the observations has the same importance in the control and treatment branches. A weight of 5 means that the corresponding observations is five times more influential in the test branch than before resampling. We set the constraints on features f_1, f_4, f_7 and f_{10} to be same multiples of the corresponding averages. The multiples are reported on the y axis. The further from the original means (for which the multiple is 1.0), the larger the weights' spread.

2.4 Limitations

Not all constraints are feasible or lead to realistic outcomes. It is important to remember that predictions are obtained by giving more importance to users who help satisfying the constraints. If no user can help, then the constraint is unrealizable. Returning to the example above, if the constraint required that the average male age be 100 years old instead of 65 (see Equation (2)), no combination of weights would satisfy the convex problem. Less obvious sets of constraints might also be unrealizable, but in practice they are easy to detect because the quadratic solver will fail to return a solution.

A more subtle case arises when conditions are realizable, but the solution assigns large weights to a limited number of observations, potentially leading to a high variability. In the previous example, this would occur if only a few users were aged 65 or older. In such cases, the solver will give large weights to these few observations, resulting in an estimate based on a small number of data points. This problem can be diagnosed by examining the weights distribution to identify extreme values. Figure 1 show the effect of applying constraints set to 104, 108 and 112% of the original feature averages. More details are provided in Section 4 where we discuss the Criteo dataset, but it should be clear that the further the constraints are from the control data averages, the more likely we are to find outlier weights.

It is also important to remember that the ForTune is useful for predicting averages, not absolute values. For instance, in the shoe shop example from Section 2.1, we predicted the average sell price, not the total sales revenue. This is because the campaign likely increased the number of customers by an amount that we ignore.

---

[1]We require the problem to be convex for convenience. We could generalize it to any type of constraints but we have not encountered in practice a case where this is useful.

Not accounting for changes in the user base can lead to apparent paradoxes. If the campaign targeted women instead of men, who in this particular example buy cheaper shoes, the average price would decrease even though total sales might increase due to the additional customers.

As in observational studies [9], obtaining an unbiased estimate of the effect requires identifying confounding covariates. This typically relies on domain knowledge, although automated methods have been developed to help identify causal graphs [25]. At Spotify we built knowledge by comparing ForTune predictions with the results of past controlled experiments. When predictions were off, we investigated the reasons and identified missing variables. An example is provided in Section 4.2.

3 RELATED WORKS

This work began with an investigation into Sensitivity Analysis, which is typically defined as studying the relationship between model output uncertainty and each input variable. However, in practice it encompasses much more. As Iooss & Lemaître [12] note, one application aligns with ForTune: "mapping the output behavior as a function of the inputs, focusing on a specific domain of inputs if necessary."

Sensitivity Analysis can be divided into local and global methods [24]. Local Sensitivity Analysis focuses on the impact of small input perturbations around a nominal value of the features (typically the mean) on the model output. When the model is sufficiently simple, Taylor series expansions can approximate the model, allowing for an analytical differential sensitivity index to be derived [30]. Global sensitivity, on the other hand, examines the model's overall response (averages over variations of all features) by exploring a finite region of the input domain. While early global sensitivity analyses technique assumed feature independence, newer approaches do not [15, 30].

ForTune can be viewed as a Sensitivity Analysis method because it examines how output depends on input variations. However, there are fundamental methodological differences and differences in applicability. Sensitivity Analysis requires a model of the relationship between inputs and outputs, which can incorporate external knowledge (e.g., Bayesian models that explicitly define relationships between features). In contrast, ForTune only relies only on resampling / re-weighting the data and implicitly accounts for features correlations. While accurate feature selection is crucial for ForTune, this is also true for designing analytical models.

Another fundamental methodological difference between Sensitivity Analysis and ForTune is how the scenarios are defined. While the latter typically requires only a set of global constraints on feature averages to be set, the predictive models used in Sensitivity Analysis requires all the individual features to be specified before generating a prediction. To see why this is a hard problem, imagine that Spotify is interested in estimating the impact on podcast consumption of an increase of 5% in music consumption. Do we increase uniformly music consumption by 5% before running the predictive model? That is probably not realistic. Other features used by the predictive model probably need adjustment because they are not independent from music consumption. How is their values to be decided?

While ForTune and Sensitivity Analysis share a similar goal, other methods share a similar approach. Weighting observations to query the data is common in statistics. Examples include stratified sampling [3] and the Horvitz–Thompson estimator [11]. In election polling [29], data from a small sample is used to predict larger population behavior. To ensure accuracy, cohorts within the sample are weighted to represent the broader population, considering factors such as race, age, gender, education, and geographical location. These factors play a significant role in voting behavior, and the demographics of the sample group should match the demographics of the voting population. The success of Stratified Sampling and ForTune depends heavily on selecting the right set of features to match the original data to the target population.

Inverse Probability Weighting (IWS) and ForTune both aim to estimate quantities related to a target population different from the one the data was collected from. The key difference is that IWS assumes a sample of the target dataset is available, while ForTune relies on a limited set of global constraints to characterize the target population. Propensity Score Matching (PSM) [20] is another technique for estimating the effect of an intervention by accounting for covariates that predict receiving the treatment. While we use this method in Section 4.1.3 to compare it with ForTune unlike PSM, ForTune does not require a treatment set. The process of matching is improved upon using Entropy maximization in [7, 16].

Both IWS and PSM are used in *counterfactual analysis*, a crucial research area for hypothesis testing, offline evaluation, and learning in multi-armed bandits and Reinforcement Learning (RL) exploration policies [23]. *Off-Policy Evaluation (OPE)*, a large subset of counterfactual analysis, estimates outcomes from deploying a target policy to a population from which data has already been collected. OPE uses existing data collected based on a logging policy (typically randomization) and re-weights it with propensities given by the target policy to estimate what would happen to quantities of interest if the target policy were to be deployed. Research in OPE includes applications in slate recommendation [26], sequential search [19]), reducing the variance of OPE [5], leveraging historic data [1], minimally-invasive randomized interventions [13], long-term off-policy estimation [22], and interdependent reward models [18].

Approaches to long-term causal inference (LCI) are also relevant to our aims. LCI typically uses short-term metrics as surrogates for long-term effects [2]. Research on LCI focuses on identifying good surrogates [17, 28, 31] or handling confounding factors [27].

Among these methods, only Sensitivity Analysis requires an analytic model of the data. Both IWS and PSM require access to a sample of the target dataset. OPE is designed to evaluate new policies. ForTune however, requires no analytical model and only some global statistics about the target dataset. It is simple, relies on historical data, does not require interventions, does not make assumptions about surrogacy, does not require models of effects, and can be applied to virtually any data, attributes, and metrics.

4 EXPERIMENTS

Testing ForTune poses challenges because it is not feasible to set the treatment branch of a controlled experiment to match a predefined scenario, and without treatment results, there is no direct way to evaluate ForTune predictions. Therefore, we adopted an indirect approach: we use an existing experiment and reverse-engineering a compatible scenario.

In Section 4.1, in the interest of reproducibility, we use a publicly available dataset from Criteo. However, important information in the Criteo dataset was obfuscated by the company to protect user confidentiality, limiting our ability to analyze the results in full details. Consequently, we also use proprietary data from Spotify in Section 4.2.

4.1 Predicting the Probability of Visits on the Criteo Dataset

The `CRITEO-UPLIFT1` dataset [4] was created using data from several incrementality tests, which are randomized trials where a portion of the population is prevented from being targeted by advertising. The dataset contains 25 million rows, each representing a user with twelve features, a treatment indicator, and two binary labels (visits and conversions). Positive labels indicate whether the user visited or converted on the advertiser's website within a two-week test period. The overall treatment ratio is 84.6%, reflecting advertisers' practice of maintaining a small control population to minimize potential revenue loss. For privacy, the data has been selectively sub-sampled and anonymized to protect the benchmark's competitiveness without revealing the original incrementality level or user context. The dataset is freely accessible on the Criteo datasets web page.[2]

The twelve features, named f_0 to f_{11}, come without descriptions or definitions of how they are computed. We only know that they are predictive of the two labels. In the following experiment, we chose visits as the metric of interest rather than conversions because visits are less rare (4.7% compared to 0.3%). This choice helps avoid dealing with extremely skewed data.

In the upper and bottom panes of Figure 2, we present histograms of the probability of visits in the control and treatment branches. We performed bootstrapping by sampling $B = 199$ times with replacement,[3] creating 10, 000 samples from the original dataset and computing the probability of visits for each sample. This procedure provides an estimate of the metric's variability, which is useful for evaluating prediction accuracy.

4.1.1 The Scenario. To estimate the probability of visits in the treatment branch based on the control data, ForTune requires a scenario. This is challenging because Criteo did not provide enough details on how the dataset was generated, and building a scenario typically requires domain knowledge. To address this, we identify a scenario likely to produce the control branch data. Essentially, we ask: if we have an ideal or somewhat ideal scenario, how well does ForTune predict the metrics of interest in the treatment branch?

To keep the experiment simple and representative of a realistic scenario, we impose constraints only on the averages of a subset of features. We identify the features for which the means in the control and treatment branches are significantly different and constrain the weighted averages of this subset to match the corresponding averages in the treatment branch. The results are reported next.

---

[2]https://ailab.criteo.com/ressources
[3]Given that the Criteo dataset contains nearly 14 million observations, sampling with or without replacement is unlikely to make a significant difference.

4.1.2 ForTune Predictions. We apply ForTune to predict the probability of visits in the treatment branch. To establish the constraints, we compute the control set averages and the treatment set averages of the twelve features. Only features f_1, f_4, f_7 and f_{10} have notably different averages, so we design constraints only for these features.[4] Setting these as constraints in Equation (5) leads to the convex optimization problem in Equation (7), where the sum is over the observations in the control set.

$$
\begin{aligned}
\arg\max_{\omega} \quad & \mathcal{H}(\omega) \\
\text{subject to} \quad & \frac{1}{N}\sum_n \omega_n f_{1n} = 17.00, \\
& \frac{1}{N}\sum_n \omega_n f_{4n} = 3.59, \\
& \frac{1}{N}\sum_n \omega_n f_{7n} = -5.43, \quad (7)\\
& \frac{1}{N}\sum_n \omega_n f_{10n} = 23.34, \\
& \omega_n \geq 0 \qquad\qquad n \in \{1\ldots N\}, \\
& \sum_n^N \omega_n = 1 \qquad\quad n \in \{1\ldots N\}
\end{aligned}
$$

Once the weights ω are evaluated, we use them to estimate the probability of visits v_{test} in the treatment set as:

$$
\hat{v}_{\text{test}} = \frac{1}{N}\sum_n \omega_n v_n
$$

$v_n \in \{0, 1\}$ indicates whether the user in observation n made a visit, corresponding to the visit column of the Criteo dataset.

In practice, it is useful to resample, as we did above, to estimate the probability of visits rather than computing a single point estimate based on the entire control dataset. As described, we sample $B = 199$ times $10{,}000$ observations with replacement from the control set. We then run the procedure that implements (7) on each of the 199 data samples. The result is plotted in the pane of Figure 2 titled ForTune. While the predicted probability of visits is overestimated, the ForTune and treatment histograms overlap.

4.1.3 Nearest Neighbor Matching (NNM). ForTune only uses the treatment set averages, but suppose instead that the full treatment branch dataset was available to us.[5] In this hypothetical case, we should achieve better or equal accuracy compared to using ForTune because we have more information about the treatment. This task is amenable to Nearest Neighbor Matching (NNM), also known as greedy matching.

NNM is a type of Propensity Score Matching (PSM) [8, 21], a family of methods typically used to identify causal relationships from observational studies. In PSM, a set of weights is evaluated to match the control as closely as possible to the treatment set. ForTune, on the other hand, finds a set of weights that satisfy the

scenario, which is designed to approximate the controlled experiment that produced the treatment set. It therefore makes sense to compare the two methods.

The idea behind NNM is the following. For each individual in the treatment group, it finds the most similar individual (or individuals) in the control group based on a set of observed characteristics — hence the term "nearest neighbors". Here, we use the `matchit` function from the `MatchIt` R package [10], which is based on the propensity score computed using generalized linear model (`glm`).

NNM cannot be used when we only have a scenario like ForTune's. Instead, it requires a full treatment dataset to match the observations. However, we expect it to be more accurate because it utilizes more detailed information, making it an upper bound on what ForTune can achieve.

The results of NNM are reported in the "match" pane of Figure 2. We observe that the histogram in the "match" pane coïncides more closely with the "treatment" pane and sits between ForTune and "treatment". Despite this, ForTune's performance is quite good, considering the simple scenario needed to specify the problem.
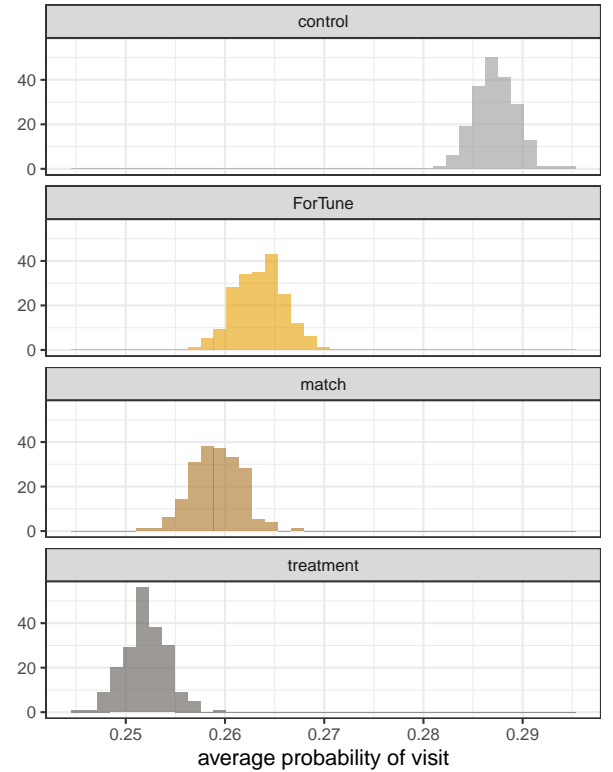

Figure 2: Probabilities of Visit. The "control" and "treatment" panes report the probability of visit in the control and treatment sets. The panes titled "ForTune" and "match" show the estimated probability of visit $\hat{v}_{\text{test}}$ on the treatment set by the respective methods. We observe that even though the "match" predictions align better with the histogram in the "treatment" pane the "ForTune" predictions are quite good.

---

[4]We also ran the experiment using constraints based on the twelve features and the results were similar.
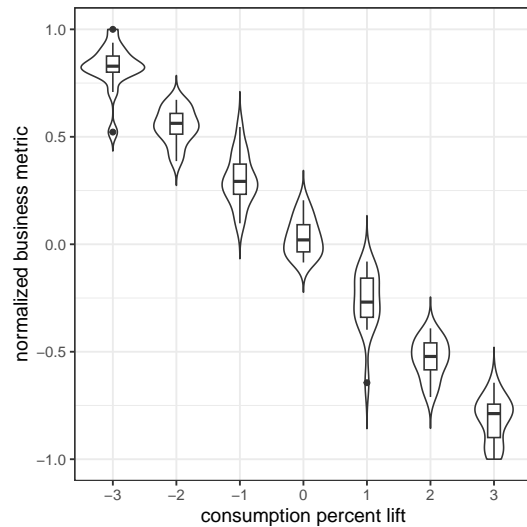[5]Excluding the key metrics we intend to predict using ForTune, otherwise the solution would be trivial.

Figure 3: The business metric is scaled to range between -1 and 1. Consumption distribution is evaluated by bootstrapping (B=50) for each value of the consumption percent lift on the x-axis. The distribution is represented both by a violin plot and a regular box plot. The business metric value is distributed around 0 when the consumption lift is null. The variability results from bootstrapping and gives an estimate of the intrinsic noise in the data.

4.1.4 Analysis. The actual probabilities of visits for control and treatment are reported in the top and bottom panes of Figure 2. Some might be surprised to see that the probability of visits is lower in the treatment group. A possible explanation is that users in the treatment group are targeted less frequently but with more accuracy, leading to both less exposure and higher conversion rates.[6] The predictions from the Matching and ForTune methods are plotted in the two middle panes. As expected, Matching performs better; the histograms generated by the bootstrap runs partially overlap. ForTune underestimates the effect of the treatment, but the direction is correct and the magnitude is relatively close.

Overall, considering that the constraints in Equation (7) provides only a high level description of the actual treatment effects, ForTune's predictions are remarkably close. This result is representative of what we often observe; the predictions are not perfect but point in the right direction, making them useful for informing decision-making.

4.2 ForTune at Spotify

Spotify is among the largest audio streaming services in the world, offering music, podcasts, and audiobooks to users worldwide. As a large business, Spotify consists of many different teams, each with its own key performance metrics, some of which may compete with each other. For instance, one team may prioritize driving conversion to the premium product, while another may focus on long-term

---

[6]Based on a personal conversation with a former Criteo employee.

retention or engagement. These teams run numerous experiments, typically optimizing for one or two metrics.

Historically, one of the most influential areas within the Spotify app is the homepage, where users discover new content and revisit familiar content. Given that changes to the homepage can significantly impact key metrics, we decided to analyze ForTune through a series of experiments conducted on this surface. Specifically, we chose to focus on business metrics, as they are of utmost importance and have historically been less emphasized in favor of easier-to-measure, short-term engagement metrics.

The business estimates reported here are obfuscated to preserve confidentiality. We always apply a monotonic transform to the business estimates and the variables they depend on. Additionally, we only report relative change, not absolute values and normalized them to span the $[-1, 1]$ interval. The different figures report results for various user cohorts, but we do not specify how these cohorts are defined. When we report the obfuscated business metric changes against certain feature values, the definitions of these features are intentionally vague. While these measures protect Spotify's business confidentiality, they do not prevent the analysis from illustrating the potential of the method proposed in this paper.

The methodology is the same as in the previous section: we utilize the control branch data and a scenario to predict the metrics of interest observed in the treatment branch. The main result is that out of 10 monitored metrics from 5 experiments run on Spotify's mobile app homepage, 6 were statistically significant. ForTune estimated mean was directionally aligned 9 times, and the estimated mean histograms overlapped the observed values 8 times. While this is anecdotal evidence, it helped build our confidence in the tool. In Section 4.2.3 we revisit the failed experiment to analyze it in more detail.

We proceed by illustrating representative results obtained from these experiment and how they can be used.

4.2.1 Dealing with Scenario Uncertainty. When designing a scenario, it is often challenging to identify precise values for setting constraints. For example, Spotify can influence user consumption by altering what is surfaced or by changing the user interface. While we can build some knowledge on the extent of these changes, uncertainty about the exact amplitude of the consumption change always remain.

A natural step then, is to evaluate different scenarios where consumption varies across a range of values that reflects this uncertainty. Figure 3 shows the relationships between user consumption and a business metric of interest. In this particular case, the relationship is approximately linear and decreasing. It is also important to note that ForTune is not limited to linear relationships; in other cases, we have observed that the business metric flattens beyond certain values of the control feature.

The information provided by Figure 3 is useful for deciding how to act on specific cohorts and markets. Depending on whether the slope is positive or negative in a given market, the company might apply different strategies. If the slope is close to zero for business metrics like user retention, user satisfaction, revenue, expenses, this might suggest deprioritizing a project in favor of another one with more promising outcomes.
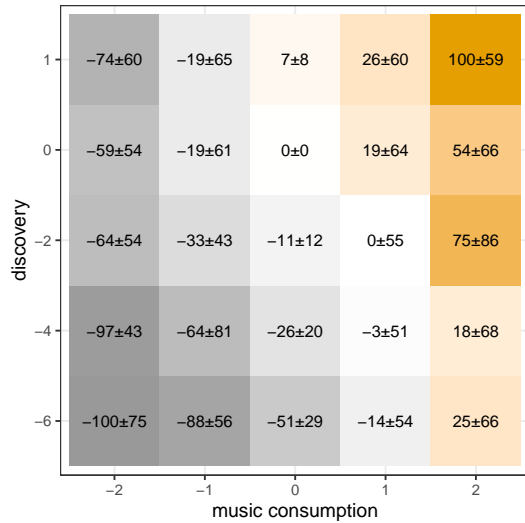
Figure 4: Scaled User Satisfaction. Estimation of user satisfaction in relation to music consumption and discovery of new content based on 50 bootstraps.
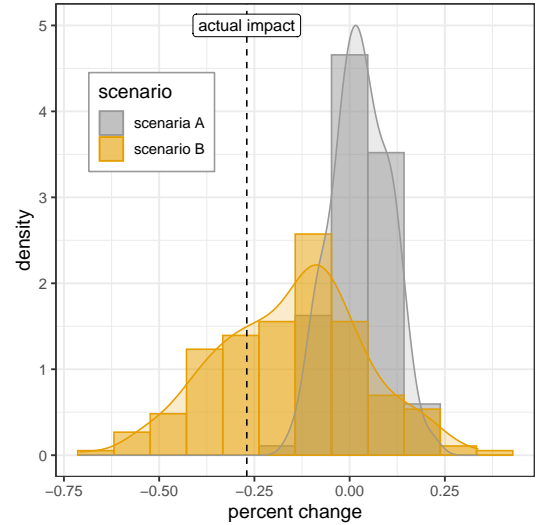

Figure 5: Distribution of the estimations given by two different scenarios and comparison with the true value. Adding more constraints shifted the distribution of predictions for each bootstrap and made the median of the distribution closer to the actual value.

4.2.2 Exchange Rates. The need to trade off between two metrics naturally arises at many decision points. For example, podcast consumption might compete with music consumption for user time, although increase consumption of one type might expose the user to more opportunities to consume the other type. Predicting which of these opposing effects will dominate is often difficult, especially when the control variables vary in intensity; the trade-offs between podcast and music consumption could differ for users with light versus heavy consumption.

We can use ForTune to quantify the impact of such trade-offs on business metrics of interest and introduce the concept of an exchange rate between control variables. The objective is to answer the question: for a fixed value of the business metric, how much must podcast consumption change to compensate for a change in music consumption?

Such trade-offs are ubiquitous in complex applications. In web search, advertisements compete with organic search results and the diversity of the result list competes with recall. At Spotify, the discovery of new music competes with users' desire for familiar content, and popular content competes with more personalized niche content. In the example below, we examine the trade-offs between music consumption and the discovery of new content on a metric akin to user satisfaction.

We run a ForTune experiment with bootstrapping ($B = 50$) over a grid of music consumption and discovery values, and report the associated business metric in Figure 4. Each cell displays the median of the business metrics and the empirical standard deviation. We observe a clear trend where user satisfaction increases with discovery and with music consumption, even though it is not statistically significant. The figure also suggests that a decrease in the discovery rate below -4 is associated with a more severe decrease in user satisfaction, although this effect is mitigated by an increase in music consumption.

This technique, while still being refined, is used at Spotify to inform decisions about top-level targets for company strategies related to content discovery and user satisfaction.

4.2.3 Example of a Miss-Specified Scenario. We examine a case where ForTune failed to provide a prediction directionally aligned with the treatment branch outcome. We demonstrate how this scenario was improved by adding more constraints, emphasizing the importance of how the scenario is defined, as noted in Section 2.4.

In this example, a new ranking function for podcasts was tested on Spotify's homepage. The experiment showed a negative, statistically significant impact on two business metrics related to consumption and subscriptions. The control metrics used in the experiment were podcast consumption and user activity on the homepage. Based on this scenario, the tool failed to predict the negative impact on the two business metrics. Instead, it predicted an increase.

After a thorough analysis, we added constraints to the features related to the consumption of other content types and subscription behavior. This not only led to a more accurate prediction, but also prompted the experiment designers to revise their hypotheses regarding the treatment effect and the specification of control metrics during the experiment. Figure 5 shows histograms of the predictions before and after adding the extra constraints (Scenarios A and B, respectively). The figure also reports the target metric value observed in the treatment.

CONCLUSION

The limitations of online testing and our desire to produce useful insights for product decisions led us to develop ForTune, a flexible, lightweight, and inexpensive approach to investigating hypotheses

about changes in consumption behavior, business metrics, and trade-offs between the two. ForTune is an offline, model-free solution that can explore many hypotheses at low cost, providing powerful support for product leaders making key decisions.

However, like most technological advances, ForTune comes with trade-offs. It is important to emphasize that not all of ForTune's predictions will be accurate or precise, although better domain knowledge and experience help build higher confidence. We expect predictions to have a large variance to accommodate the uncertainty inherent in the vague and under-specified scenarios for which the tool is intended. These scenarios are typically described by a limited number of simple constraints, such as a new average or a new proportion, without addressing the causality of relationships among constraints or user behaviors. ForTune is best used to identify trends in trade-offs, extrapolate changes in consumption due to algorithmic changes to longer-term business metrics, and generate hypotheses for deeper analysis.

Nevertheless, teams at Spotify are using the tool to generate insights that have previously eluded many teams and leads. These insights have been crucial in making several key product decisions, providing decision-makers with an understanding of relationships between key metrics. We intend to continue expanding the tool by adding new diagnostics, particularly measures derived from information theory, and extending it to end-to-end offline evaluation by linking it to off-policy estimation of consumption shifts and projections of user growth.

REFERENCES

[1] Aman Agarwal, Soumya Basu, Tobias Schnabel, and Thorsten Joachims. 2017. Effective Evaluation Using Logged Bandit Feedback from Multiple Loggers. *KDD* (2017), 687–696.
[2] Susan Athey, Raj Chetty, Guido W Imbens, and Hyunseung Kang. 2019. *The surrogate index: Combining short-term proxies to estimate long-term treatment effects more rapidly and precisely*. Technical Report. National Bureau of Economic Research.
[3] Zdravko Botev and Ad Ridder. 2017. Variance reduction. *Wiley statsRef: Statistics reference online* (2017), 1–6.
[4] Diemert Eustache, Betlei Artem, Christophe Renaudin, and Amini Massih-Reza. 2018. A Large Scale Benchmark for Uplift Modeling. In *Proceedings of the AdKDD and TargetAd Workshop, KDD, London,United Kingdom, August, 20, 2018*. ACM. /Users/gdupret/References/2023/20231114T175543--a-large-scale-benchmark-for-uplift-modeling__criteo_scenario.pdf
[5] Miroslav Dudík, John Langford, and Lihong Li. 2011. Doubly Robust Policy Evaluation and Learning. In *Proceedings of the 28th International Conference on International Conference on Machine Learning* (Bellevue, Washington, USA) *(ICML'11)*. Omnipress, Madison, WI, USA, 1097–1104.
[6] Bradley Efron. 2000. The bootstrap and modern statistics. *J. Amer. Statist. Assoc.* 95, 452 (2000), 1293–1296.
[7] Jens Hainmueller. 2012. Entropy balancing for causal effects: A multivariate reweighting method to produce balanced samples in observational studies. *Political analysis* 20, 1 (2012), 25–46.
[8] M.A. Hernan and J.M. Robins. 2023. *Causal Inference: What If*. CRC Press. https://books.google.com/books?id=_KnHIAAACAAJ
[9] Miguel A Hernán and James M Robins. 2010. Causal inference: What If.
[10] Daniel E. Ho, Kosuke Imai, Gary King, and Elizabeth A. Stuart. 2011. MatchIt: Nonparametric Preprocessing for Parametric Causal Inference. *Journal of Statistical Software* 42, 8 (2011), 1–28. doi:10.18637/jss.v042.i08
[11] Daniel G Horvitz and Donovan J Thompson. 1952. A generalization of sampling without replacement from a finite universe. *Journal of the American statistical Association* 47, 260 (1952), 663–685.
[12] Bertrand Iooss and Paul Lemaître. 2015. A review on global sensitivity analysis methods. *Uncertainty management in simulation-optimization of complex systems: algorithms and applications* (2015), 101–122.
[13] Thorsten Joachims, Adith Swaminathan, and Tobias Schnabel. 2017. Unbiased Learning-to-Rank with Biased Feedback. In *Proceedings of the 10th ACM International Conference on Web Search and Data Mining*. 781–789.
[14] Ron Kohavi, Alex Deng, Brian Frasca, Toby Walker, Ya Xu, and Nils Pohlmann. 2013. Online controlled experiments at large scale. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. 1168–1176.
[15] Genyuan Li, Herschel Rabitz, Paul E Yelvington, Oluwayemisi O Oluwole, Fred Bacon, Charles E Kolb, and Jacqueline Schoendorf. 2010. Global sensitivity analysis for systems with independent and/or correlated inputs. *The journal of physical chemistry A* 114, 19 (2010), 6022–6032.
[16] Sicheng Lin, Meng Xu, Xi Zhang, Shih-Kang Chao, Ying-Kai Huang, and Xiaolin Shi. 2023. Balancing Approach for Causal Inference at Scale. arXiv:2302.05549 [stat.ME]
[17] Thomas M McDonald, Lucas Maystre, Mounia Lalmas, Daniel Russo, and Kamil Ciosek. 2023. Impatient Bandits: Optimizing Recommendations for the Long-Term Without Delay. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 1687–1697.
[18] James McInerney, Brian Brost, Praveen Chandar, Rishabh Mehrotra, and Benjamin Carterette. 2020. Counterfactual Evaluation of Slate Recommendations with Sequential Reward Interactions. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1779–1788.
[19] Dadong Miao, Yanan Wang, Guoyu Tang, Lin Liu, Sulong Xu, Bo Long, Yun Xiao, Lingfei Wu, and Yunjiang Jiang. 2021. Sequential Search with Off-Policy Reinforcement Learning. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management*.
[20] Paul R Rosenbaum and Donald B Rubin. 1983. The central role of the propensity score in observational studies for causal effects. *Biometrika* 70, 1 (1983), 41–55.
[21] Donald B Rubin. 1973. Matching to remove bias in observational studies. *Biometrics* (1973), 159–183.
[22] Yuta Saito, Himan Abdollahpouri, Jesse Anderton, Ben Carterette, and Mounia Lalmas. 2024. Long-term Off-Policy EvaluationandLearning. In *Proceedings of the 2024 ACM Web Conference (to appear)*.
[23] Yuta Saito and Thorsten Joachims. 2021. Counterfactual Learning and Evaluation for Recommender Systems: Foundations, Implementations, and Recent Advances. In *Proceedings of the 15th ACM Conference on Recommender Systems*. 828–830.
[24] A. Saltelli, K. Chan, and E.M. Scott. 2009. *Sensitivity Analysis*. Wiley. https://books.google.com/books?id=gOcePwAACAAJ
[25] Marco Scutari. 2010. Learning Bayesian Networks with the bnlearn R Package. *Journal of Statistical Software* 35, 3 (2010), 1–22. doi:10.18637/jss.v035.i03
[26] Adith Swaminathan, Akshay Krishnamurthy, Alekh Agarwal, Miro Dudik, John Langford, Damien Jose, and Imed Zitouni. 2017. Off-Policy Evaluation for Slate Recommendation. In *Advances in Neural Information Processing Systems*, Vol. 30. 3632–3642.
[27] Graham Van Goffrier, Lucas Maystre, and Ciarán Gilligan-Lee. 2023. Estimating long-term causal effects from short-term experiments and long-term observational data with unobserved confounding. *arXiv preprint arXiv:2302.10625* (2023).
[28] Yuyan Wang, Mohit Sharma, Can Xu, Sriraj Badam, Qian Sun, Lee Richardson, Lisa Chung, Ed H Chi, and Minmin Chen. 2022. Surrogate for long-term user experience in recommender systems. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 4100–4109.
[29] Herbert Weisberg, Jon A Krosnick, and Bruce D Bowen. 1996. *An introduction to survey research, polling, and data analysis*. Sage.
[30] Chonggang Xu and George Zdzislaw Gertner. 2008. Uncertainty and sensitivity analysis for models with correlated parameters. *Reliability Engineering & System Safety* 93, 10 (2008), 1563–1573.
[31] Jeremy Yang, Dean Eckles, Paramveer Dhillon, and Sinan Aral. 2020. Targeting for long-term outcomes. *arXiv preprint arXiv:2010.15835* (2020).