Improving Human-Algorithm Collaboration: Causes and Mitigation of Over- and Under-Adherence

Maya Balakrishnan
Harvard Business School, maya@hbs.edu

Kris Ferreira
Harvard Business School, kferreira@hbs.edu

Jordan Tong
Wisconsin School of Business, jordan.tong@wisc.edu

February 29, 2024

Even if algorithms make better predictions than humans on average, humans may sometimes have private information which an algorithm does not have access to that can improve performance. How can we help humans effectively use and adjust recommendations made by algorithms in such situations? When deciding whether and how to override an algorithm's recommendations, we hypothesize that people are biased towards following a naïve advice weighting (NAW) heuristic: they take a weighted average between their own prediction and the algorithm's, with a constant weight across prediction instances, regardless of whether they have valuable private information. This leads to humans over-adhering to the algorithm's predictions when their private information is valuable and under-adhering when it is not. In an online experiment where participants are tasked with making demand predictions for 20 products while having access to an algorithm's predictions, we confirm this bias towards NAW and find that it leads to a 20-61% increase in prediction error. In a second experiment, we find that feature transparency – even when the underlying algorithm is a black box – helps users more effectively discriminate how to deviate from algorithms, resulting in a 25% reduction in prediction error. We make further improvements in a third experiment via an intervention designed to get users to move away from advice weighting heuristics altogether and instead use only their private information to inform deviations, leading to a 34% reduction in prediction error.

Key words: human-algorithm interaction, forecasting, behavioral operations, algorithm transparency, advice taking

1. Introduction

Organizations are seeking to incorporate data-driven algorithms into their decision making processes. PricewaterhouseCoopers (2022) reports that 86% of executives consider AI algorithms a "mainstream technology," with 74% believing these algorithms would add value to their companies by, for example, improving operations, marketing, and HR decisions. However, despite the promise of algorithms and their widespread adoption, implementations are not always successful. In fact, only 10% of companies that adopted advanced algorithms like AI report seeing significant financial gains, according to Ransbotham et al. (2020). When algorithms show significant promise in simulations and on historical data, but have disappointing performance once implemented, blame is commonly aimed at the humans that use the algorithms. This blame is sometimes warranted: people often have the power to override algorithm recommendations, and there are many examples of such overrides degrading performance (see §2.1).

Based on such disappointing examples, it can be tempting to take the perspective that humans are a *barrier* to achieving the benefits of algorithms: If algorithm aversion causes people to override superior algorithms, then shouldn't the goal be to get people to trust algorithms more, or even remove them from the process altogether? Indeed, finding ways to increase people's adherence to and trust in algorithms has been the focus of substantial research (see §2.3). While this response may be appropriate in some cases, in this paper we take a different and more collaborative perspective. Namely, we start with the premise that humans are not merely barriers, but that humans and algorithms have relative strengths and weaknesses. Even when algorithms are superior to humans on average, theoretically, their collaboration should be able to outperform either on their own. In such a setting where this is possible, we seek to better understand: What is it about human overriding behavior that causes them to miss this opportunity? How can we design the interaction between humans and algorithms such that their collaboration is more successful?

To investigate these questions, we focus on prediction tasks (e.g., forecasting demand) where human decision makers are given recommendations in the form of predictions from an algorithm, which they are freely able to override to make a final prediction. We consider key relative strengths of humans and algorithms. Prediction algorithms advance in accuracy and sophistication every year, so we simplify their strength by assuming an algorithm uses its available information optimally. Relative to algorithms, humans are noisy and more limited in information processing power, so what advantages do humans have? One of their most important relative strengths is that humans sometimes have access to *private information*: any information with predictive value that the algorithm does not take into account. There are many examples: A fashion retail manager may know from social media that a product is trending, but such information may not be used by the company's forecasting algorithm (Cui et al. 2018); a doctor may know a patient's surgery is unusually complicated based on how it looks, even if such information is not in the hospital's information system (Ibrahim et al. 2021, Kim and Song 2022); HR managers making hiring decisions interview candidates, though a scoring algorithm may be based only on test scores and resumes (Hoffman et al. 2018). Of course, having valuable private information does not necessarily mean humans will make override decisions that improve upon the algorithm's performance. But, eliminating humans from the process precludes being able to take advantage of such private information.

Focusing our attention on these relative strengths allows us to approach our research questions in the context of an information aggregation problem, which allows us to leverage existing decision analysis and judgment and decision making literature to better theorize how and why overriding behaviors tend to be biased. To develop our theory, we construct a mathematical model that defines the information setting, algorithm's predictions, and best prediction benchmark. We then examine the impact of certain behavioral assumptions about how people make final predictions after seeing an algorithm's recommendation. We theorize that due to cognitive limitations, people are biased towards following a predictable heuristic we call *naïve advice weighting* (NAW). A person who follows NAW arrives at a final prediction by taking a constant weighted average between what the algorithm recommends and what their own prediction would have been without the algorithm. We show mathematically that NAW is suboptimal because it is overly constant: it causes people to over-adhere to the algorithm when they have highly valuable private information and under-adhere to the algorithm when they do not. Moreover, it is suboptimal because the best prediction may not even lie between the algorithm's recommendation and the person's own initial prediction.

Next, we conduct a controlled online experiment that seeks to test the over- and under-adherence pattern predicted by our theory. In Study 1, following training and feedback about the algorithm's and their own performances independently, participants make demand predictions for 20 products with the algorithm's recommendations. The only difference between experimental conditions is that some participants always have high-impact (very valuable) private information, some always have low-impact private information, and some face a mixed set of the two instances. We find that participants who always have low-impact private information generally adhere to the algorithm, while those who always have high-impact private information generally do not. However, consistent with NAW, participants seeing a mixed set adhere to the algorithm to about the same degree across high- and low-impact private information instances, resulting in the predicted over- and under-adherence pattern. This leads to a 20 - 61% increase in prediction error relative to the non-mixed conditions. In summary, participants in the mixed set condition failed to use their private information to collaborate effectively with the algorithm because they couldn't differentiate when their private information was valuable.

Based on these results, we design and test a type of algorithm transparency aimed at mitigating the underlying driver of bias. Specifically, we hypothesize that *feature transparency* – explicit training on the variables that the algorithm takes into account – helps participants address the core problems of NAW by making it easier for them to identify what their private information is, when it warrants a substantial deviation from the algorithm, and in which direction. In Study 2, we compare *feature transparency* to *no transparency*. Using the same mixed set condition from Study 1, Study 2 shows that *feature transparency* indeed helps people detect when they should adhere more or less to the algorithm, resulting in a 25% reduction in prediction error over *no transparency*. Study 2 also provides evidence that *feature transparency* helps people deviate from the algorithm in the correct direction more often, even when the correct direction is opposite to that of their initial prediction. We leverage this insight in Study 3 to design an intervention aimed at increasing the effectiveness of *feature transparency* by nudging people away from advice weighting strategies altogether, thus making it more likely that they adjust the algorithm's predictions solely using their key relative strength: their private information. We find that this intervention is highly effective, leading to an additional 21% reduction in prediction error over *feature transparency* alone.

We summarize our main contributions as follows:

1. We define and examine new theory that describes algorithm overriding behavior when people have private information. We propose that people are biased towards a *naïve advice weighting* heuristic, and analyze how it degrades human-algorithm collaborative performance.
2. We provide experimental evidence supporting how, consistent with NAW, a predictable over- and under-adherence pattern emerges depending on the value of people's private information. We illustrate that the cost of these biases can be significant.
3. We design and experimentally test *feature transparency* as an implementable mitigation approach that can help people better identify and use their private information. We achieve further improvements by nudging humans away from advice weighting heuristics and instead towards using only their private information to adjust algorithmic recommendations.

2. Literature Review

2.1. Algorithm Overriding

Do human overrides to algorithm predictions help or hurt in practice? Field evidence from a variety of business contexts provides several examples of overriding degrading performance: doctor overrides of task-scheduling algorithms decreased productivity (Ibanez et al. 2018), retail store manager overrides of price markdown algorithms decreased revenue (Caro and Saez de Tejada Cuenca 2022), and auto-part manager overrides of SKU phase-out algorithms decreased profits (Kesavan and Kushwaha 2020). Laboratory experiments provide further examples (e.g., see Snyder et al.

2022, Lehmann et al. 2022). Of course, human overrides don't *always* degrade performance. For example in Fildes et al. (2009), forecaster overrides improved accuracy on average in 3 out of 4 supply chain companies investigated. Moreover, even in settings where human overrides degrade performance on average, they may improve performance on predictable subsets of situations. For example, in Kesavan and Kushwaha (2020), although overrides hurt profits on average, they improve profits for growth-stage products. Similarly, for demand forecasters in Khosrowabadi et al. (2022), algorithm overrides didn't improve accuracy on average, but did help for expensive and non-fresh products. In general, these field studies point to the importance of understanding *how* people make override decisions so that we can help humans override in such a way that consistently yields improvement over the algorithm alone.

Relatedly, there is diverse research that suggests a variety of possible *psychological* factors contributing to why people override algorithms. For example, even if an algorithm performs well, people may be averse to feeling like they don't understand how its process works (Yeomans et al. 2019). They might feel like the task is too subjective for an algorithm (Castelo et al. 2019). People may be more tolerant of their own mistakes relative to an algorithm's (Dietvorst et al. 2015). They may also perceive an algorithm as too simple or too complex (Lehmann et al. 2022). In contrast to this body of work on psychological reasons for overriding, we focus on a setting with a purely *rational* reason for overriding: people may have access to private information that the algorithm does not use.

2.2. Aggregation Strategies

How *should* one aggregate an algorithm's predictions with a human who has private information? Such a prescriptive question belongs to a broader area of research on judgment aggregation strategies, which has addressed this question primarily in the context of aggregating multiple human judgments. A main finding is that simple averaging strategies work surprisingly well (e.g., Clemen 1989, Blattberg and Hoch 1990, Surowiecki 2005). However, substantial improvements over simple averaging can be achieved when people have shared information. Recent research has developed strategies for combining judges' predictions to address this shared information problem, for example by asking an additional question that can help an algorithm infer the amount of shared information (e.g., Palley and Soll 2019). Other strategies include strategic identification of experts and upweighting their predictions (e.g., Soule et al. 2023). Our work similarly seeks to understand and address the shared information problem. However, unlike most of the above papers, it addresses the problem between an algorithm and a human. A closely-related exception is Ibrahim et al. (2021), who study how to address the shared information problem between humans and algorithms in a different setting than ours, where the algorithm has final decision authority.

In a review paper, Arvan et al. (2019) outline two paradigms of how human and algorithm forecasts can be combined to produce superior forecasts. In one paradigm, an algorithm makes the final forecast after receiving a human's prediction as an input. Several papers study how best to elicit forecasts from humans and how to use them as inputs to algorithms (e.g., Ball and Ghysels 2018, Flicker 2018, Ibrahim and Kim 2019, Ibrahim et al. 2021, Brau et al. 2023). An alternate paradigm, which we study, is when a human makes the final forecast (e.g., Luong et al. (2020), though their focus is on binary – rather than continuous – prediction tasks). Such a human-oversight policy is extremely common in practice and is even required across many settings for both legal and ethical reasons (Green 2022).

This second paradigm is related to a broader stream of literature on how people make final decisions when they receive a recommended prediction from an external advisor. A common model used to study this question is a Judge-Advisor System (JAS), which can be applied to human advisors (e.g., Bonaccio and Dalal 2006, Soll et al. 2021) and algorithmic advisors (Logg et al. 2019, Lehmann et al. 2022). In JAS, a human judge first forms their initial prediction, then they receive a prediction from an advisor, and then they make a final prediction. The Weight on Advice (WOA) metric refers to where the final prediction lies on the interval between the initial prediction and the advice; a WOA of 0 represents ignoring the advice, while a WOA of 1 represents completely taking the advice. Many papers report on various factors that impact WOA, but all typically report very high rates (often over 95%) of what we call "advice weighting" behavior, where people's final predictions are a weighted average of their initial prediction and the advice (e.g., Soll and Larrick (2009), Gino and Moore (2007), Logg et al. (2019)). We examine how people take the advice of an algorithm and also report WOA measurements. However, unlike most papers in this stream of work, we examine settings in which the participant has objective and measurable reasons to deviate from the algorithm's advice that may vary across instances. Moreover, we examine how participants can improve their performance by deviating from an advice weighting approach.

Beyond the aggregation strategies detailed above, numerous other strategies for integrating human and algorithm predictions have been proposed for different settings. For example, some strategies use algorithms to provide actionable tips (Bastani et al. 2022). Others use humans only to select which algorithm to use from a set of models (Petropoulos et al. 2018). Other strategies include delegating instances to either a human or algorithm (Fügener et al. 2022), having algorithms and humans work sequentially across instances (Beer et al. 2022), and having algorithms suggest predictions for humans to choose from (Rios et al. 2022).

2.3. Algorithm Transparency

How do you design algorithms to mitigate end-user biases? Though there are a variety of strategies, our paper focuses on providing algorithm transparency. There are several types of transparency. For example, transparency can refer to post-hoc explanations for why a specific prediction was made (Lipton 2017). In contrast, the type of transparency we consider is an ex ante form of algorithm transparency, where aspects of the model are described to the user. Comprehensive ex ante transparency could allow people to theoretically fully *simulate* the algorithm's predictions (Lage et al. 2019). One can also provide transparency into certain components, such as how the model was trained (Anik and Bunt 2021, Gebru et al. 2021). These types of component transparency have been advocated and implemented by industry leaders like Google and IBM (Hind et al. 2020, Gebru et al. 2021, Mitchell et al. 2019), and are the types we consider in our Study 2. However, unlike the above papers, we make precise predictions about how private information interacts with human cognitive biases to make different types of transparency help or hurt in predictable circumstances.

Commonly, algorithm transparency is provided in an attempt to increase end-user trust, which can be subjectively measured (e.g., Likert scale as in Cadario et al. 2021) or observed (e.g., algorithm use rate as in Yin et al. 2019). Nevertheless, there is not uniform evidence that algorithm transparency increases trust. Effects vary by algorithm or user characteristics: Lehmann et al. (2022) find that whether comprehensive transparency increases trust depends on the perceived complexity of the model, and Bolton et al. (2022) show that recommendation uncertainty transparency has heterogeneous effects depending on users' levels of numeracy. Also, increased trust does not always lead to better outcomes, as people may suffer from information overload (Poursabzi-Sangdeh et al. 2021) or overly trust the algorithm when they shouldn't (Lakkaraju and Bastani 2020). We contribute to these papers by identifying a type of transparency that increases trust when the algorithm should be superior and decreases trust otherwise.

3. Theory Development

3.1. Model Setting, Definitions, and Assumptions

Consider a setting where outcome Y_i is a function of public feature vector $\boldsymbol{X}_i^{pub}$, private feature vector $\boldsymbol{X}_i^{priv}$, and mean-zero random noise ϵ_i for each instance $i = 1, 2, ..., n$, i.e.,

$$Y_i = f_{actual}(\boldsymbol{X}_i^{pub}, \boldsymbol{X}_i^{priv}) + \epsilon_i \; . \tag{1}$$

We assume $\{(\boldsymbol{X}_i^{pub}, \boldsymbol{X}_i^{priv}, \epsilon_i)\}_{i=1,..,n}$ are independent and identically distributed. (However, $\boldsymbol{X}_i^{pub}$ and $\boldsymbol{X}_i^{priv}$ can be dependent.)

The human decision-maker is tasked with predicting outcome Y_i given a realization of the feature vector $(\boldsymbol{x}_i^{pub}, \boldsymbol{x}_i^{priv})$. We model human j's prediction without seeing an algorithm, which we call their *initial prediction*, for instance i as:

$$\hat{Y}_{ij}^{init} = f_{init,j}(\boldsymbol{X}_i^{pub}, \boldsymbol{X}_i^{priv}) + \eta_{ij} \ . \tag{2}$$

Here, $f_{init,j}$ may differ from f_{actual}, reflecting the idea that humans may use feature information in a systematically incorrect manner. Also, η_{ij} is a zero-mean, random variable, reflecting the idea that even given identical information, humans may be noisy (e.g., Su 2008, Kahneman et al. 2022).

We consider the situation where the decision-maker has access to an algorithm that they can use to help predict Y_i. The algorithm uses only public features as inputs. Its prediction is:

$$\hat{Y}_i^{alg} = f_{alg}(\boldsymbol{X}_i^{pub}) \ . \tag{3}$$

Finally, we can define the *best prediction benchmark* as:

$$Y_i^* = f_{actual}(\boldsymbol{X}_i^{pub}, \boldsymbol{X}_i^{priv}). \tag{4}$$

The gap between the best prediction benchmark Y_i^* and the algorithm's prediction $\hat{Y}_i^{alg}$ represents the potential improvement that the human could theoretically make over the algorithm's prediction, which could come from either (i) the use of private features $\boldsymbol{X}_i^{priv}$ for which the algorithm does not have access, and/or (ii) better use of public features $\boldsymbol{X}_i^{pub}$. For ease of exposition and in line with our experiments, we assume that the algorithm optimally uses $\boldsymbol{X}_i^{pub}$ when making its predictions, and thus any improvement that the human can make over the algorithm's prediction is due to their use of $\boldsymbol{X}_i^{priv}$. With this in mind, we define the *impact of private features* as follows:

Definition 1. *The impact of private features for feature vector* $(\boldsymbol{X}_i^{pub}, \boldsymbol{X}_i^{priv})$ *is*

$$V_i \triangleq Y_i^* - \hat{Y}_i^{alg} = f_{actual}(\boldsymbol{X}_i^{pub}, \boldsymbol{X}_i^{priv}) - f_{alg}(\boldsymbol{X}_i^{pub}).$$

Since realization v_i can be positive or negative, we will use the squared impact of private features $(V_i)^2$ as a measure of the impact in an absolute sense. The following two assumptions are sufficient, but not always necessary for our results.

Assumption 1. *The human's initial prediction and the algorithm's forecasts are unbiased, i.e.,* $\mathbb{E}[\hat{Y}_i^{init} - Y_i] = \mathbb{E}[\hat{Y}_i^{alg} - Y_i] = 0.$

Note that the expectation is over random variables $\boldsymbol{X}_i^{pub}, \boldsymbol{X}_i^{priv}, \eta_{ij}$, and ϵ_i. It does *not* imply that the human and algorithm are unbiased for every instance i; rather, it implies that the human and algorithm will not be systematically too high or low over a large number of instances.

Assumption 2. *The human's initial prediction error* $\hat{Y}_i^{init} - Y_i$ *and the algorithm's prediction error* $\hat{Y}_i^{alg} - Y_i$ *are independent.*

Assumption 2 states that the human's initial prediction accuracy does not provide information about the algorithm's accuracy, and vice versa.

3.2. Behavioral Model of Advice Weighting

How do humans with access to an algorithm's prediction make a final prediction? Following the advice weighting literature (see §2.2), we hypothesize that humans tend to take a weighted average of their initial prediction and the algorithm's prediction to make a final prediction:

$$\hat{Y}_{ij}^{final} = \lambda_{ij}\hat{Y}_i^{alg} + (1-\lambda_{ij})\hat{Y}_{ij}^{init}, \tag{5}$$

where $\lambda_{ij} \in [0,1]$ is the weight that human j places on the algorithm's prediction. For brevity, we omit subscript j when the context is clear. A larger (smaller) value of λ_i means that the human places more (less) weight on the algorithm's prediction.

One can see that the human's final prediction accuracy depends, in part, on how they choose the value of λ_i. We believe humans tend to be biased towards *naïve advice weighting* (NAW), which we define as placing the same weight on the algorithm's prediction, regardless of feature vector $(\boldsymbol{x}_i^{pub}, \boldsymbol{x}_i^{priv})$, i.e. $\lambda_i = \lambda$ for each instance i. Naïve advice weighting is suboptimal for two reasons:

- First, differential weights are not applied across instances with different impacts of private features. We examine this driver of suboptimality in §3.3. To do so, we contrast naïve advice weighting with *sophisticated advice weighting* – where differential weights are applied depending on feature vector $(\boldsymbol{x}_i^{pub}, \boldsymbol{x}_i^{priv})$.
- Second, advice weighting in general – even the best possible sophisticated advice weighting strategy – is suboptimal when Y_i^* cannot be characterized as a convex combination of $\hat{Y}_i^{init}$ and $\hat{Y}_i^{alg}$. We examine this driver of suboptimality in §3.4.

Our results will guide our experimental designs and hypotheses which we preview in §3.5.

3.3. Naïve Advice Weighting is Suboptimal Because the Weights are Overly Constant

We now show that a sophisticated advice weighting strategy that implements non-constant weights can outperform even the best possible naïve advice weighting strategy, found by solving

$$NAW: \min_{\lambda\in[0,1]} \mathbb{E}\big[\big(Y_i - (\lambda\hat{Y}_i^{alg} + (1-\lambda)\hat{Y}_i^{init})\big)^2\big]. \tag{6}$$

NAW finds the constant weight on algorithm that minimizes the expected squared error of the final prediction, $\hat{Y}_i^{final}$. Practically, it could be approximated by solving for the λ that minimizes the sum of squared errors over a large historical dataset of realizations $\hat{y}_i, \hat{y}_i^{alg}, \hat{y}_i^{init}$. The follow proposition characterizes the optimal weight on the algorithm's predictions under NAW.

Proposition 1. *Under naïve advice weighting, the optimal weight-on-algorithm λ^{NAW} is*

$$\lambda^{NAW} = \frac{\mathbb{E}[(Y_i - \hat{Y}_i^{init})^2]}{\mathbb{E}[(Y_i - \hat{Y}_i^{init})^2] + \mathbb{E}[(Y_i - \hat{Y}_i^{alg})^2]}.$$

The optimal weight on algorithm under NAW is intuitively appealing: it puts more weight on the algorithm if it has smaller expected squared errors relative to the human's initial forecasts. Empirical evidence also suggests that humans generally weight advice in a similar manner: they increase their weight on advice as the advisor's accuracy increases (Harvey and Fischer 1997, Yaniv and Kleinberger 2000, Sniezek and Van Swol 2001, Sah et al. 2013, Soll et al. 2021).

Although intuitively appealing, naïve advice weighting is suboptimal in part because the weight is constant, but the algorithm's accuracy is not. Intuitively, the human could use $(\boldsymbol{x}_i^{pub}, \boldsymbol{x}_i^{priv})$ to predict when the algorithm will be more or less accurate and put more weight on the algorithm when they expect it to be more accurate. To formalize this intuition, we define a *sophisticated advice weighter* as one who first categorizes feature vectors into groups with different average impacts of private features, and then calculates a different weight on algorithm for each group. (We consider only 2 groups, but the analyses could be extended to more.) Let $\mathcal{S}$ denote the domain of $(\boldsymbol{X}_i^{pub}, \boldsymbol{X}_i^{priv})$. A sophisticated advice weighter can partition $\mathcal{S}$ into sets $\mathcal{S}_L, \mathcal{S}_H$ such that

$$E[(V_i)^2|(\boldsymbol{X}_i^{pub}, \boldsymbol{X}_i^{priv}) \in \mathcal{S}_L] < E[(V_i)^2|(\boldsymbol{X}_i^{pub}, \boldsymbol{X}_i^{priv}) \in \mathcal{S}_H].$$

Thus, $\mathcal{S}_L$ has a lower expected squared impact of private features, while $\mathcal{S}_H$ has a higher expected squared impact of private features. Given such partitions, a sophisticated advice weighter then solves the following problem to determine the optimal weights, λ_L^{SAW} and λ_H^{SAW}, in each partition:

$$SAW: \min_{\lambda_L, \lambda_H \in [0,1]} \sum_{k=L,H} \mathbb{E}\big[\big(Y_i - (\lambda_k \hat{Y}_i^{alg} + (1-\lambda_k)\hat{Y}_i^{init})\big)^2 |(\boldsymbol{X}_i^{pub}, \boldsymbol{X}_i^{priv}) \in \mathcal{S}_k\big]\mathbb{P}((\boldsymbol{X}_i^{pub}, \boldsymbol{X}_i^{priv}) \in \mathcal{S}_k). \tag{7}$$

Proposition 2. *Under sophisticated advice weighting, the optimal weights λ_L^{SAW} and λ_H^{SAW} satisfy $\lambda_H^{SAW} < \lambda^{NAW} < \lambda_L^{SAW}$. Furthermore, $OPT^{SAW} \le OPT^{NAW}$, where OPT^{SAW} and OPT^{NAW} represent the optimal squared errors of SAW and NAW, respectively.*

The first part of Proposition 2 shows that the sophisticated advice weighter adheres more to the algorithm in the partition where the expected squared impact of private features is low and adheres less where it is high. Moreover, the naïve advice weighter's optimal weight falls between the sophisticated advice weighter's optimal weights for the two sets. The second part of Proposition 2 states that the naïve advice weighter's failure to differentiate amongst instances where there is less versus more impact of private features degrades its performance. Putting the two parts together, we conclude that the naïve advice weighter under-weights the algorithm when the impact of private features is small, and over-weights the algorithm when the impact of private features is large.

3.4. Naïve Advice Weighting is Suboptimal Because the Best Prediction is Often Outside the Advice-Weighting Region

In addition to being suboptimal due to overly-constant weights, naïve advice weighting – like *any* advice weighting strategy – is suboptimal because the best prediction may not be in the *advice-weighting region*. We define the advice-weighting region as the interval between the algorithm's prediction and the human's initial prediction, which corresponds to $\lambda_i \in [0,1]$. When the best prediction benchmark, y_i^*, does not lie within the advice-weighting region, no choice of $\lambda_i \in [0,1]$ can achieve the best prediction benchmark.

In what follows, we explore how often the best prediction falls in the advice-weighting region.

Definition 2. *Prediction $\hat{Z}_i$ is median-unbiased if* $\mathbb{P}[\hat{Z}_i > Y_i^*] = \mathbb{P}[\hat{Z}_i < Y_i^*]$.

Proposition 3. *Let $\hat{Y}_i^{init}$ and/or $\hat{Y}_i^{alg}$ be median-unbiased and assume* $\mathbb{P}[\hat{Y}_i^{init} = Y_i^*] = \mathbb{P}[\hat{Y}_i^{alg} = Y_i^*] = 0$. *Then, Y_i^* is within the advice-weighting region with probability* $1/2$*, i.e.,*

$$\mathbb{P}\big(\min\{\hat{Y}_i^{alg}, \hat{Y}_i^{init}\} \leq Y_i^* \leq \max\{\hat{Y}_i^{alg}, \hat{Y}_i^{init}\}\big) = \frac{1}{2}.$$

The median-unbiased and zero-probability of perfect predictions assumptions are required to obtain a result of precisely $1/2$. Figure 1 illustrates the general intuition behind the proof. For the best prediction to be inside the advice-weighting region, $\hat{y}^{alg}$ and $\hat{y}^{init}$ must lie on opposite sides of y^*. If $\hat{Y}^{init}$ and $\hat{Y}^{alg}$ are equally likely to err high versus low (and assuming the two errors are independent from Assumption 2), then it is equally likely for them to err on opposite sides as it is for them to err on the same side. Thus, Proposition 3 demonstrates that often it is impossible for humans to achieve the best prediction via any advice weighting strategy.

3.5. Hypotheses & Preview of Experiments

Our first study, presented in §4, is designed to test the key intuition developed from Propositions 1 and 2. To do this, we use three conditions in which participants experience (i) only instances

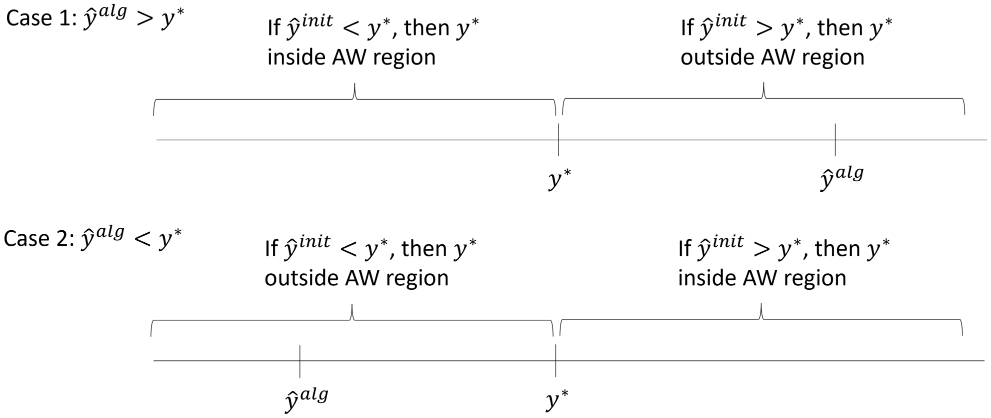

Case 1: $\hat{y}^{alg} > y^*$
If $\hat{y}^{init} < y^*$, then y^* inside AW region
If $\hat{y}^{init} > y^*$, then y^* outside AW region
y^*
$\hat{y}^{alg}$
Case 2: $\hat{y}^{alg} < y^*$
If $\hat{y}^{init} < y^*$, then y^* outside AW region
If $\hat{y}^{init} > y^*$, then y^* inside AW region
$\hat{y}^{alg}$
y^*

Figure 1 **Depiction of when the advice-weighting (AW) region contains y^*.**

with a low impact of private features (analogous to $\mathcal{S}_L$), (ii) only instances with a high impact of private features (analogous to $\mathcal{S}_H$), and (iii) a mixture of instances that contain both low and high impact of private features (analogous to $\mathcal{S}$). In practice, the third condition – where humans occasionally have access to valuable private features – is most common, and we aim to empirically study whether participants in this condition are biased towards naïve advice weighting, and ultimately whether this leads to degradation in prediction accuracy. We give all participants access to historical instances consisting of $(\boldsymbol{x}_k^{pub}, \boldsymbol{x}_k^{priv})$, $\hat{y}_k^{alg}$, and y_k, and then we elicit $\hat{y}_i^{init}$ and $\hat{y}_i^{final}$ for a series of new instances; this allows us to estimate participants' weight on the algorithm.

Hypothesis 1. *Relative to humans who experience instances of only high impact private features or only low impact private features, humans who experience a mix of both cases place a lower weight on the algorithm when the impact of private features is low, and place a higher weight on the algorithm when the impact of private features is high.*

Hypothesis 2. *Relative to humans who experience instances of only high impact private features or only low impact private features, humans who experience a mix of both cases have worse final prediction accuracy for both cases.*

If humans are indeed unable to sufficiently distinguish between instances with low and high impact of private features, we hypothesize that one way to mitigate the overly-constant weights problem of naïve advice weighting would ideally be to tell decision makers which of the information available to them is public vs. private. We study the impact of a practical intervention: providing *feature transparency* – telling humans which features the algorithm does take into account, i.e., which features are public. We believe that feature transparency will mitigate naïve advice weighting behavior by helping humans recognize when they have impactful private features that warrant a substantial deviation from the algorithm, leading to the following two hypotheses that we test in a second study presented in §5.

Hypothesis 3. *Feature transparency increases humans' weight on algorithm when the impact of private features is low, but decreases it when the impact of private features is high.*

Hypothesis 4. *Feature transparency improves human's final prediction accuracy both when the impact of private features is low and when it is high.*

Finally, Proposition 3 suggests that humans often need to deviate from the entire class of advice weighting strategies altogether to achieve the best performance. We hypothesize that if – instead of advice weighting – humans anchor on the algorithm's prediction and make an adjustment to specifically account for only their private features, they may increase their likelihood of going outside the advice-weighting region in a way that improves performance. We test this hypothesis in a third study presented in §6.

Hypothesis 5. *Nudging humans to follow a strategy in which they anchor on the algorithm and adjust based on their private features improves their final prediction accuracy.*

4. Study 1: Uncovering a Naïve Advice Weighting Bias

Study 1 tests Hypotheses 1 and 2 in a controlled online experiment[1].

4.1. Design

4.1.1. Participant Experience Participants are tasked with predicting demand for new products. Each new product i is characterized by two features – "Feature A" and "Feature B" – where Feature A corresponds to x_i^{pub} and Feature B corresponds to x_i^{priv}. The outcome, Y_i, is the actual demand for product i. After predicting demand for several products using only x_i^{pub} and x_i^{priv}, participants are then additionally given an algorithm's demand prediction, $\hat{y}_i^{alg}$, which they can choose if/how to use when making their demand predictions for the remaining products. Notably, participants are not explicitly given $f_{alg}(\cdot)$ or told that the algorithm only uses x_i^{pub} to make its demand predictions. The following sequence of steps provides more details about the participant's experience; select screenshots are included in Appendix E.

1. *Instructions & Comprehension Checks.* Participants are introduced to the demand prediction task and objective of minimizing absolute prediction error, and are tested for comprehension.
2. *Historical Data Review.* Participants view historical data for 20 products, with the option to continue to view more historical data for as many products as they wish. For each product i, they observe x_i^{pub}, x_i^{priv}, and realized (actual) demand y_i.

[1] We pre-registered our sample size, treatment conditions, data exclusion criteria, and planned analyses (see `https://aspredicted.org/CN9_KTK`). All statistical tests reported in the results are pre-registered unless otherwise indicated.

3. *Demand Predictions without Algorithm.* Sequentially for each of $i = 1, ..., 20$ new products not seen in Step 2, participants are given x_i^{pub} and x_i^{priv} and are asked for their demand prediction, $\hat{y}_i^{init}$. After predicting demand for product i, the participant is given the actual demand, y_i, and their absolute prediction error, $|\hat{y}_i^{init} - y_i|$.
4. *Algorithm Introduction.* Participants are informed that an algorithm has been developed to help them predict demand. To give participants experience with the algorithm, they are shown a summary table consisting of the following data for each of the 20 products from Step 3: x_i^{pub}, x_i^{priv}, y_i, $\hat{y}_i^{alg}$, and both the algorithm's and their absolute prediction errors, $|\hat{y}_i^{alg} - y_i|$ and $|\hat{y}_i^{init} - y_i|$.
5. *Demand Predictions with Algorithm.* Sequentially for each of $i = 1, ..., 20$ new products not seen in Steps 2 or 3, participants are first given only x_i^{pub} and x_i^{priv} and are asked for their demand prediction, $\hat{y}_i^{init}$. Then the participant is given the algorithm's demand prediction, $\hat{y}_i^{alg}$, and asked for their final demand prediction, $\hat{y}_i^{final}$. Finally, they are told the actual demand, y_i, as well as both the algorithm's and their absolute prediction errors, $|\hat{y}_i^{alg} - y_i|$ and $|\hat{y}_i^{init} - y_i|$.

Participants' demand predictions in Step 3 ($\hat{y}_i^{init}$) and Step 5 ($\hat{y}_i^{final}$) were incentivized for accuracy: participants received a base compensation of \$7 for completing the experiment (as in Study 2 and 3) plus an additional bonus of \$7 – \$0.15 × (Root Mean Squared Error). A majority of our analyses focuses on Step 5, where participants have access to an algorithm to make their final demand predictions.

4.1.2. Behind the Scenes: Data Generation For product i, actual demand is generated as

$$Y_i = 131 + 1.6X_i^{pub} + 0.75X_i^{priv} + \epsilon_i, \tag{8}$$

where ϵ_i is drawn from a normal distribution with mean 0 and standard deviation 4, X_i^{pub} is drawn from a discrete uniform distribution with support $\{20, 80\}$, and X_i^{priv} is drawn from zero-mean distributions that differ across our three conditions, which will be described in §4.1.3; for all three conditions, X_i^{pub} and X_i^{priv} are independent.

The algorithm's demand prediction is generated as

$$\hat{Y}_i^{alg} = 131 + 1.6X_i^{pub}. \tag{9}$$

From Definition 1, the impact of private feature is $V_i \triangleq Y_i^* - \hat{Y}_i^{alg} = 0.75X_i^{priv}$.

Participants are not explicitly given equations (8) and (9), nor are told how ϵ_i, X_i^{pub}, and X_i^{priv} are generated. That said, because participants have unlimited access to historical data in Step 2, a participant could theoretically recover (8) and achieve the best prediction benchmark.

4.1.3. Conditions Participants are randomly assigned to one of the following three conditions, where the only difference across conditions is the distribution used to generate X_i^{priv}, or equivalently, the impact of private feature, V_i. With slight abuse of language, we use "low" ("high") impact of private feature to describe products with small (large) $|v_i|$.

1. *Always Low Impact of Private Feature (Always Low $|v_i|$).* The values of X_i^{priv} are drawn from a discrete uniform distribution with support {-10, 10}. The impact of private feature is low relative to the other conditions, with $|v_i| \in [0, 7.5]$; this leads to relatively strong algorithm performance.
2. *Always High Impact of Private Feature (Always High $|v_i|$).* The values of X_i^{priv} are drawn from a discrete uniform distribution with support {-150, -50} $\bigcup$ {50, 150}. The impact of private feature is high relative to the other conditions, with $|v_i| \in [37.5, 112.5]$; this leads to relatively poor algorithm performance.
3. *Mixed Impact of Private Feature (Mixed $|v_i|$).* Each value of X_i^{priv} is drawn from a discrete uniform distribution with support {-10, 10} with probability 0.5, and from a discrete uniform distribution with support {-150, -50} $\bigcup$ {50, 150} with probability 0.5. In expectation, this leads to half of the products being identical to products in the first condition where the impact of private feature is low, and the other half of the products being identical to products in the second condition where the impact of private feature is high.

It is helpful to cast our experimental design as a 2×2 mixed design (see Table 1). The first dimension is the impact of private feature, which is either low or high. The second dimension is the participant's exposure set: whether the participant is exposed to a mixture of products with low and high impact of private feature ("mixed" exposure set) or is exposed to only a single impact of private feature – either always low or always high ("single" exposure set).

Table 1 How our three conditions achieve a 2×2 mixed design.

		Exposure Set	
		Mixed	Single
Impact of Private Feature	Low	*Mixed* $\lvert v_i \rvert$	*Always Low* $\lvert v_i \rvert$
	High	*Mixed* $\lvert v_i \rvert$	*Always High* $\lvert v_i \rvert$

We are most interested in studying participants' predictions in the *Mixed* $|v_i|$ condition, as this is reflective of practice where humans occasionally have access to valuable private features. We use participants in the other two conditions to represent benchmarks for sophisticated advice weighters

– humans who make final predictions based on impact of private features – since, by construction, all of their predictions are based on a single exposure set; note that these benchmarks are valid regardless of the sophistication of participants in these two conditions, since for a single exposure set, naïve and sophisticated advice weighting behaviors are approximately equivalent. By comparing across the columns in Table 1, we can identify whether humans experiencing a mixed exposure set can sufficiently distinguish instances with low and high impact of private features – i.e., behave as sophisticated advice weighters – vs. have a bias towards naïve advice weighting.

4.1.4. Dependent Variables We use *median weight on algorithm (MedWOA)* as our dependent variable for Hypothesis 1. We first define participant j's *weight on algorithm* for product $i = 1, ..., 20$ in Step 5 as

$$WOA_{ij} = \min\Big(\max\Big(\frac{\hat{y}_{ij}^{final} - \hat{y}_{ij}^{init}}{\hat{y}_i^{alg} - \hat{y}_{ij}^{init}}, 0\Big), 1\Big). \tag{10}$$

We note that WOA_{ij} is an estimate for λ_{ij} defined for advice weighting behavior in (5), i.e., the weight that the participant places on the algorithm's prediction. Following the advice weighting literature, we note that WOA_{ij} is a winsorized value between zero and one, and we exclude WOA_{ij} if $\hat{y}_i^{alg} = \hat{y}_{ij}^{init}$. Subsequently, we define

$$MedWOA_j^L = \text{median}(WOA_{ij}\ \forall i\ s.t.\ x_i^{priv} \in \{-10, 10\}); \tag{11}$$

$$MedWOA_j^H = \text{median}(WOA_{ij}\ \forall i\ s.t.\ x_i^{priv} \in \{-150, -50\} \cup \{50, 150\}). \tag{12}$$

$MedWOA_j^L$ ($MedWOA_j^H$) can be interpreted as participant j's median value of WOA_{ij} for all products with low (high) impact of private feature. We note that only participants in the *Mixed* $|v_i|$ condition will have values for both $MedWOA_j^L$ and $MedWOA_j^H$, since participants in the other conditions only experience products in a single exposure set.

We use *root median squared error (RMedSE)* as our dependent variable for Hypothesis 2. For each participant j and considering products $i = 1, ..., 20$ in Step 5, we define

$$RMedSE_j^L = \sqrt{\text{median}([\hat{y}_{ij}^{final} - y_i]^2\ \forall i\ s.t.\ x_i^{priv} \in \{-10, 10\})}; \tag{13}$$

$$RMedSE_j^H = \sqrt{\text{median}([\hat{y}_{ij}^{final} - y_i]^2\ \forall i\ s.t.\ x_i^{priv} \in \{-150, -50\} \cup \{50, 150\})}. \tag{14}$$

These *RMedSE* metrics are measures of the participant's prediction accuracy.

4.2. Results

Our analyses include data from 359 participants from Mechanical Turk who successfully passed two initial comprehension checks and completed the full study[2]. By randomly assigning participants across conditions, we had 119 participants in the *Always Low* $|v_i|$ condition, 121 in the *Always High* $|v_i|$ condition, and 119 in the *Mixed* $|v_i|$ condition. The mean time to complete the study was 31.55 minutes, and the mean bonus payment was $1.75.

4.2.1. Weight on Algorithm Results Figure 2 summarizes the results on participants' median weight on algorithm. Recall that participants in the *Always Low* $|v_i|$ and *Always High* $|v_i|$ conditions represent benchmarks for sophisticated advice weighters – humans who make predictions based on impact of private features – since, by construction, all of their predictions are for a single exposure set. As expected, these participants place more weight on the algorithm when they are only exposed to products with a low impact of private feature compared to when they are only exposed to products with a high impact of private feature, since the algorithm performs considerably better for products with a low impact of private feature ($t(237.92) = -12.723, p < 0.0001$).

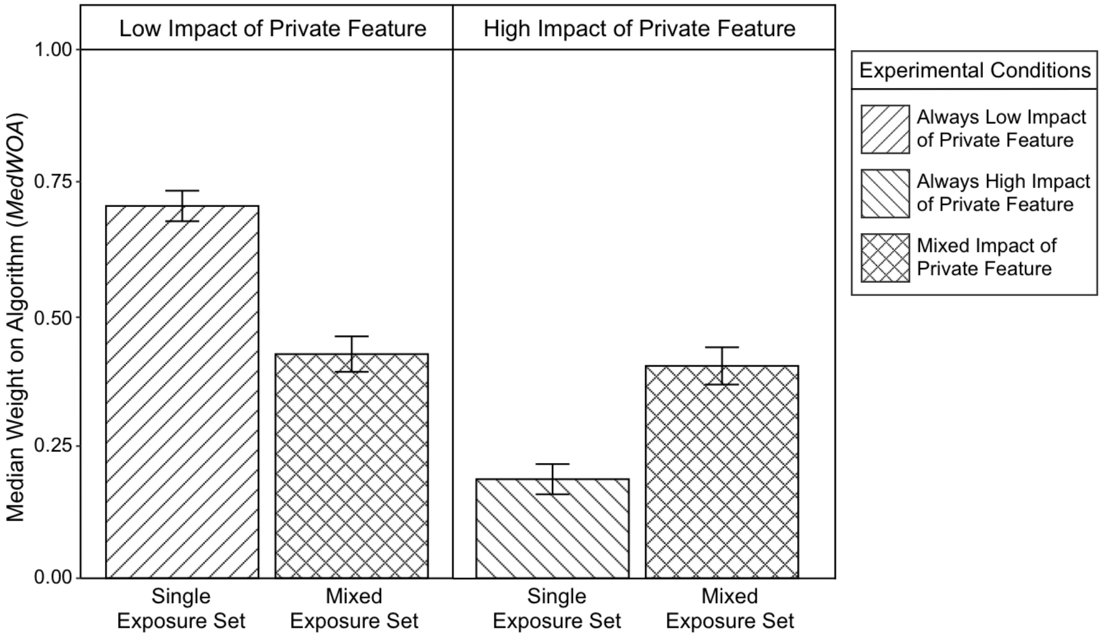


Figure 2 Median weight on algorithm results are averaged (mean) by exposure set, separately for low and high impact of private feature; standard error bars are shown.

Our primary interest is studying behavior of participants in the *Mixed* $|v_i|$ condition. As detailed in Hypothesis 1, we hypothesize that participants in this condition would be unable to sufficiently

[2] 534 MTurkers attempted the study, each with a 99%+ approval rating and 1000+ approvals. Among the 359 participants, 208 were male, 256 had a Bachelor's or advanced degree, 315 were White, and 185 had a yearly household income of $50,000 or more.

distinguish between instances with low and high impact of private features and would instead be biased towards naïve advice weighting. We perform two, one-sided t-tests comparing mean values of *MedWOA* across each row in Table 1. For convenience, let $\mathcal{C}_L$, $\mathcal{C}_H$, and $\mathcal{C}_M$ be the set of participants assigned to the *Always Low* $|v_i|$, *Always High* $|v_i|$, and *Mixed* $|v_i|$ conditions, respectively.

For the first part of Hypothesis 1, we test whether

$$\frac{\sum_{j\in\mathcal{C}_M} MedWOA_j^L}{|\mathcal{C}_M|} \leq \frac{\sum_{j\in\mathcal{C}_L} MedWOA_j^L}{|\mathcal{C}_L|}, \tag{15}$$

i.e., considering only products with a low impact of private feature, whether participants exposed to a mixed exposure set place less weight on the algorithm than participants exposed to a single exposure set. As shown in the left two bars in Figure 2, for low impact of private feature products, participants in the *Mixed* $|v_i|$ condition had a significantly smaller mean *MedWOA*^{L} compared to participants in the *Always Low* $|v_i|$ condition ($t(230.83) = -6.332, p < 0.0001$).

For the second part of Hypothesis 1, we test whether

$$\frac{\sum_{j\in\mathcal{C}_M} MedWOA_j^H}{|\mathcal{C}_M|} \geq \frac{\sum_{j\in\mathcal{C}_H} MedWOA_j^H}{|\mathcal{C}_H|}, \tag{16}$$

i.e., considering only products with a high impact of private feature, whether participants exposed to a mixed exposure set place more weight on the algorithm than participants exposed to a single exposure set. As shown in the right two bars in Figure 2, for high impact of private feature products, participants in the *Mixed* $|v_i|$ condition had a significantly larger mean *MedWOA*^{H} compared to participants in the *Always High* $|v_i|$ condition ($t(227.53) = 4.723, p < 0.0001$).

In addition to conducting t-tests that study behavior on products with low and high impact of private features separately, we can also test how the difference in average $MedWOA$ between products with low and high impact of private features compares across participants who experience a mixed vs. single exposure set. A bias towards naïve advice weighting should result in a smaller difference in average $MedWOA$ for participants experiencing a mixed exposure set, i.e.,

$$\frac{\sum_{j\in\mathcal{C}_M} MedWOA_j^L}{|\mathcal{C}_M|} - \frac{\sum_{j\in\mathcal{C}_M} MedWOA_j^H}{|\mathcal{C}_M|} \leq \frac{\sum_{j\in\mathcal{C}_L} MedWOA_j^L}{|\mathcal{C}_L|} - \frac{\sum_{j\in\mathcal{C}_H} MedWOA_j^H}{|\mathcal{C}_H|}. \tag{17}$$

When we regress the *MedWOA* for each participant on impact of private feature interacted with exposure set, clustering standard errors by participant, we indeed find a significant positive coefficient on the interaction term ($\beta = 0.494, p < 0.0001$). In other words, the difference in average *MedWOA* between products with low vs. high impact of private feature is larger when participants experience a single exposure set than a mixed exposure set; see Appendix B.1 for the full regression table. In fact, we find that for participants who experience a mixed exposure set, their average *MedWOA* is not significantly different for products with a low vs. high impact of private feature ($t(235.44) = -0.459, p = 0.342$).

Our results confirm Hypothesis 1 by showing that participants who experience a mixed exposure set under-weight the algorithm when the impact of private feature is low and over-weight the algorithm when the impact of private feature is high, compared to participants who experience a single exposure set (representing hypothetical sophisticated advice weighters who fully differentiate between the sets). These findings illustrate that humans are biased towards naïve advice weighting, insufficiently distinguishing when they should place more/less weight on the algorithm as a function of the impact of private features.

4.2.2. Prediction Error Results We next present results showing the impact of the naïve advice weighting bias on prediction error; Figure 3 summarizes results on participants' root median squared error (*RMedSE*). As one would expect, participants in the *Always High* $|v_i|$ condition have larger prediction error compared to participants in the *Always Low* $|v_i|$ condition, since the algorithm provides considerably less value when the impact of private feature is high ($t(238.00) = 5.743, p < 0.0001$). Similarly, participants in the *Mixed* $|v_i|$ condition have larger prediction error on products with high impact of private feature compared to their prediction error on products with low impact of private feature ($t(235.66) = 4.733, p < 0.0001$).

Our primary interest is studying the prediction error of participants in the *Mixed* $|v_i|$ condition to understand how their bias towards naïve advice weighting impacts their prediction error. As detailed in Hypothesis 2, we hypothesize that these participants will perform worse compared to sophisticated advice weighter benchmarks for both subsets of products with low and high impact of private feature. We perform two, one-sided t-tests comparing mean values of *RMedSE* across each row in Table 1.

First considering only products with a low impact of private feature, we test whether participants who experience a mixed exposure set have larger prediction error than participants who experience a single exposure set; specifically, we test whether

$$\frac{\sum_{j\in\mathcal{C}_M} RMedSE_j^L}{|\mathcal{C}_M|} \geq \frac{\sum_{j\in\mathcal{C}_L} RMedSE_j^L}{|\mathcal{C}_L|}. \tag{18}$$

As shown in the left two bars in Figure 3, participants in the *Mixed* $|v_i|$ condition had a significantly larger mean *RMedSE*^{L} vs. those in the *Always Low* $|v_i|$ condition ($t(229.08) = 1.940, p = 0.0268$).

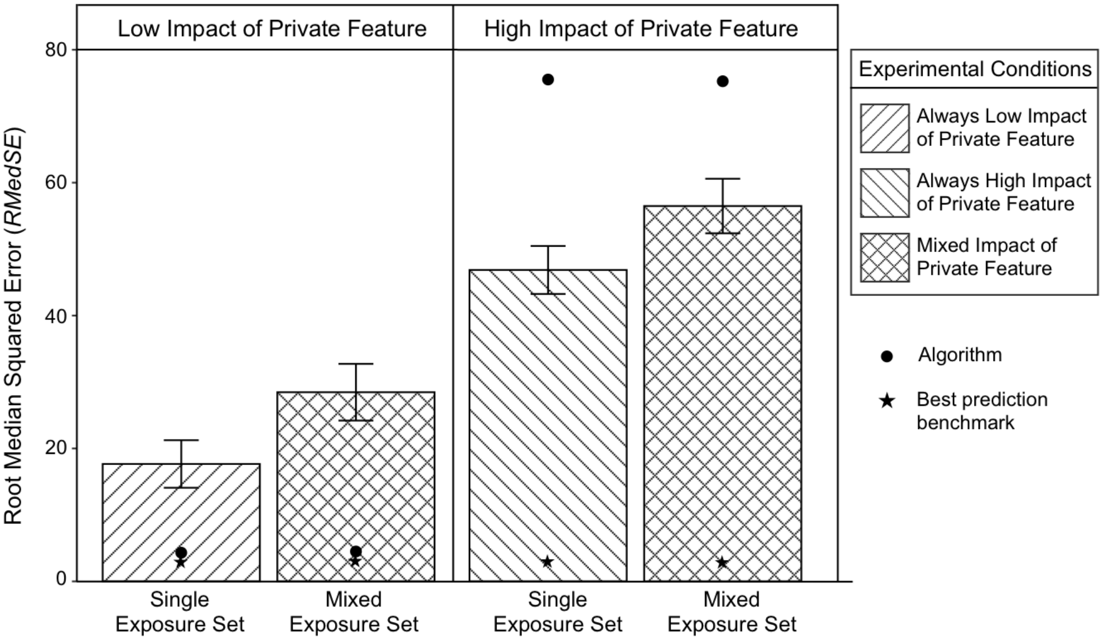


Figure 3 Root median squared error results are averaged (mean) by exposure set, separately for low and high impact of private feature; standard error bars are shown. Additionally, the mean *RMedSE* is reported for the algorithm and the best prediction benchmark.

Next considering only products with a high impact of private feature, we test whether participants who experience a mixed exposure set have larger prediction error than participants who experience a single exposure set; specifically, we test whether

$$\frac{\sum_{j\in\mathcal{C}_M} RMedSE_j^H}{|\mathcal{C}_M|} \geq \frac{\sum_{j\in\mathcal{C}_H} RMedSE_j^H}{|\mathcal{C}_H|}. \tag{19}$$

As shown in the right side of Figure 3, participants in the *Mixed* $|v_i|$ condition had a significantly larger *RMedSE*^{H} vs. those in the *Always High* $|v_i|$ condition ($t(233.67) = 1.761, p = 0.0398$).

Together, our results confirm Hypothesis 2 by showing that participants who experience a mixed exposure set perform worse for both subsets of products with low impact and high impact of private features, compared to sophisticated advice weighter benchmarks. This illustrates that a bias towards naïve advice weighting has a negative impact on prediction accuracy across the board.

4.3. Additional Analyses and Discussion

We report several supplementary analyses in Appendix B. These include (1) replicating results at the task-level, (2) observing minimal learning effects over time, (3) verifying robustness to unwindsorized *WOA*, (4) studying drivers of *WOA*, (5) conducting mediation analysis, (6) describing the performance of participants' demand predictions without the algorithm (in Step 3) to show how it compares to the algorithm, (7) examining participants' initial predictions that precede seeing the algorithm (in Step 5) to show that their accuracy is not significantly different from predictions without the algorithm (in Step 3), and (8) reporting task completion time statistics.

Study 1 provides evidence supporting a bias towards *naïve advice weighting* as opposed to sophisticated advice weighting behavior: people take an overly-constant weighted average of the algorithm's predictions with their initial predictions. As outlined in the theory presented in §3.3, our empirical results confirm that this bias leads to predictable patterns of over- and under-adherence to the algorithm and degrades performance when people face a mixed exposure set.

Although this study was designed to test the theory presented in §3.3, recall from §3.4 that the naïve advice weighting strategy is suboptimal not only because the weight on advice is overly constant, but also the best prediction benchmark (Y_i^*) may not even be in the advice-weighting region (see, e.g., Figure 1). Similar to what Proposition 3 predicted, we find that in Study 1, the best prediction benchmark falls within participants' advice-weighting regions in 55% of instances. In contrast, participants' actual final predictions $\hat{y}_i^{final}$ fall within their advice-weighting regions in 82% of instances across all treatment conditions; comparing these two numbers illustrates that people are advice weighting significantly more than they should be.

5. Study 2: The Impact of Transparency on Naïve Advice Weighting

What can system designers do to mitigate naïve advice weighting behavior? Study 1 demonstrates that extensive algorithm performance feedback is not enough for people to figure out when their private information warrants a large or small deviation from the algorithm. How can system designers help people with bounded cognitive ability with this issue? Of course, by its very nature, system designers often do not know peoples' private information. However, as detailed in Hypotheses 3 and 4, providing *feature transparency* (training humans about which features the algorithm *does* take into account) may help mitigate naïve advice weighting behavior by improving their ability to recognize when they have impactful private features, leading to improved performance accuracy for both products with low and high impact of private features. In practice, if system designers *were* aware of private features, communicating this information would likely be even more helpful.

We contrast this insight-inspired *feature transparency* intervention with another unrelated *training data transparency* intervention, in which we describe in more detail how much data the algorithm uses in its training process. Similar types of transparency have been shown to increase overall trust in algorithms (e.g., see Anik and Bunt 2021 and Balayn et al. 2022). However, we hypothesize that it will not be effective in mitigating naïve advice weighting because it is not designed to help participants effectively discriminate between situations where they should vs. should not adhere to the algorithm. We include this *training data transparency* manipulation as an additional control condition to disentangle the effect of providing people with knowledge of public information from the effect of providing people with more knowledge of the algorithm in general, which may also cause them to more carefully use the algorithm's forecasts.

5.1. Design

The participant experience, data generation, and dependent variables are identical to the *Mixed* $|v_i|$ condition in Study 1, except for the additions outlined in the three treatment conditions defined below.[3] Notably, these conditions only differ by information shared when introducing the algorithm; thus, both the algorithm and best prediction benchmark each have identical performance across all conditions.

1. *No Transparency.* This condition is identical to the *Mixed* $|v_i|$ condition in Study 1.
2. *Feature Transparency.* We add the following language when introducing the algorithm in Step 4: "The company has informed you that the algorithm uses only Feature A to make its demand predictions"[4].
3. *Training Data Transparency.* We add the following language when introducing the algorithm in Step 4: "The company has informed you that the algorithm was trained on a dataset of 9,834 products".

In both the *Feature Transparency* and *Training Data Transparency* conditions, we add a comprehension check question verifying that participants understood the transparency description. We also remind participants of the transparency description when predicting demand with the algorithm for each of the 20 products in Step 5; screenshots are included in Appendix F. For convenience, we define $\mathcal{C}_{NT}$, $\mathcal{C}_{FT}$, and $\mathcal{C}_{TDT}$ to be the set of participants assigned to the *No Transparency*, *Feature Transparency*, and *Training Data Transparency* conditions, respectively.

5.2. Results

Our analyses include data from 521 Prolific participants who passed the comprehension check criteria by answering at least three of five questions correctly on their first try[5]. By randomly

[3] We pre-registered our sample size, treatment conditions, data exclusion criteria, and planned analyses (see `https://aspredicted.org/6KL_L8F`). All statistical tests reported in the results are pre-registered unless otherwise indicated.

[4] Recall that Feature A corresponds to x_i^{pub} in our model.

[5] 525 workers were recruited to complete the study, each with a 99%+ approval rating, 25+ previous submissions, and English listed as a fluent language. Among the 521 participants, 229 were male, 307 had a Bachelor's or advanced degree, 412 were White, and 291 had a yearly household income of $50,000 or more.

assigning participants across conditions, we had 172 participants in the *No Transparency* condition, 171 in the *Feature Transparency* condition, and 178 in the *Training Data Transparency* condition. The mean study completion time was 30.80 minutes, and the mean bonus payment was $1.54.

5.2.1. Weight on Algorithm Results Figure 4 summarizes the results on participants' median weight on algorithm (*MedWOA*). We are most interested in how the difference in average $MedWOA$ between products with low and high impact of private features compares across participants who are provided feature transparency vs. no transparency. If feature transparency indeed mitigates the bias towards naïve advice weighting, we should see a larger difference in average $MedWOA$ for participants provided feature transparency, i.e.,

$$\frac{\sum_{j\in\mathcal{C}_{FT}} MedWOA_j^L}{|\mathcal{C}_{FT}|} - \frac{\sum_{j\in\mathcal{C}_{FT}} MedWOA_j^H}{|\mathcal{C}_{FT}|} \geq \frac{\sum_{j\in\mathcal{C}_{NT}} MedWOA_j^L}{|\mathcal{C}_{NT}|} - \frac{\sum_{j\in\mathcal{C}_{NT}} MedWOA_j^H}{|\mathcal{C}_{NT}|}. \tag{20}$$

When we regress the *MedWOA* for each participant on impact of private feature interacted with transparency type, clustering standard errors by participant, we indeed find a significant coefficient on the interaction term ($\beta = 0.173, p < 0.0001$), supporting Hypothesis 3. Similarly, we find that feature transparency mitigates naïve advice weighting behavior more than training data transparency; namely, we repeat the same analysis as above replacing no transparency with training data transparency ($\beta = 0.178, p < 0.0001$). Finally (as an ex post test), we find that training data transparency does not significantly mitigate naïve advice weighting behavior ($\beta = -0.005, p = 0.871$). Results are detailed in Table 2.

Our results confirm Hypothesis 3 by showing that feature transparency mitigates naïve advice weighting behavior by helping humans recognize when they have impactful private features that warrant a substantial deviation from the algorithm. Further, we show that a different kind of transparency – training data transparency – is not effective in mitigating naïve advice weighting behavior because it is not designed to help participants effectively discriminate between situations where they should vs. should not adhere to the algorithm.

5.2.2. Prediction Error Results We next present results showing the impact of transparency type on prediction error; Figure 5 summarizes results on participants' root median squared error. As expected, within each condition, participants have larger prediction error on products with high impact of private feature vs. low impact, since the algorithm provides considerably less value when the impact of private feature is high.

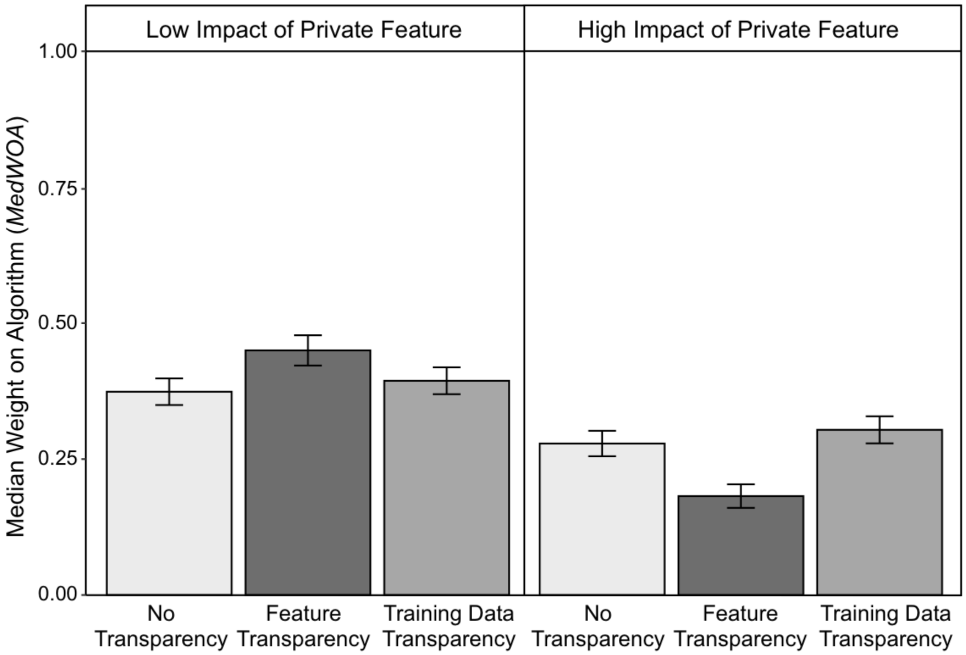


Figure 4 Median weight on algorithm results are averaged (mean) by transparency type, separately for low and high impact of private feature; standard error bars are shown.

Table 2 Regression Analyses of *MedWOA* by Impact of Private Feature and Transparency Type

Dependent Variable:	$MedWOA$		
Model:	Feature vs. No Transp	Feature vs. Training Data	Training Data vs. No Transp
Variables			
(Intercept)	0.2786***	0.3038***	0.2786***
	(0.0234)	(0.0249)	(0.0234)
Low $\|v_i\|$	0.0953***	0.0902***	0.0953***
	(0.0205)	(0.0231)	(0.0205)
Feature Transparency	-0.0967***	-0.1219***	
	(0.0319)	(0.0330)	
Low $\|v_i\|$ × Feature Transparency	0.1727***	0.1778***	
	(0.0352)	(0.0368)	
Training Data Transparency			0.0252
			(0.0342)
Low $\|v_i\|$ × Training Data Transparency			-0.0050
			(0.0309)
Fit statistics			
Observations	686	698	700
R^2	0.09028	0.08604	0.02153
Adjusted R^2	0.08627	0.08209	0.01732

Clustered (Participant) standard-errors in parentheses
*Signif. Codes: ***: 0.01, **: 0.05, *: 0.1*

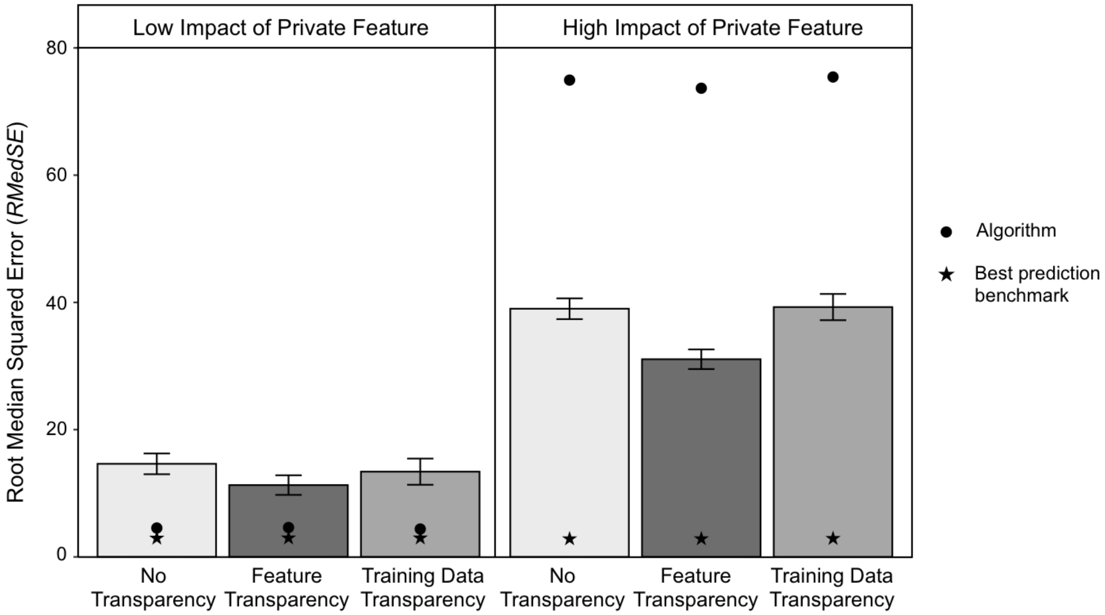


Figure 5 Root median squared error results are averaged (mean) by transparency type, separately for low and high impact of private feature; standard error bars are shown. Additionally, the mean *RMedSE* is reported for the algorithm and the best prediction benchmark.

Our primary interest is studying how the prediction error of participants who are provided feature transparency compares to participants given no transparency. As detailed in Hypothesis 4, we hypothesize that feature transparency will lead to smaller prediction error for both subsets of products with low and high impact of private feature. We first consider only products with a low impact of private feature, and we use a one-sided t-test to test whether participants provided with feature transparency have smaller prediction error than participants provided with no transparency:

$$\frac{\sum_{j\in\mathcal{C}_{FT}} RMedSE_j^L}{|\mathcal{C}_{FT}|} \leq \frac{\sum_{j\in\mathcal{C}_{NT}} RMedSE_j^L}{|\mathcal{C}_{NT}|}. \tag{21}$$

As shown in Figure 5, participants in the *Feature Transparency* condition had a significantly smaller mean *RMedSE* compared to participants in the *No Transparency* condition for products with low impact of private feature ($t(340.99) = 2.718, p = 0.0035$).

We next repeat this analysis on products with a high impact of private feature; we test whether

$$\frac{\sum_{j\in\mathcal{C}_{FT}} RMedSE_j^H}{|\mathcal{C}_{FT}|} \leq \frac{\sum_{j\in\mathcal{C}_{NT}} RMedSE_j^H}{|\mathcal{C}_{NT}|}. \tag{22}$$

As shown in Figure 5, participants in the *Feature Transparency* condition had a significantly smaller *RMedSE* compared to participants in the *No Transparency* condition for products with high impact of private feature ($t(339.98) = 3.536, p = 0.0002$). When considering overall *RMedSE* across products with both low and high impact of private feature, we also find that participants in the *Feature Transparency* condition had a significantly smaller *RMedSE* compared to participants in the *No Transparency* condition ($t(334.38) = 4.112, p < 0.0001$).

Similarly, we can compare prediction errors across participants who are provided feature transparency vs. training data transparency. We find that participants in the *Feature Transparency* condition had a smaller mean *RMedSE* compared to participants in the *Training Data Transparency* condition for products with low impact of private feature ($t(303.85) = 1.331, p = 0.0921$), as well as for products with high impact of private feature ($t(342.58) = 3.186, p = 0.0008$); considering overall *RMedSE*, we find that feature transparency leads to significant improvements ($t(283.66) = 2.550, p = 0.0057$). As expected, training data transparency is less effective in mitigating naïve advice weighting behavior because it is not designed to help participants effectively discriminate between situations where they should vs. should not adhere to the algorithm.

Together, our results confirm Hypothesis 4 by showing that participants who are provided feature transparency perform better for both subsets of products with low impact and high impact of private features, compared to participants given no transparency or training data transparency.

5.3. Additional Analyses and Discussion

We report several supplementary analyses in Appendix C.

Text Analysis of Free-Response Question We included an open-ended question at the end of the study asking participants to explain their decision-making process. Ex post text analysis indicates that *Feature Transparency* causes people to be 32% less likely to mention "averaging" but 31% more likely to mention "adjusting." Moreover, regression analysis reveals that people who mention "adjusting" had an 18% lower prediction error and suffered significantly less from naïve advice weighting than participants who mention "averaging".

Supplementary DV We can directly examine participants' adjustment errors by defining a new *absolute percent adjustment error (APAE)* dependent variable (i.e., how far away a participant's adjustment from the algorithm's recommendation is from the optimal adjustment). It is zero when the adjustment is optimal and becomes more positive as the adjustment is further from optimal. Consistent with the patterns with *RMedSE*, we find that participants' median *APAE* are significantly lower in *Feature Transparency* than in *No Transparency* or *Training Data Transparency*.

Time There are no significant differences between treatment conditions in the average time for initial predictions nor final predictions.

In summary, Study 2 shows that providing feature transparency – training humans on what information the algorithm does use – helps to mitigate *naïve advice weighting* behavior more effectively than other types of algorithm transparency (e.g., training data transparency) that uniformly increase humans' adherence to the algorithm both when they should and should not do so. That said, similar to Study 1, the best prediction benchmark fell outside the advice-weighting region 52% of the time, yet participants' final predictions fell within their advice-weighting regions in 92% of instances under *No Transparency*, 93% of instances under *Training Data Transparency*, and 86% of instances under *Feature Transparency*. This suggests that our *Feature Transparency* intervention only modestly helps humans move away from advice weighting strategies; rather, the benefit of the intervention seems to primarily be driven by helping humans move away from applying overly constant weights.

6. Study 3: Improving Feature Transparency

Due to our findings in Study 2, we are motivated to improve upon feature transparency by helping humans move away from advice weighting strategies. We hypothesize that – in addition to feature transparency – nudging humans to follow a strategy in which they anchor on the algorithm and adjust based only on their private features may further improve their use of algorithmic predictions. Study 3 tests this improved feature transparency intervention in an online experiment.

We additionally use Study 3 to generalize some of our prior results to an experimental setup with several key differences from Studies 1 and 2. In Study 3, we situate participants in a naturalistic setting where they make demand forecasts for clothing items using information they have intuition around (e.g., price and advertising spend). We use a more realistic demand model that cannot easily be intuited, and we increase the complexity of the structure of the private feature (ad spend), generating it according to a non-negative random variable with a non-zero mean.

6.1. Design

All participants encounter data generated according to a mixed impact of private feature, with three treatment conditions that differ by information shared when introducing the algorithm.[6]

6.1.1. Conditions

1. *No Transparency.* This is similar to the *No Transparency* condition in Study 2.

[6] We pre-registered our sample size, treatment conditions, data exclusion criteria, and planned analyses (see `https://aspredicted.org/P7T_1WC`). All statistical tests reported in the results are pre-registered unless otherwise indicated.

2. *Feature Transparency.* This is similar to the *Feature Transparency* condition in Study 2. Participants are informed the algorithm uses only price to make its demand predictions.
3. *Adjusting Nudge.* Participants are provided with all the information as in the *Feature Transparency* condition and are additionally nudged towards using a strategy of anchoring on the algorithm's predictions and adjusting them using only private features.

6.1.2. Behind the Scenes: Data Generation For clothing item i, actual demand is generated as

$$Y_i = \begin{cases} 1250(X_i^{pub} - 25)^{-0.3} + 70 + 1.2X_i^{priv} + \epsilon_i & X_i^{pub} \leq 99 \\ 1250(X_i^{pub} - 70)^{-0.3} - 45 + 1.2X_i^{priv} + \epsilon_i & 99 < X_i^{pub} \leq 149 \\ 1250(X_i^{pub} - 120)^{-0.3} - 160 + 1.2X_i^{priv} + \epsilon_i & 149 < X_i^{pub} \end{cases}$$

where ϵ_i is drawn from a normal distribution with mean 0 and standard deviation 5, X_i^{pub} represents price and is randomly sampled from $\{54, 59, 64, 69, ..., 194, 199\}$ reflecting common prices for clothing items, and X_i^{priv} represents advertising spend and is drawn from a discrete uniform distribution with support $\{90, 110\}$ with probability 0.5 and $\{0, 50\} \bigcup \{150, 200\}$ with probability 0.5. This demand model mimics a constant elasticity demand function where demand exponentially decreases as price (X_i^{pub}) increases, with discontinuities leading to larger drops in demand as price increases from \$99 to \$100 and \$149 to \$150.

The algorithm's demand prediction is generated by the equation:

$$Y_i = \begin{cases} 1250(X_i^{pub} - 25)^{-0.3} + 70 + 1.2 * 100 & X_i^{pub} \leq 99 \\ 1250(X_i^{pub} - 70)^{-0.3} - 45 + 1.2 * 100 & 99 < X_i^{pub} \leq 149 \\ 1250(X_i^{pub} - 120)^{-0.3} - 160 + 1.2 * 100 & 149 < X_i^{pub} \end{cases}$$

Note that in the last term, the 100 comes from the fact that $\mathbb{E}[X_i^{priv}] = 100$. Thus, the impact of private feature is $V_i = 1.2(X_i^{priv} - 100)$.

6.1.3. Participant Experience Study 3 follows a similar participant experience as in Study 2 but with the following key changes in addition to the contextual setting and data generation process. Select screenshots are included in Appendix G.

In *Step 4: Algorithm Introduction*, participants in this study are given additional information about the algorithm and strategies for using it based on their assigned treatment condition. Namely, all participants are shown a row below the summary table with the average value for each displayed column (e.g., average price, average ad spend, etc.). Participants in the *Feature Transparency* and *Adjusting Nudge* conditions are told, “The company has informed you that the algorithm uses only Price to make its demand forecasts. It doesn’t have access to any other information.” Participants in the *Adjusting Nudge* condition are also told, “The algorithm optimally uses the price information, but you may have extra information beyond price that the algorithm doesn’t have access to. The algorithm assumes an average value of any extra information when making a forecast; therefore the algorithm makes great forecasts when your extra information is close to its average value, but not when your extra information is far from its average value. Therefore, we recommend you collaborate with the algorithm using the following strategy: Follow the algorithm when your extra information is close to its average value. Only override the algorithm if your extra information is far from its average value. If you override, focus on using your extra information to adjust the algorithm up or down.”

In *Step 5: Demand Predictions with Algorithm*, unlike in Studies 1 and 2, in Study 3, participants are not asked for an initial demand prediction $\hat{y}_i^{init}$. We made this choice in order to try to prevent people in the *Adjusting Nudge* condition from anchoring on their initial prediction, and we chose to make this consistent across all three conditions to ensure that any differences we see across conditions are not due to the absence of eliciting $\hat{y}_i^{init}$. For participants in the *Adjusting Nudge* condition, before asking for $\hat{y}_i^{final}$, we ask “Do you have any extra information that is far from its average value?” If they select “No, I’ll use the algorithm,” then the algorithm’s demand prediction, $\hat{y}_i^{alg}$, is recorded as their final demand prediction, $\hat{y}_i^{final}$. If they select “Yes, my extra information is far from its average value and I’d like to override the algorithm,” then they are asked for their final demand prediction, $\hat{y}_i^{final}$. Here, participants are prompted with the strategy to, “Use only your extra information to adjust the algorithm’s forecast up or down.”

6.2. Results

Our analyses include data from 549 Prolific participants who successfully passed the comprehension check criteria by answering at least three of four questions correctly on their first try[7]. By randomly assigning participants across conditions, we had 183 participants in each condition. The mean study completion time was 29.39 minutes, and the mean bonus payment was \$1.37[8].

We first present results showing the impact of transparency type on participants’ root median squared error (Figure 6). Our primary interest is studying how the prediction error of participants

[7] 600 workers were recruited to complete the study who had not previously completed Study 2, each with a 99%+ approval rating, 25+ previous submissions, and English listed as a fluent language. Among the 549 participants, 275 were male, 321 had a Bachelor’s or advanced degree, 382 were White, and 342 had a yearly household income of \$50,000 or more.

[8] Participants received a bonus of \$7 – \$0.08 × (Root Mean Squared Error) based on their demand predictions in Steps 3 and 5.

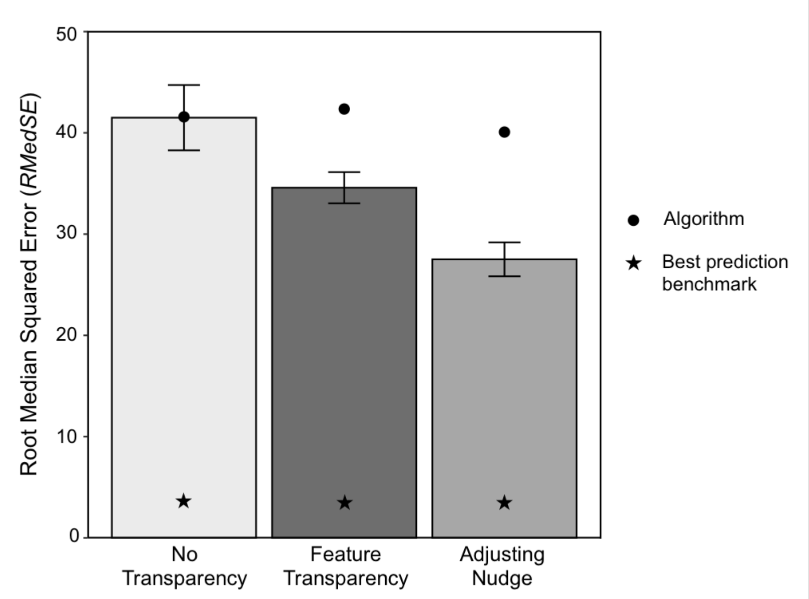


Figure 6 Root median squared error results are averaged (mean) by transparency type; standard error bars are shown. Additionally, the mean *RMedSE* is reported for the algorithm and the best prediction benchmark.

who are provided with feature transparency paired with an adjusting nudge compares to participants given either no transparency or feature transparency alone. Using a one-sided t-test, we find that participants in the *Adjusting Nudge* condition had a significantly smaller *RMedSE* compared to participants in the *No Transparency* condition ($t(273.60) = 3.850, p < 0.0001$). Additionally, using a one-sided t-test, we find that participants in the *Adjusting Nudge* condition had a significantly smaller *RMedSE* compared to participants in the *Feature Transparency* condition ($t(361.44) = 3.109, p = 0.0010$). Together, these results confirm Hypothesis 5, showing that participants who are provided with the adjusting nudge have lower prediction error compared to participants given no transparency or feature transparency alone. Furthermore, we replicate our results in Section 5.2.2 across a contextual setting and different data generation process: we find that participants who are provided with feature transparency have a lower prediction error compared to participants given no transparency ($t(260.90) = 1.936, p = 0.0270$). Note that we cannot calculate *WOA* to replicate our results in Section 5.2.1 since we do not elicit $\hat{y}_i^{init}$ in Study 3.

To understand how the adjusting nudge drives further improvement in decreasing participants' prediction error relative to feature transparency alone, we first examine whether participants provided with the adjusting nudge have a lower prediction error for clothing items with low impact of private feature.[9] Since our nudge explains that adjustments should only be made when the private feature is far from its average value, we would expect smaller adjustments – and therefore

[9] We define low impact of private feature clothing items as those where x_i^{priv} (ad spend) is between \$90 to \$110 (inclusive) while high impact of private feature clothing items are those where x_i^{priv} is between \$0 to \$50 or \$150 to \$200 (inclusive).

better predictions – when the impact of private features is low. Using a one-sided t-test, we indeed find that participants in the *Adjusting Nudge* condition had a significantly smaller *RMedSE* on low impact of private feature clothing items (*RMedSE* = 9.4) compared to participants in the *Feature Transparency* condition (*RMedSE* = 19.3) ($t(293.25) = 6.776, p < 0.0001$). The graph of results split by low and high impact of private features is included in Appendix D.1.

We next examine whether under the adjusting nudge, participants more frequently adjust the algorithm's predictions in the correct direction, indicating they are no longer advice weighting and instead using only private features to adjust the algorithm. We define a participant's adjustment of the algorithm's prediction as being in the *correct direction* iff the following holds:

$$(\hat{y}_i^{alg} \geq \hat{y}_i^{final} \wedge \hat{y}_i^{alg} \geq y_i^*) \vee (\hat{y}_i^{alg} \leq \hat{y}_i^{final} \wedge \hat{y}_i^{alg} \leq y_i^*).$$

Participants provided with *Feature Transparency* have an average rate of making adjustments to the algorithm in the correct direction of 74.4% ($SD = 15.386$) while participants provided with the *Adjusting Nudge* have an average rate of adjustments in the correct direction of 93.5% ($SD = 9.467$). A one-sided t-test reveals this difference is significant with participants in the *Adjusting Nudge* condition making significantly more frequent adjustments to the algorithm in the correct direction ($t(302.53) = -14.301, p < 0.0001$). See Appendix D.2 for more details.

6.3. Discussion

Study 3 confirms our hypothesis that feature transparency can be further improved by educating humans to use a strategy of anchoring on algorithmic predictions and adjusting based only on their private features. Our results suggest that our improved intervention – pairing feature transparency with this adjusting nudge – helps humans overcome *both* reasons why naïve advice weighting is suboptimal. Namely, feature transparency primarily helps humans move away from overly-constant weights, and the adjusting nudge further helps humans move away from advice weighting heuristics in general.

7. Conclusion

This paper proposes and provides experimental evidence that people's algorithm overrides are biased towards naïve advice weighting, taking a constant weighted average of the algorithm's prediction and their own prediction without the algorithm. This causes people to over-adhere to the algorithm when they have highly valuable private information and under-adhere to the algorithm when they do not, as well as frequently adjust the algorithm in the incorrect direction. However, providing people with feature transparency and pairing it with education on adjusting algorithmic predictions based only on private information can help mitigate their bias towards NAW and improve predictive performance.

Our results generate insights for managers seeking to design algorithms and how they interface with humans. First, we help identify *when* human-algorithm collaborative performance is most hurt by NAW: when humans *sometimes* have valuable private information. In these settings, interventions that uniformly increase people's trust in the algorithm (as many types of algorithm transparency are designed to do) do not help address the underlying issue. Instead, interventions such as feature transparency – which are designed to help people discern when they have more or less valuable private information – are more appropriate. Because there are a plethora of types of algorithm transparency (see §2.3) each with their own goals in changing how people interact with the algorithm, it is important for system designers to understand when they are in a situation which warrants addressing NAW rather than a different fundamental issue (e.g., incentive or trust issues).

Second, by illuminating *why* feature transparency and education on adjusting strategies helps, our results provide insights for algorithm developers. We recommend that algorithms are designed so that (i) features used in the algorithm can be communicated and explained to non-experts, and (ii) features are chosen in such a way that people correctly recognize when they have private information that warrants deviation. Such guidelines can help algorithm designers with feature engineering as well as choosing amongst algorithms of different levels of complexity and amongst different sets of features that have similar predictive performance (e.g., Xin et al. 2022).

Finally, our results shed light on when system designers should let humans override algorithms in general. Letting humans perform the final aggregation task subjects the system to human cognitive limitations and noise, which leads some experts to suggest avoiding this setup when possible (e.g., Kahneman et al. 2022). However, even when the organizational setting does not require a human to have final decision authority, our results suggest that we should still let humans have override authority when they have access to information that is unknown to the algorithm but predictive of the outcome. In these types of settings, letting humans override the algorithm can potentially add more value through incorporating their private information than harm by exposing the system to their noise and other biases. Of course, we also recommend that system designers work to identify, collect, and codify private information if possible. In general, we see research opportunity in improving our understanding of how to develop systems in which humans and algorithms focus and hone their relative strengths to enhance their long-run collaborative performance.

References

Anik AI, Bunt A (2021) Data-centric explanations: Explaining training data of machine learning systems to promote transparency. *CHI Conference on Human Factors in Computing Systems*.

Arvan M, Fahimnia B, Reisi M, Siemsen E (2019) Integrating human judgement into quantitative forecasting methods: A review. *Omega*, 86:237–252.

Balayn A, Rikalo N, Lofi C, Yang J, Bozzon A (2022) How can explainability methods be used to support bug identification in computer vision models? *CHI Conference on Human Factors in Computing Systems*.

Ball RT, Ghysels E (2018) Automated earnings forecasts: Beat analysts or combine and conquer? *Management Science*, 64(10):4936–4952.

Bastani H, Bastani O, Sinchaisri WP (2022) Improving human decision-making with machine learning, working paper.

Beer R, Qi A, Rios I (2022) Behavioral externalities of process automation, working paper.

Blattberg RC, Hoch SJ (1990) Database models and managerial intuition: 50% model + 50% manager. *Management Science*, 36(8):887–899.

Bolton GE, Katok E, Stangl T (2022) Failures in the communication of risk: Decisions and numeracy. *Production and Operations Management*.

Bonaccio S, Dalal RS (2006) Advice taking and decision-making: An integrative literature review, and implications for the organizational sciences. *Organizational Behavior and Human Decision Processes*, 101(2):127–151.

Brau R, Aloysius J, Siemsen E (2023) Demand planning for the digital supply chain: How to integrate human judgment and predictive analytics. *Journal of Operations Management*, 69(6):965–982.

Cadario R, Longoni C, Morewedge CK (2021) Understanding, explaining, and utilizing medical artificial intelligence. *Nature Human Behaviour*.

Caro F, Saez de Tejada Cuenca A (2022) Believing in analytics: Managers' adherence to price recommendations from a DSS. *Manufacturing & Service Operations Management*.

Castelo N, Bos MW, Lehmann DR (2019) Task-dependent algorithm aversion. *Journal of Marketing Research*, 56(5):809–825.

Clemen RT (1989) Combining forecasts: A review and annotated bibliography. *International journal of forecasting*, 5(4):559–583.

Cui R, Gallino S, Moreno A, Zhang DJ (2018) The operational value of social media information. *Production and Operations Management*, 27(10):1749–1769.

Dietvorst BJ, Simmons JP, Massey C (2015) Algorithm aversion: People erroneously avoid algorithms after seeing them err. *Journal of Experimental Psychology: General*, 144(1):114–126.

Fildes R, Goodwin P, Lawrence M, Nikolopoulos K (2009) Effective forecasting and judgmental adjustments: An empirical evaluation and strategies for improvement in supply-chain planning. *International Journal of Forecasting*, 25(1):3–23.

Flicker BA (2018) Managerial insight and ”optimal” algorithms. *Working Paper*.

Fügener A, Grahl J, Gupta A, Ketter W (2022) Cognitive challenges in human–artificial intelligence collaboration: Investigating the path toward productive delegation. *Information Systems Research*, 33(2):678–696.

Gebru T, Morgenstern J, Vecchione B, Vaughan JW, Wallach H, III HD, Crawford K (2021) Datasheets for datasets. *Communications of the ACM*, 64(12):86–92.

Gino F, Moore DA (2007) Effects of task difficulty on use of advice. *Journal of Behavioral Decision Making*, 20(1):21–35.

Green B (2022) The flaws of policies requiring human oversight of government algorithms. *Computer Law & Security Review*, 45:105681.

Harvey N, Fischer I (1997) Taking advice: Accepting help, improving judgment, and sharing responsibility. *Organizational Behavior and Human Decision Processes*, 70(2):117–133.

Hind M, Houde S, Martino J, Mojsilovic A, Piorkowski D, Richards J, Varshney KR (2020) Experiences with improving the transparency of AI models and services. *CHI Conference on Human Factors in Computing Systems*.

Hoffman M, Kahn LB, Li D (2018) Discretion in hiring. *The Quarterly Journal of Economics*, 133(2):765–800.

Ibanez MR, Clark JR, Huckman RS, Staats BR (2018) Discretionary task ordering: Queue management in radiological services. *Management Science*, 64(9):4389–4407.

Ibrahim R, Kim SH (2019) Is expert input valuable? The case of predicting surgery duration. *Seoul Journal of Business*, 25(2):1–34.

Ibrahim R, Kim SH, Tong J (2021) Eliciting human judgment for prediction algorithms. *Management Science*, 67(4):2314–2325.

Kahneman D, Sibony O, Sunstein C (2022) *Noise* (HarperCollins UK).

Kesavan S, Kushwaha T (2020) Field experiment on the profit implications of merchants’ discretionary power to override data-driven decision-making tools. *Management Science*, 66(11):5182–5190.

Khosrowabadi N, Hoberg K, Imdahl C (2022) Evaluating human behaviour in response to AI recommendations for judgemental forecasting. *European Journal of Operational Research*, 303(3):1151–1167.

Kim SH, Song H (2022) How digital transformation can improve hospitals’ operational decisions. *Harvard Business Review*.

Lage I, Chen E, He J, Narayanan M, Kim B, Gershman SJ, Doshi-Velez F (2019) Human evaluation of models built for interpretability. *AAAI Conference on Human Computation and Crowdsourcing*, 7(1):59–67.

Lakkaraju H, Bastani O (2020) ”How do I fool you?”: Manipulating user trust via misleading black box explanations. *AAAI/ACM Conference on AI, Ethics, and Society*, 79–85.

Lehmann CA, Haubitz CB, Fügener A, Thonemann UW (2022) The risk of algorithm transparency: How algorithm complexity drives the effects on the use of advice. *Production and Operations Management*.

Lipton ZC (2017) The mythos of model interpretability, working paper.

Logg JM, Minson JA, Moore DA (2019) Algorithm appreciation: People prefer algorithmic to human judgment. *Organizational Behavior and Human Decision Processes*, 151:90–103.

Luong A, Kumar N, Lang KR (2020) Algorithmic decision-making: Examining the interplay of people, technology, and organizational practices through an economic experiment. *Working Paper*.

Mitchell M, Wu S, Zaldivar A, Barnes P, Vasserman L, Hutchinson B, Spitzer E, Raji ID, Gebru T (2019) Model cards for model reporting. *Proceedings of the Conference on Fairness, Accountability, and Transparency*, 220–229.

Palley AB, Soll JB (2019) Extracting the wisdom of crowds when information is shared. *Management Science*, 65(5):2291–2309.

Petropoulos F, Kourentzes N, Nikolopoulos K, Siemsen E (2018) Judgmental selection of forecasting models. *Journal of Operations Management*, 60(1):34–46.

Poursabzi-Sangdeh F, Goldstein DG, Hofman JM, Wortman Vaughan JW, Wallach H (2021) Manipulating and measuring model interpretability. *CHI Conference on Human Factors in Computing Systems*.

PricewaterhouseCoopers (2022) PwC 2022 AI business survey. Technical report, PricewaterhouseCoopers.

Ransbotham S, Khodabandeh S, Kiron D, Candelon F, Chu M, LaFountain B (2020) Expanding AI's impact with organizational learning. Technical report, MIT Sloan Management Review and Boston Consulting Group.

Rios I, Saban D, Zheng F (2022) Improving match rates in dating markets through assortment optimization. *Manufacturing & Service Operations Management*.

Sah S, Moore DA, MacCoun RJ (2013) Cheap talk and credibility: The consequences of confidence and accuracy on advisor credibility and persuasiveness. *Organizational Behavior and Human Decision Processes*, 121(2):246–255.

Sniezek JA, Van Swol LM (2001) Trust, confidence, and expertise in a judge-advisor system. *Organizational Behavior and Human Decision Processes*, 84(2):288–307.

Snyder C, Keppler S, Leider S (2022) Algorithm reliance under pressure: The effect of customer load on service workers, working paper.

Soll JB, Larrick RP (2009) Strategies for revising judgment: How (and how well) people use others' opinions. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 35(3):780–805.

Soll JB, Palley AB, Rader CA (2021) The bad thing about good advice: Understanding when and how advice exacerbates overconfidence. *Management Science*, 68(4):2377–3174.

Soule D, Grushka-Cockayne Y, Merrick J (2023) A heuristic for combining correlated experts when there are few data. *Management Science*.

Su X (2008) Bounded rationality in newsvendor models. *Manufacturing & Service Operations Management*, 10(4):566–589.

Surowiecki J (2005) *The wisdom of crowds* (Anchor).

Xin R, Zhong C, Chen Z, Takagi T, Seltzer M, Rudin C (2022) Exploring the whole rashomon set of sparse decision trees, working paper.

Yaniv I, Kleinberger E (2000) Advice taking in decision making: Egocentric discounting and reputation formation. *Organizational Behavior and Human Decision Processes*, 83(2):260–281.

Yeomans M, Shah A, Mullainathan S, Kleinberg J (2019) Making sense of recommendations. *Journal of Behavioral Decision Making*, 32(4):403–414.

Yin M, Wortman Vaughan J, Wallach H (2019) Understanding the effect of accuracy on trust in machine learning models. *CHI Conference on Human Factors in Computing Systems*, 1–12.

Appendix Contents

Appendix A: Proofs

Proof of Proposition 1: We can rewrite the objective function as follows:

$$\begin{aligned}
&\mathbb{E}\big[\big(Y_i-(\lambda\hat{Y}_i^{alg}+(1-\lambda)\hat{Y}_i^{init})\big)^2\big]\\
&=\mathbb{E}\big[\big(\lambda(Y_i-\hat{Y}_i^{alg})+(1-\lambda)(Y_i-\hat{Y}_i^{init})\big)^2\big]\\
&=Var\big[\lambda(Y_i-\hat{Y}_i^{alg})+(1-\lambda)(Y_i-\hat{Y}_i^{init})\big]\\
&=\lambda^2 Var\big[(Y_i-\hat{Y}_i^{alg})\big]+(1-\lambda)^2 Var\big[(Y_i-\hat{Y}_i^{init})\big]\\
&=\lambda^2\mathbb{E}\big[(Y_i-\hat{Y}_i^{alg})^2\big]+(1-\lambda)^2\mathbb{E}\big[(Y_i-\hat{Y}_i^{init})^2\big].
\end{aligned}$$

The first equality simply distributes Y_i into λY_i and $(1-\lambda)Y_i$. The second equality follows from $Var[X]=\mathbb{E}[X^2]-\mathbb{E}[X]^2$ and because $\mathbb{E}[Y_i-\hat{Y}_i^{init}]=\mathbb{E}[Y_i-\hat{Y}_i^{alg}]=0$ from Assumption 1. The third equality follows from the independence in Assumption 2. The final equality again leverages $Var[X]=\mathbb{E}[X^2]-\mathbb{E}[X]^2$ and Assumption 1.

The first order conditions are

$$0=2\lambda\mathbb{E}\big[(Y_i-\hat{Y}_i^{alg})^2\big]+(2\lambda-2)\mathbb{E}\big[(Y_i-\hat{Y}_i^{init})^2\big]$$

and solving for λ gives λ^{NAW}, which is between 0 and 1 because $\mathbb{E}\big[(Y_i-\hat{Y}_i^{alg})^2\big]$ and $\mathbb{E}\big[(Y_i-\hat{Y}_i^{init})^2\big]$ are both nonnegative. The second order conditions confirm convexity.

Proof of Proposition 2: The SAW problem is separable into problems SAW_L and SAW_H, where

$$SAW_L:\ \min_{\lambda_L\in[0,1]}\mathbb{E}\big[\big(Y_i-(\lambda_L\hat{Y}_i^{alg}+(1-\lambda_L)\hat{Y}_i^{init})\big)^2|(\boldsymbol{X}_i^{pub},\boldsymbol{X}_i^{priv})\in\mathcal{S}_L\big]\mathbb{P}((\boldsymbol{X}_i^{pub},\boldsymbol{X}_i^{priv})\in\mathcal{S}_L) \tag{23}$$

$$SAW_H:\ \min_{\lambda_H\in[0,1]}\mathbb{E}\big[\big(Y_i-(\lambda_H\hat{Y}_i^{alg}+(1-\lambda_H)\hat{Y}_i^{init})\big)^2|(\boldsymbol{X}_i^{pub},\boldsymbol{X}_i^{priv})\in\mathcal{S}_H\big]\mathbb{P}((\boldsymbol{X}_i^{pub},\boldsymbol{X}_i^{priv})\in\mathcal{S}_H). \tag{24}$$

Now, $\mathbb{P}((\boldsymbol{X}_i^{pub},\boldsymbol{X}_i^{priv})\in\mathcal{S}_L)$ and $\mathbb{P}((\boldsymbol{X}_i^{pub},\boldsymbol{X}_i^{priv})\in\mathcal{S}_H)$ do not depend on λ_L or λ_H. Thus, the process for solving for the optimal weight in each of these problems is the same as that in NAW. Doing so yields

$$\begin{aligned}
\lambda_L^{SAW}=&\frac{\mathbb{E}[(Y_i-\hat{Y}_i^{init})^2|(\boldsymbol{X}_i^{pub},\boldsymbol{X}_i^{priv})\in\mathcal{S}_L]}{\mathbb{E}[(Y_i-\hat{Y}_i^{init})^2|(\boldsymbol{X}_i^{pub},\boldsymbol{X}_i^{priv})\in\mathcal{S}_L]+\mathbb{E}[(Y_i-\hat{Y}_i^{alg})^2|(\boldsymbol{X}_i^{pub},\boldsymbol{X}_i^{priv})\in\mathcal{S}_L]}\\
=&\frac{\mathbb{E}[(Y_i-\hat{Y}_i^{init})^2]}{\mathbb{E}[(Y_i-\hat{Y}_i^{init})^2]+\mathbb{E}[(Y_i-\hat{Y}_i^{alg})^2|(\boldsymbol{X}_i^{pub},\boldsymbol{X}_i^{priv})\in\mathcal{S}_L]}.
\end{aligned}$$

where the second equality follows from the independence Assumption 2. Similarly,

$$\lambda_H^{SAW}=\frac{\mathbb{E}[(Y_i-\hat{Y}_i^{init})^2]}{\mathbb{E}[(Y_i-\hat{Y}_i^{init})^2]+\mathbb{E}[(Y_i-\hat{Y}_i^{alg})^2|(\boldsymbol{X}_i^{pub},\boldsymbol{X}_i^{priv})\in\mathcal{S}_H]}.$$

Now, the definition of the partitions

$$\mathbb{E}[(V_i)^2|(\boldsymbol{X}_i^{pub},\boldsymbol{X}_i^{priv})\in\mathcal{S}_L]<\mathbb{E}[(V_i)^2|(\boldsymbol{X}_i^{pub},\boldsymbol{X}_i^{priv})\in\mathcal{S}_H]$$

implies

$$\mathbb{E}[(Y_i-\hat{Y}_i^{alg})^2|(\boldsymbol{X}_i^{pub},\boldsymbol{X}_i^{priv})\in\mathcal{S}_L]<\mathbb{E}[(Y_i-\hat{Y}_i^{alg})^2|(\boldsymbol{X}_i^{pub},\boldsymbol{X}_i^{priv})\in\mathcal{S}_H].$$

And, by the law of total expectation, we have

$$\mathbb{E}[(Y_i-\hat{Y}_i^{alg})^2|(\boldsymbol{X}_i^{pub},\boldsymbol{X}_i^{priv})\in\mathcal{S}_L]<\mathbb{E}[(Y_i-\hat{Y}_i^{alg})^2]<\mathbb{E}[(Y_i-\hat{Y}_i^{alg})^2|(\boldsymbol{X}_i^{pub},\boldsymbol{X}_i^{priv})\in\mathcal{S}_H].$$

Combining this inequality with the expressions for $\lambda_L^{SAW},\lambda_H^{SAW},\lambda^{NAW}$ gives $\lambda_H^{SAW}<\lambda^{NAW}<\lambda_L^{SAW}$. The second part of the proposition follows from the fact that the NAW solution is feasible to the SAW problem.

Proof of Proposition 3: We can write the probability as follows:

$$\begin{aligned}&\mathbb{P}\big(\min\{\hat{Y}_i^{alg},\hat{Y}_i^{init}\}\le Y_i^*\le\max\{\hat{Y}_i^{alg},\hat{Y}_i^{init}\}\big)\\&=\mathbb{P}\big(\hat{Y}_i^{init}\le Y_i^*|Y_i^*<\hat{Y}_i^{alg}\big)\mathbb{P}[Y_i^*<\hat{Y}_i^{alg}]+\mathbb{P}\big(Y_i^*\le\hat{Y}_i^{init}|\hat{Y}_i^{alg}\le Y_i^*\big)\mathbb{P}[\hat{Y}_i^{alg}\le Y_i^*].\\&=\mathbb{P}\big(\hat{Y}_i^{init}\le Y_i^*\big)\mathbb{P}[Y_i^*<\hat{Y}_i^{alg}]+\mathbb{P}\big(Y_i^*\le\hat{Y}_i^{init}\big)\mathbb{P}[\hat{Y}_i^{alg}\le Y_i^*]\\&=\frac{1}{2}.\end{aligned}$$

The first equality follows from the law of total probability. The second equality holds because $\hat{Y}_i^{init}-Y_i^*$ and $\hat{Y}_i^{alg}-Y_i^*$ are independent from Assumption 2. The last equality follows from the median-unbiased assumption and the assumption that $\mathbb{P}[\hat{Y}_i^{init}=Y_i^*]=\mathbb{P}[\hat{Y}_i^{alg}=Y_i^*]=0$.

Appendix B: Experiment 1: Supplementary Analyses

B.1. Experiment 1: Regression Analysis

In order to test whether participants who observe a mixed exposure set more variably weight the algorithm's recommended predictions across high vs. low impact of private feature products relative to participants who observe a single exposure set (Equation 17), we use a regression model. A t-test is not appropriate for this analysis given that participants in the *Mixed Impact of Private Feature* treatment condition generate two $MedWOA$ observations each while participants in the *Always Low Impact of Private Feature* and *Always High Impact of Private Feature* conditions generate one $MedWOA$ observation each, therefore differences in $MedWOA$ by low vs. high impact of private feature products are within participant for those in the *Mixed Impact of Private Feature* treatment condition and across participants in the other two conditions.

To conduct this analysis, we use a fully-interacted regression model with an outcome of $MedWOA$ regressed on a binary variable indicating the Exposure Set type (0 = Mixed Exposure Set, 1 = Single Exposure Set), interacted with a binary variable indicating the Impact of Private Features (0 = High Impact, 1 = Low Impact). Then our regression model is the following where j indexes each participant and k indexes a set of products they observe (Low or High Impact of Private Feature):

$$MedWOA_j^k=\beta_0+\beta_1\text{Low Impact}_j^k+\beta_2\text{Single Exposure Set}_j+\beta_3\text{Low Impact}_j^k\times\text{Single Exposure Set}_j \qquad (25)$$

Then $\beta_0 = \frac{\sum_{j\in\mathcal{C}_M} MedWOA_j^H}{|\mathcal{C}_M|}$, $\beta_0+\beta_1 = \frac{\sum_{j\in\mathcal{C}_M} MedWOA_j^L}{|\mathcal{C}_M|}$, $\beta_0+\beta_2 = \frac{\sum_{j\in\mathcal{C}_H} MedWOA_j^H}{|\mathcal{C}_H|}$, and $\beta_0+\beta_1+\beta_2+\beta_3 = \frac{\sum_{j\in\mathcal{C}_L} MedWOA_j^L}{|\mathcal{C}_L|}$. Equation 17 then reduces to:

$$(\beta_0+\beta_1)-(\beta_0) \le (\beta_0+\beta_1+\beta_2+\beta_3)-(\beta_0+\beta_2) \tag{26}$$

which is equivalent to determining whether $\beta_3 \ge 0$. In the below table we see the coefficient in the interaction term (corresponding to β_3) is in fact positive and significant.

Table B.1 The Effect of High vs. Low Impact Private Features and Mixed vs. Single Exposure Sets on Participants' Median Weight on Algorithm

Dependent Variable:	$MedWOA$
Model:	(1)
Variables	
(Intercept)	0.4016***
	(0.0352)
Low Impact of Private Feature (Low $\lvert v_i \rvert$)	0.0223
	(0.0220)
Single Exposure Set	-0.2143***
	(0.0454)
Low $\lvert v_i \rvert$ × Single Exposure Set	0.4945***
	(0.0462)
Fit statistics	
Observations	478
R^2	0.22156
Adjusted R^2	0.21664

Clustered (ResponseId) standard-errors in parentheses
*Signif. Codes: ***: 0.01, **: 0.05, *: 0.1*

B.2. Experiment 1: Task Level Analyses

We repeat similar analyses as in 4.2 but conducted at the task level (instead of the participant/product-type level). As dependent variables we use participants' windsorized WOA per task to examine algorithmic advice-taking behavior and final absolute error per task as a measure of prediction error. We add task number fixed effects and cluster all standard errors by participant. We find very similar results when these analyses are conducted on a task level as on a participant/product-type level, observing both a bias towards naïve advice weighting and its negative impact on prediction accuracy. In Table B.2, focusing on the last three columns we find support for Hypothesis 1, observing that (1) across products with a low impact of private feature, participants exposed to a mixed exposure set place less weight on the algorithm's prediction than participants exposed to a single exposure set, (2) across products with a high impact of private feature, participants exposed to a mixed exposure set place more weight on the algorithm's prediction than participants exposed to a single exposure set, and (3) participants exposed to a single exposure set more variably weight the algorithm's predictions across products with low vs. high impact of private feature compared to participants exposed to a mixed exposure set. In Table B.3, focusing on the last two columns we find support for Hypothesis 2, observing that for products with a low impact of private feature and for products with a high impact of private feature, participants who experience a mixed exposure set have larger prediction errors than participants who experience a single exposure set.

Table B.2 Participants' Task-Level Weight on Algorithm Results

Dependent Variable:	WOA (Winsorized)				
Model:	Single Expo Set	Mixed Expo Set	Low $\|v_i\|$	High $\|v_i\|$	All Data
Variables					
Low $\|v_i\|$	0.4172^{***}	0.0258			0.0246
	(0.0332)	(0.0169)			(0.0167)
Single Exposure Set			0.2136^{***}	-0.1791^{***}	-0.1791^{***}
			(0.0355)	(0.0366)	(0.0366)
Low $\|v_i\|$ × Single Exposure Set					0.3926^{***}
					(0.0371)
Fixed-effects					
Task Number	Yes	Yes	Yes	Yes	Yes
Fit statistics					
Observations	4,743	2,364	3,549	3,558	7,107
R^2	0.23962	0.01077	0.06691	0.05338	0.16736
Within R^2	0.23804	0.00103	0.06284	0.04779	0.16582

Clustered (ResponseId) standard-errors in parentheses
*Signif. Codes: ***: 0.01, **: 0.05, *: 0.1*

Table B.3 Participants' Task-Level Prediction Error Results

Dependent Variable:	Final Absolute Error ($\|\hat{y}_{ij}^{final} - y_i\|$)			
Model:	Single Expo Set	Mixed Expo Set	Low $\|v_i\|$	High $\|v_i\|$
Variables				
Low $\|v_i\|$	-32.41^{***}	-28.05^{***}		
	(5.178)	(2.868)		
Single Exposure Set			-12.48^{**}	-7.937
			(5.925)	(5.405)
Fixed-effects				
Task Number	Yes	Yes	Yes	Yes
Fit statistics				
Observations	4,800	2,380	3,616	3,564
R^2	0.09269	0.04946	0.01522	0.00880
Within R^2	0.08946	0.04642	0.01228	0.00400

Clustered (ResponseId) standard-errors in parentheses
*Signif. Codes: ***: 0.01, **: 0.05, *: 0.1*

B.3. Experiment 1: Learning Over Time

In this ex post analysis, we test whether participants show evidence of learning or fatiguing over time. Examining the data from Step 5, we first calculate each participant j's *weight on algorithm* for each product in task number i giving us every WOA_{ij}. We then aggregate these metrics across all four conditions (exposure set × impact of private feature) per task number, giving us the mean WOA for each condition for products 1 through 20. We plot these mean WOA's across time in Figure 7. From this plot, we see some evidence of learning. Participants in the *Always High Impact of Private Feature* condition appear to be weighting the algorithm's recommendations less over time, as indicated by the downward slope of their best fit line. Similarly, participants in the *Mixed Impact of Private Feature* condition across High $|v_i|$ products also appear to be weighting the algorithm's recommendations less over time. For conditions with Low Impact of Private Feature, participants appear to be weighting the algorithm's recommendation slightly more over time, as indicated by positive slopes of their best fit lines.

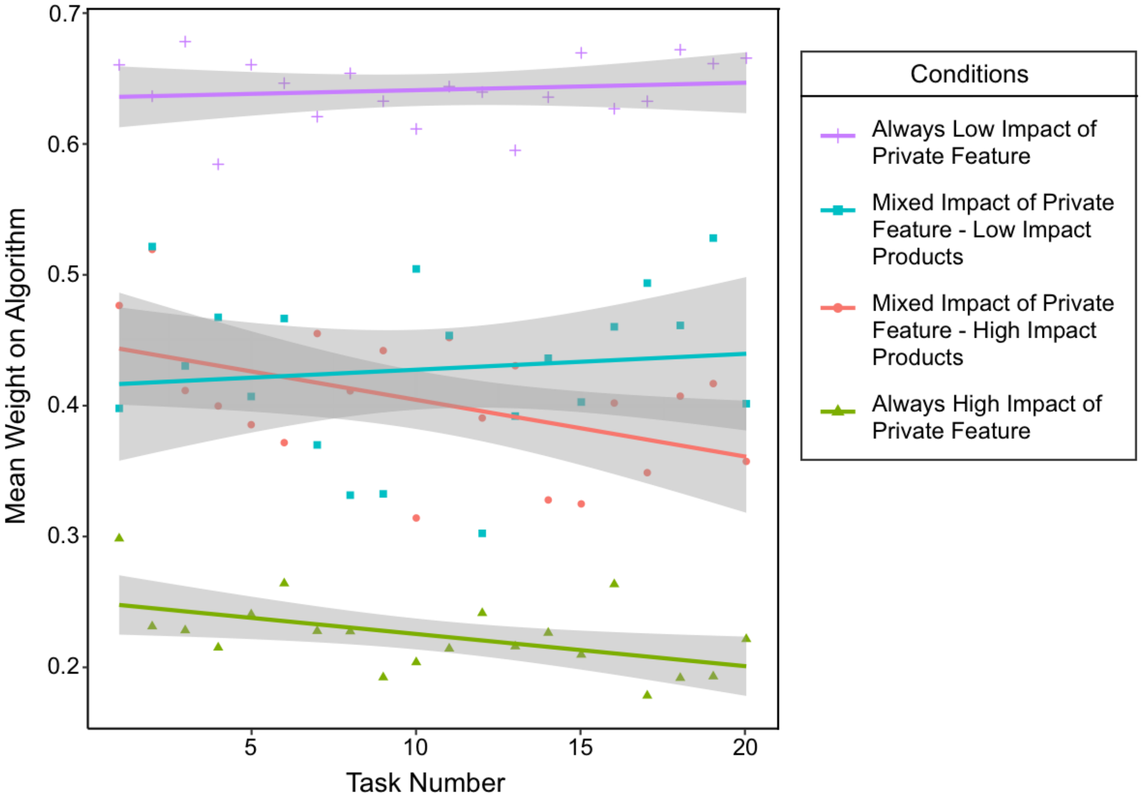


Figure 7 Weight on algorithm results across task number (time). WOA's are averaged (mean) by exposure set, separately for low and high impact of private feature; best fit lines per condition are shown.

We complement this visual evidence with regression analyses. For each of the four conditions, we run a separate regression, looking at the effects of task number on WOA, with participant fixed effects and standard errors clustered by participant. Regression results are shown below in Table B.4. We observe similar patterns where participants in both the single exposure set and mixed exposure set conditions learn to weight the algorithm's recommendations significantly less over time for the products for which they have high impact of private features. For the products where they have low impact of private features, participants in both the single and mixed exposure set conditions directionally appear to learn to weight the algorithm's advice more across time, however this learning effect is not statistically significant.

Table B.4 The effects of task number (time) on WOA per condition.

Dependent Variable:	WOA (Windsorized)			
Model:	Single Expo Set Low $\lvert v_i \rvert$	Mixed Expo Set Low $\lvert v_i \rvert$	Single Expo Set High $\lvert v_i \rvert$	Mixed Expo Set High $\lvert v_i \rvert$
Variables				
Task Number	0.0006	0.0017	-0.0024**	-0.0047**
	(0.0015)	(0.0019)	(0.0011)	(0.0019)
Fixed-effects				
ResponseId	Yes	Yes	Yes	Yes
Fit statistics				
Observations	2,327	1,222	2,416	1,142
R^2	0.43508	0.51945	0.51007	0.55711
Within R^2	0.00016	0.00114	0.00314	0.00966

Clustered (ResponseId) standard-errors in parentheses
*Signif. Codes: ***: 0.01, **: 0.05, *: 0.1*

Given that participants are learning across both the *Always High Impact of Private Features* and *Mixed Impact of Private Features* conditions, and given that the rate of learning observed is very low, this learning effect is not enough to fully counteract naïve advice weighting. We analyze this dynamic by repeating the regression shown in Table B.2 in the far right column separately for tasks 1-5, 6-10, 11-15, and 16-20 and observing how the coefficients on the interaction term change. Results are shown in Table B.5. We observe that in all four columns, the interaction term is large and significant indicating that participants in the single exposure set conditions are weighting the algorithm's recommendations much more variably across low and high impact of private feature products relative to participants in the mixed exposure set condition. While this effect is smaller across the last five tasks relative to the first five tasks (the coefficient on the interaction term decreases) it is still large and significant even across tasks 16-20. This indicates that while participants in the mixed exposure set are more variably weighting the algorithm's recommendations across time depending on the impact of private features for each product, they are still biased towards naïve advice weighting, not placing as variable weights as a compared to what a more sophisticated advice weighter would do.

Table B.5 The Effect of High vs. Low Impact Private Features and Mixed vs. Single Exposure Sets on Participants' Weight on Algorithm in subsets of tasks over time.

Dependent Variable:	WOA (Windsorized)			
Task Numbers Subset:	1-5	6-10	11-15	16-20
Variables				
Low $\lvert v_i \rvert$	0.0016	0.0036	0.0123	0.0805***
	(0.0338)	(0.0331)	(0.0315)	(0.0300)
Single Exposure Set	-0.1984***	-0.1768***	-0.1642***	-0.1771***
	(0.0422)	(0.0429)	(0.0422)	(0.0428)
Low $\lvert v_i \rvert$ × Single Exposure Set	0.3998***	0.4062***	0.4032***	0.3615***
	(0.0489)	(0.0501)	(0.0489)	(0.0481)
Fixed-effects				
Task Number	Yes	Yes	Yes	Yes
Fit statistics				
Observations	1,774	1,778	1,773	1,782
R^2	0.15781	0.16088	0.16637	0.18591
Within R^2	0.15579	0.15987	0.16590	0.18521

Clustered (ResponseId) standard-errors in parentheses
*Signif. Codes: ***: 0.01, **: 0.05, *: 0.1*

B.4. Experiment 1: Robustness without Windsorizing WOA

In this set of ex post analyses, we verify our results are robust when we do not windsorize WOA between 0 and 1. We first repeat the analyses in Section 4.2.1 now defining

$$\text{unwindsor}WOA_{ij} = \frac{\hat{y}_{ij}^{final} - \hat{y}_{ij}^{init}}{\hat{y}_i^{alg} - \hat{y}_{ij}^{init}}. \tag{27}$$

We first test Equation (15), this time replacing all WOA_{ij} with unwindsorWOA_{ij}. Considering only products with a low impact of private feature, we again find participants exposed to a mixed exposure set place less weight on the algorithm than participants exposed to a single exposure set ($t(226.62) = -6.124, p < 0.0001$). We next test Equation (16), this time replacing all WOA_{ij} with unwindsorWOA_{ij}. Considering only products with a high impact of private feature, we again find participants exposed to a mixed exposure set place more weight on the algorithm than participants exposed to a single exposure set ($t(198.44) = 3.076, p = 0.002$). Finally we test Equation (17). When we regress the median unwindsorWOA_{ij} for each participant on impact of private feature interacted with exposure set, clustering standard errors by participant, we continue to find a significant positive coefficient on the interaction term ($\beta = 0.469, p < 0.0001$). We continue to find that for participants who experience a mixed exposure set, their average Median unwindsorWOA_{ij} is not significantly different for products with a low vs. high impact of private feature ($t(214.90) = -0.750, p = 0.454$).

We further repeat our task-level analyses as in Appendix B.2, this time using the unwindsorized weight on algorithm measure unwindsorWOA_{ij}. Given that task-level analyses are heavily biased by outlier values of unwindsorWOA_{ij}, we minimally trim our data to remove observations where unwindsorWOA_{ij} is greater than the top 99.5 percentile value of unwindsorWOA_{ij} (7.227) and less than the bottom 0.5 percentile value (-8.333). This leaves us with 7,036 observations out of the original 7,107 tasks for which we have non-null values, preserving 99% of our data.

We run regressions with unwindsorWOA_{ij} as a dependent variable, adding task number fixed effects and clustering all standard errors by participant. Results are shown in Table B.6. Focusing on the last three columns, we again find that for products with a low impact of private feature, participants faced with a single exposure set place significantly more weight on the algorithm than participants faced with a mixed exposure set; for products with a high impact of private feature, participants faced with a single exposure set place significantly less weight on the algorithm than participants faced with a mixed exposure set; and across all participants, participants faced with a single exposure set weight the algorithm much more variably across products with a low vs. high impact of private feature relative to participants faced with a mixed exposure set.

Table B.6 Task-Level Results using Unwindsorized Weight on Algorithm.

Dependent Variable:	Unwindsorized WOA				
Model:	Single Expo Set	Mixed Expo Set	Low $\|v_i\|$	High $\|v_i\|$	All Data
Variables					
Low $\|v_i\|$	0.4765^{***}	0.0514^{*}			0.0539^{*}
	(0.0443)	(0.0283)			(0.0281)
Single Exposure Set			0.2603^{***}	-0.1624^{***}	-0.1625^{***}
			(0.0422)	(0.0552)	(0.0553)
Low $\|v_i\|$ × Single Exposure Set					0.4225^{***}
					(0.0523)
Fixed-effects					
Task Number	Yes	Yes	Yes	Yes	Yes
Fit statistics					
Observations	4,696	2,340	3,507	3,529	7,036
R^2	0.09420	0.01185	0.04238	0.01305	0.06527
Within R^2	0.09164	0.00114	0.03717	0.00773	0.06350

Clustered (ResponseId) standard-errors in parentheses
*Signif. Codes: ***: 0.01, **: 0.05, *: 0.1*

B.5. Experiment 1: Drivers of WOA

In this ex post descriptive analysis we examine how various factors might play a role in determining how participants choose a weight to place on the algorithm's predictions (λ_{ij}) across each task. Past research shows that people rely less on algorithmic predictions after observing the algorithm make errors (Dietvorst et al. 2015), therefore one of the factors we examine is the algorithm's lagged forecasting error on the previously observed product (algorithm's lagged absolute error). We further test whether there is an interaction effect between treatment condition and lagged algorithm error. Similarly, as this research highlights that people are more forgiving of their own forecasting errors, we also control for participants' lagged forecasting error (participant's lagged absolute error). Additionally, we also study the role of participants' lagged weight on the algorithm, as a high weight on this factor would lend support to naïve advice weighting behavior given that it implies participants' weight on the algorithm are relatively constant from task to task. Finally, we examine the effect of the current impact of private features ($|v_i|$), as a low weight on this factor would further indicate participants are naïve advice weighting and not changing their weight on the algorithm to align with their current impact of private features.

To conduct this analysis, we run several regression models analyzing our data on a task level, focusing on task numbers 2 through 20 in order to have lagged metrics on what the participant observes on the previous task. Each regression model uses each participant's weight on the algorithm for a given task WOA_{ij} as a dependent variable. Additionally, across all models we control for the participant's experimental condition as well as for the task number. Standard errors are clustered by participant. Regression results are presented in Table B.7.

Table B.7 Task-Level Factors Determining Participants' Current Weight on Algorithm.

Dependent Variable:	WOA (Windsorized)			
Model:	(1)	(2)	(3)	(4)
Variables				
Always High $\|v_i\|$	0.2832^{***}	0.2891^{***}	0.1497^{***}	0.1690^{***}
	(0.0290)	(0.0354)	(0.0172)	(0.0198)
Mixed $\|v_i\|$	0.1724^{***}	0.1646^{***}	0.0777^{***}	0.0682^{***}
	(0.0360)	(0.0429)	(0.0190)	(0.0200)
Always Low $\|v_i\|$	0.3727^{***}	0.3762^{***}	0.1728^{***}	0.1546^{***}
	(0.0357)	(0.0430)	(0.0239)	(0.0255)
Lagged Algorithm Abs. Error	-0.0006^{***}	-0.0006^{**}	-0.0006^{***}	-0.0006^{***}
	(0.0002)	(0.0003)	(0.0002)	(0.0002)
Lagged Participant Abs. Error	-0.0002	-0.0002	6.14×10^{-6}	6.73×10^{-6}
	(0.0003)	(0.0003)	(0.0001)	(0.0001)
Task Number	-0.0007	-0.0007	-1.74×10^{-5}	-2.65×10^{-5}
	(0.0008)	(0.0008)	(0.0005)	(0.0005)
Mixed $\|v_i\|$ × Lagged Algorithm Abs. Error		0.0001		
		(0.0003)		
Always Low $\|v_i\|$ × Lagged Algorithm Abs. Error		-0.0018		
		(0.0024)		
Lagged WOA (Windsorized)			0.5011^{***}	0.5011^{***}
			(0.0227)	(0.0227)
$\|v_i\|$				-0.0002
				(0.0001)
Fit statistics				
Observations	6,754	6,754	6,687	6,687
R^2	0.17021	0.17031	0.37857	0.37881
Adjusted R^2	0.16960	0.16945	0.37801	0.37816

Clustered (ResponseId) standard-errors in parentheses
*Signif. Codes: ***: 0.01, **: 0.05, *: 0.1*

Consistent with the literature, in column (1) we observe that there is a significant (albeit very small) effect of the lagged algorithm's error on participants' current weight on the algorithm, with a larger algorithm forecasting error on the previous task leading to participants weighting the algorithm's predictions less for the current task. Concurrently, there is no significant effect of the participant's own lagged forecasting error on their current weight on the algorithm. In column (2), we examine whether there is an interaction effect between the algorithm's lagged error and treatment condition and find no evidence for this, indicating that participants treat the algorithm's past forecasting errors uniformly across treatment conditions when deciding how much to currently weight the algorithm's predictions. In column (3), we further examine the effects of participants' lagged weight on the algorithm and observe that this plays a very large and significant role in determining their current weight on the algorithm. This provides evidence that participants place relatively constant weights on the algorithm from one task to the next. Finally in column (4) we find no significant evidence that the impact of private features on the current task plays a role in determining participants' current weight on the algorithm, indicating that they are ignoring this information and not differentially placing weights on the algorithm's predictions depending on this value.

B.6. Experiment 1: Mediation Analyses

In this ex post mediation analysis, we test and find support that the differences in $MedWOA$ mediate the observed differences in prediction error. We run separate mediation analyses for products with low vs. high impact of private features and find evidence that for both Low $|v_i|$ and High $|v_i|$ products, the indirect effect of exposure set on prediction error via $MedWOA$ is statistically significant ($p < 0.0001$). This analysis provides additional support for the mechanism that participants perform systematically worse in the *Mixed Impact of Private Feature* condition because they suffer from an overly-constant weight-on-algorithm.

Table B.8 Mediation Analysis for Low Impact of Private Feature products

	Causal Mediation Analysis of Exposure Set on $RMedSE$ via $MedWOA$			
	Estimate	95% CI Lower	95% CI Upper	p-value
Average Causal Mediation Effect	-13.226***	-20.03	-7.31	<2e-16
Average Direct Effect	2.427	-9.23	14.30	0.724
Total Effect	-10.799**	-22.11	-0.11	0.047
Proportion Mediated	1.225**	0.954	595.51	0.047
Sample Size Used: 238				

Nonparametric Bootstrap Confidence Intervals with the BCa Method and 5000 simulations
*Signif. Codes: ***: 0.01, **: 0.05, *: 0.1*

Table B.9 Mediation Analysis for High Impact of Private Feature products

	Causal Mediation Analysis of Exposure Set on $RMedSE$ via $MedWOA$			
	Estimate	95% CI Lower	95% CI Upper	p-value
Average Causal Mediation Effect	-7.139***	-12.42	-3.42	<2e-16
Average Direct Effect	-2.487	-15.23	8.96	0.712
Total Effect	-9.625*	-20.26	1.29	0.072
Proportion Mediated	0.742*	-2360.58	0.36	0.072
Sample Size Used: 240				

Nonparametric Bootstrap Confidence Intervals with the BCa Method and 5000 simulations
*Signif. Codes: ***: 0.01, **: 0.05, *: 0.1*

B.7. Experiment 1: Step 3 Predictions without the Algorithm

To examine whether participants qualitatively self-reported that they noticed the treatment condition they were in, participants were asked to evaluate their performance relative to the algorithm's after Step 4, where participants selected a number from 1-5. A one-way ANOVA test was performed to compare the effect of the 3 treatment conditions on self-reported relative performance. The one-way ANOVA revealed that there was a statistically significant difference in self-reported relative prediction performance between at least two groups ($F(2,356) = 71.57, p < 0.0001$). The table below reports the results of participants' and the algorithm's actual prediction error ($RMedSE$) across conditions in Step 3 (Demand Predictions without Algorithm), finding significant differences in performance between participants and the algorithm on average in directions that align with participants' self evaluations.

Table B.10 Participants' vs. Algorithm's mean prediction error in Step 3 by Treatment Condition

	Participants' $RMedSE$	Algorithm's $RMedSE$	Paired t-test
Always Low $\|v_i\|$	M=34.167 (SD=43.471)	M=4.317 (SD=1.020)	$t(118) = 7.506, p < 0.0001$
Always High $\|v_i\|$	M=49.639 (SD=41.737)	M=75.303 (SD=7.828)	$t(120) = -6.699, p < 0.0001$
Mixed $\|v_i\|$	M=50.491 (SD=50.916)	M=27.191 (SD=20.554)	$t(118) = 4.502, p < 0.0001$

B.8. Experiment 1: Initial Predictions ($\hat{y}_{ij}^{init}$) Without the Algorithm in Step 3 vs. Step 5

One might be concerned that participants do not seriously answer the "initial" prediction questions in Step 5 because they are unincentivized and precede algorithmic advice. However, performance with these initial predictions in Step 5 are not significantly different from the predictions without the algorithm in Step 3, which are incentivized. In other words, we do not find evidence that participants treat these "initial" predictions preceding algorithmic advice any differently than if they were predicting demand without awareness of the algorithm.

Table B.11 Participants' mean initial prediction error in Step 5 vs. initial prediction error in Step 3

	Step 5 Initial $RMedSE$	Step 3 $RMedSE$	Paired t-test
Low $\|v_i\|$ & Single Expo Set	M=32.716 (SD=42.318)	M=34.157 (SD=43.471)	$t(118) = -0.741, p = 0.460$
High $\|v_i\|$ & Single Expo Set	M=47.873 (SD=41.455)	M=49.639 (SD=41.737)	$t(120) = -0.737, p = 0.4623$
Low $\|v_i\|$ & Mixed Expo Set	M=50.861 (SD=58.780)	M=48.503 (SD=52.360)	$t(118) = 0.913, p = 0.363$
High $\|v_i\|$ & Mixed Expo Set	M=53.326 (SD=51.057)	M=54.528 (SD=56.316)	$t(118) = -0.361, p = 0.7191$

B.9. Experiment 1: Time to Make Predictions Results

We collected data on the time it took each participant to complete each prediction task, including the time taken to make each initial prediction without the algorithm and the time taken to make each final updated prediction. In general, participants' spend longer making initial predictions for High $|v_i|$ products versus Low $|v_i|$ products. However, given an impact of private feature (low vs. high), there are no significant differences across exposure set/treatment conditions.

Table B.12 Participants' mean prediction time in Step 5 for initial ($\hat{y}_{ij}^{init}$) and final ($\hat{y}_{ij}^{final}$) predictions

	Initial Prediction Time (sec)	Final Prediction Time (sec)
Low $\lvert v_i\rvert$ & Single Expo Set	M=11.556 (SD=9.771)	M=7.948 (SD=14.151)
High $\lvert v_i\rvert$ & Single Expo Set	M=15.766 (SD=18.117)	M=7.142 (SD=8.059)
Low $\lvert v_i\rvert$ & Mixed Expo Set	M=11.054 (SD=9.393)	M=6.937 (SD=5.712)
High $\lvert v_i\rvert$ & Mixed Expo Set	M=15.306 (SD=24.300)	M=6.912 (SD=5.140)

Timings are averaged per participant and impact of private feature and means are taken across conditions

Table B.13 Unpaired t-test comparisons of Step 5 prediction time

Sample A	Sample B	Initial prediction time t-test	Final prediction time t-test
Low $\lvert v_i\rvert$ & Single Expo Set	Low $\lvert v_i\rvert$ & Mixed Expo Set	$t(235.63) = 0.404$ $p = 0.687$	$t(155.46) = 0.723$ $p = 0.471$
High $\lvert v_i\rvert$ & Single Expo Set	High $\lvert v_i\rvert$ & Mixed Expo Set	$t(218.17) = 0.166$ $p = 0.868$	$t(204.25) = 0.264$ $p = 0.792$
Low $\lvert v_i\rvert$ & Single Expo Set	High $\lvert v_i\rvert$ & Single Expo Set	$t(185.02) = -2.246$ $p = 0.0259$	$t(186.62) = 0.541$ $p = 0.589$
Low $\lvert v_i\rvert$ & Mixed Expo Set	High $\lvert v_i\rvert$ & Mixed Expo Set	$t(152.49) = -1.780$ $p = 0.0770$	$t(233.42) = 0.0352$ $p = 0.972$

Appendix C: Experiment 2: Supplementary Analyses

C.1. Experiment 2: Advice-Weighting Region Analysis

The best prediction benchmark fell within participants' advice-weighting regions at proportions similar to in Study 1. Furthermore, the proportion of instances where the best prediction benchmark was within the advice-weighting region did not significantly vary across treatment conditions, with average proportions of 46.7% under *No Transparency* as well as *Training Data Transparency*, and 47.9% under *Feature Transparency*. A one-way ANOVA of the proportion of instances where the best prediction benchmark requires advice weighting showed no statistically significant differences across treatment conditions ($F(2,519) = 0.001, p = 0.979$).

Considering only the subset of products for which the best prediction benchmark is outside the advice-weighting region, participants under *Feature Transparency* make final predictions outside of the advice-weighting region in a significantly higher proportion of instances (20.6%) compared to participants with *No Transparency* (9.64%) or *Training Data Transparency* (9.46%) (two-sided t-tests: $t(275.39) = 4.917, p < 0.0001$; $t(293.79) = 4.864, p < 0.0001$). Over all products, participants' final predictions fell within their advice-weighting regions in 91.8% of instances under *No Transparency*, 92.5% of instances under *Training Data Transparency*, and 86.4% of instances under *Feature Transparency*. Two-sided t-tests reveal this proportion is significantly lower under *Feature Transparency* than under both *No Transparency* and *Training Data Transparency* ($t(317.26) = 3.559, p < 0.001$; $t(336.227) = 3.873, p < 0.001$).

C.2. Experiment 2: Demand Prediction Strategy Text Analysis

Participants were asked at the end of the study to optionally answer the following question: "Was there a particular strategy you used to make your own demand forecasts? Did you have a specific method for using the algorithm's forecasts? Feel free to let us know any strategies you may have used." We chose to examine 3 strategies that were commonly repeated across responses. For each of these three strategies, we created a dictionary of word stems corresponding to the strategy. If a participant's response contained one or more of the word stems corresponding to a strategy's dictionary, they were marked as having followed that strategy. Participants could therefore follow multiple strategies. The strategies examined were:

1. *Averaging*: This strategy corresponded to naïve advice weighting, where participants took a constantly weighted average between their initial prediction and the algorithm's recommended prediction to make a final prediction. The dictionary for this strategy was: *averag, combin, between, middl*
2. *Adjusting*: This strategy mapped to anchoring on the algorithm's recommended prediction and adjusting it using only private features to make a final prediction. The dictionary for this strategy was: *adjust, modif, adapt, revis*
3. *Guessing*: This strategy corresponded to using some amount of guessing to make a final prediction. The dictionary for this strategy was: *guess, gut, random*

Table C.1 Percentage of participants in each treatment condition who mention words corresponding to a particular demand prediction strategy

Transparency Type	Mentions Averaging	Mentions Adjusting	Mentions Guessing
No Transparency	M=16.3%, SD=37.0	M=11.6%, SD=32.2	M=27.3%, SD=44.7
Feature Transparency	M=11.1%, SD=31.5	M=15.2%, SD=36.0	M=23.4%, SD=42.5
Training Data Transparency	M=15.2%, SD=36.0	M=11.8%, SD=32.4	M=24.7%, SD=43.3

Table C.2 The Effects of Each Self-Reported Advice-Taking Strategy on Participants' Prediction Error ($RMedSE$ across All Products) and Within-Participant Variability in Weighting the Algorithm (Standard Deviation of WOA across All Products; Difference in $MedWOA$ across Low vs. High Impact of Private Feature Products)

Dependent Variables:	$RMedSE$	SD(WOA)	$MedWOA^L - MedWOA^H$
Model:	(1)	(2)	(3)
Variables			
(Intercept)	21.31***	0.2735***	0.1422***
	(0.9087)	(0.0060)	(0.0179)
Mentions Adjusting	-4.582**	0.0325**	0.1821***
	(2.155)	(0.0143)	(0.0424)
Mentions Averaging	-0.8201	0.0104	0.0115
	(2.074)	(0.0138)	(0.0408)
Mentions Guessing	3.604**	0.0191*	-0.0676**
	(1.670)	(0.0111)	(0.0328)
Fit statistics			
Observations	521	521	521
R^2	0.01685	0.01767	0.04086
Adjusted R^2	0.01115	0.01197	0.03529

IID standard-errors in parentheses
*Signif. Codes: ***: 0.01, **: 0.05, *: 0.1*

C.3. Experiment 2: APAE

We define a participant's *absolute percent adjustment error* for task i as

$$APAE_i = \left| \frac{(\hat{y}_i^{final} - \hat{y}_i^{alg}) - (y_i^* - \hat{y}_i^{alg})}{y_i^* - \hat{y}_i^{alg}} \right| \tag{28}$$

where y_i^* is the best prediction benchmark prediction (the true demand minus the random error term). Intuitively, $APAE$ is how far away a participant's adjustment (from the algorithm's recommendation) is from the optimal adjustment. It is zero when the adjustment is optimal and becomes more positive as the adjustment is further from optimal.

Consistent with the patterns with $RMedSE$, we find that participants' median $APAE$ are significantly lower in *Feature Transparency* than in *No Transparency*. However, while participants with *Feature Transparency* do have a significantly lower $APAE$ relative to participants' with *Training Data Transparency* for High $|v_i|$ products, they do not have a significantly lower $APAE$ for Low $|v_i|$ products. This indicates that although *Training Data Transparency* does not mitigate naïve advice weighting, it may increase participants' use of the algorithm for products with both low and high impact of private features. While this will lead to more beneficial participant adjustments for Low $|v_i|$ products where relying on the algorithm is helpful, this increased adherence to the algorithm across the board will not result in better adjustments for High $|v_i|$ products for which relying too heavily on the algorithm can be harmful.

Table C.3 Means of Participants' median $APAE$ separated by low vs. high impact of private features and across all products

	Low $\|v_i\|$ products	High $\|v_i\|$ products	All products
No Transparency	M=4.314 SD=4.056	M=0.531, SD=0.280	M=1.018, SD=0.924
Feature Transparency	M=2.943, SD=2.922	M=0.441, SD=0.279	M=0.874, SD=0.654
Training Transparency	M=3.374, SD=3.716	M=0.539, SD=0.371	M=1.101, SD=1.354

Table C.4 T-test Comparisons of Participants' median $APAE$ separated by low vs. high impact of private features and across all products

	Low $\|v_i\|$ products	High $\|v_i\|$ products	All products
Feature vs. No Transparency	$t(305.34) = 3.561$ $p = 0.000214$	$t(341.00) = 2.967$ $p = 0.00161$	$t(302.71) = 1.656$ $p = 0.0494$
Training vs. No Transparency	$t(336.97) = 2.239$ $p = 0.0129$	$t(328.60) = -0.251$ $p = 0.599$	$t(308.02) = 0.0668$ $p = 0.473$
Feature vs. Training Data Transparency	$t(328.39) = 1.196$ $p = 0.116$	$t(328.04) = 2.803$ $p = 0.00268$	$t(253.26) = 1.192$ $p = 0.117$

C.4. Experiment 2: Time to Make Predictions

Across the three transparency treatment conditions there are no significant differences in the time taken to make initial predictions both for low impact and high impact of private feature products. Similarly, for High $|v_i|$ products, there is no significant difference in time taken to make final predictions across treatment conditions. The difference in time to make final predictions for Low $|v_i|$ products is significant, with participants taking longer to make their updated final predictions under *Feature Transparency*. This may be partly due to participants with *Feature Transparency* being more likely to follow an "anchor on the algorithm and adjust it" strategy, resulting in longer times to make these adjusted final predictions, as opposed to using a simpler advice weighting (averaging) heuristic.

Table C.5 Participants' mean prediction time in Step 5 for initial and final predictions

	Initial Prediction Time (sec)	Final Prediction Time (sec)
No Transparency & Low $\lvert v_i \rvert$	M=12.312 (SD=11.461)	M=5.100 (SD=2.956)
No Transparency & High $\lvert v_i \rvert$	M=16.579 (SD=27.084)	M=7.220 (SD=5.162)
Feature Transparency & Low $\lvert v_i \rvert$	M=13.548 (SD=12.391)	M=6.780 (SD=4.092)
Feature Transparency & High $\lvert v_i \rvert$	M=15.119 (SD=15.644)	M=7.272 (SD=4.163)
Training Data Transparency & Low $\lvert v_i \rvert$	M=12.646 (SD=10.962)	M=5.947 (SD=3.181)
Training Data Transparency & High $\lvert v_i \rvert$	M=13.691 (SD=12.982)	M=7.156 (SD=4.524)

Timings are averaged per participant and impact of private feature and means are taken across conditions

Table C.6 One-way ANOVA tests of prediction time across 3 transparency treatment conditions

Impact of Private Feature	Timing Metric	One-way ANOVA
All products	Initial Predictions	$F(2,518) = 0.387, p = 0.679$
Low $\lvert v_i \rvert$	Initial Predictions	$F(2,518) = 0.521, p = 0.594$
High $\lvert v_i \rvert$	Initial Predictions	$F(2,518) = 0.960, p = 0.384$
All products	Final Predictions	$F(2,518) = 0.884, p = 0.414$
Low $\lvert v_i \rvert$	Final Predictions	$F(2,518) = 3.177, p = 0.0425$
High $\lvert v_i \rvert$	Final Predictions	$F(2,518) = 0.028, p = 0.973$

Appendix D: Experiment 3: Supplementary Analyses

D.1. Experiment 3: Prediction Error by Low and High Impact of Private Feature Products

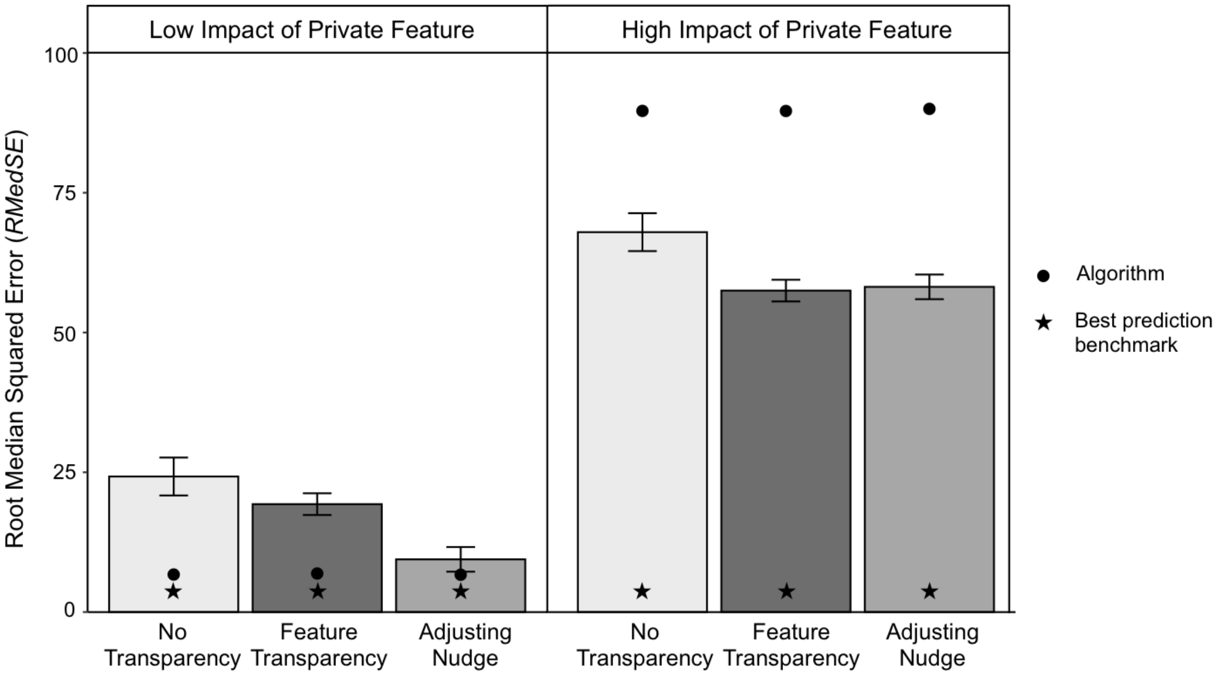


Figure 8 Root median squared error results are averaged (mean) by transparency type, separately for low and high impact of private feature; standard error bars are shown. Additionally, the mean ***RMedSE*** is reported for the algorithm and the best prediction benchmark.

D.2. Experiment 3: Rates of Algorithm Prediction Adjustments in the Correct Direction

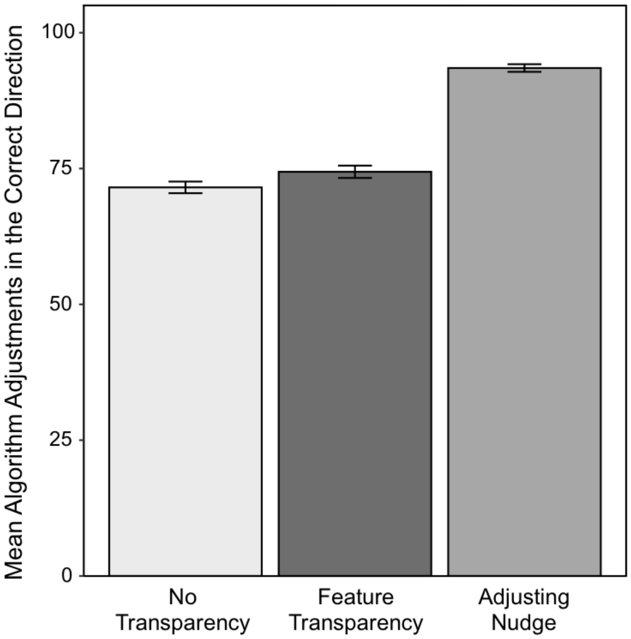
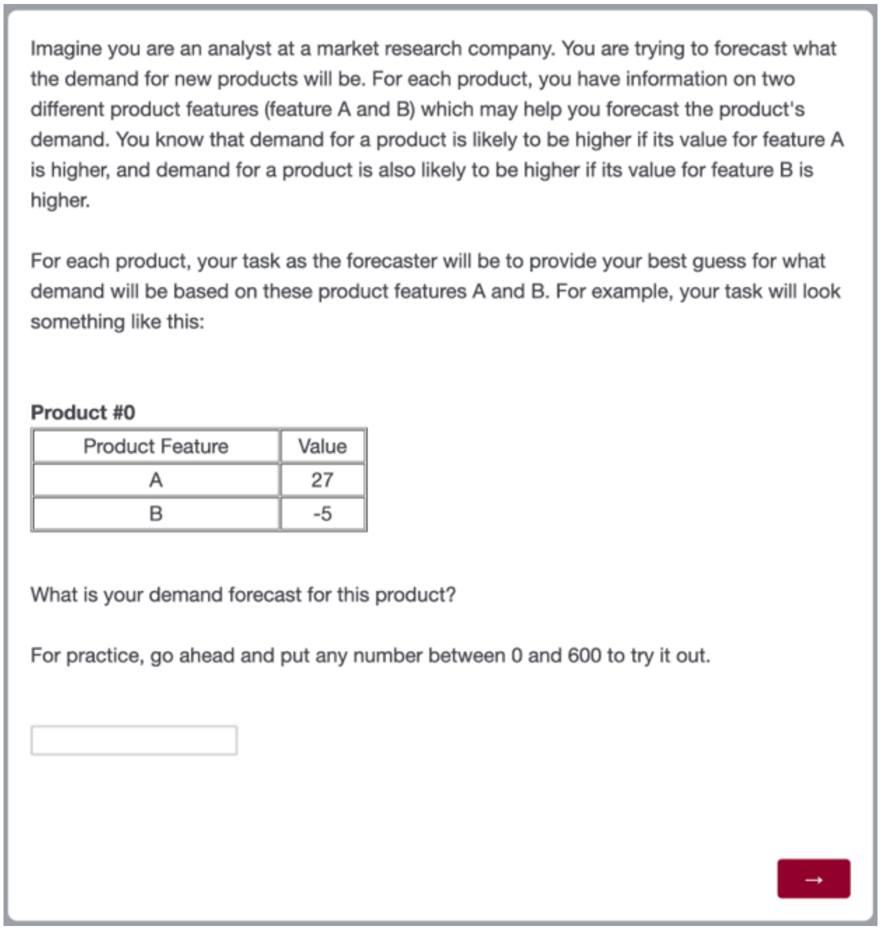


Figure 9 Each participant's percentage of algorithm prediction adjustments in the correct direction are averaged (mean) by transparency type; standard error bars are shown.

Appendix E: Experiment 1: Participant Experience

E.1. Step 1: Instructions and Comprehension Checks


Imagine you are an analyst at a market research company. You are trying to forecast what the demand for new products will be. For each product, you have information on two different product features (feature A and B) which may help you forecast the product's demand. You know that demand for a product is likely to be higher if its value for feature A is higher, and demand for a product is also likely to be higher if its value for feature B is higher.

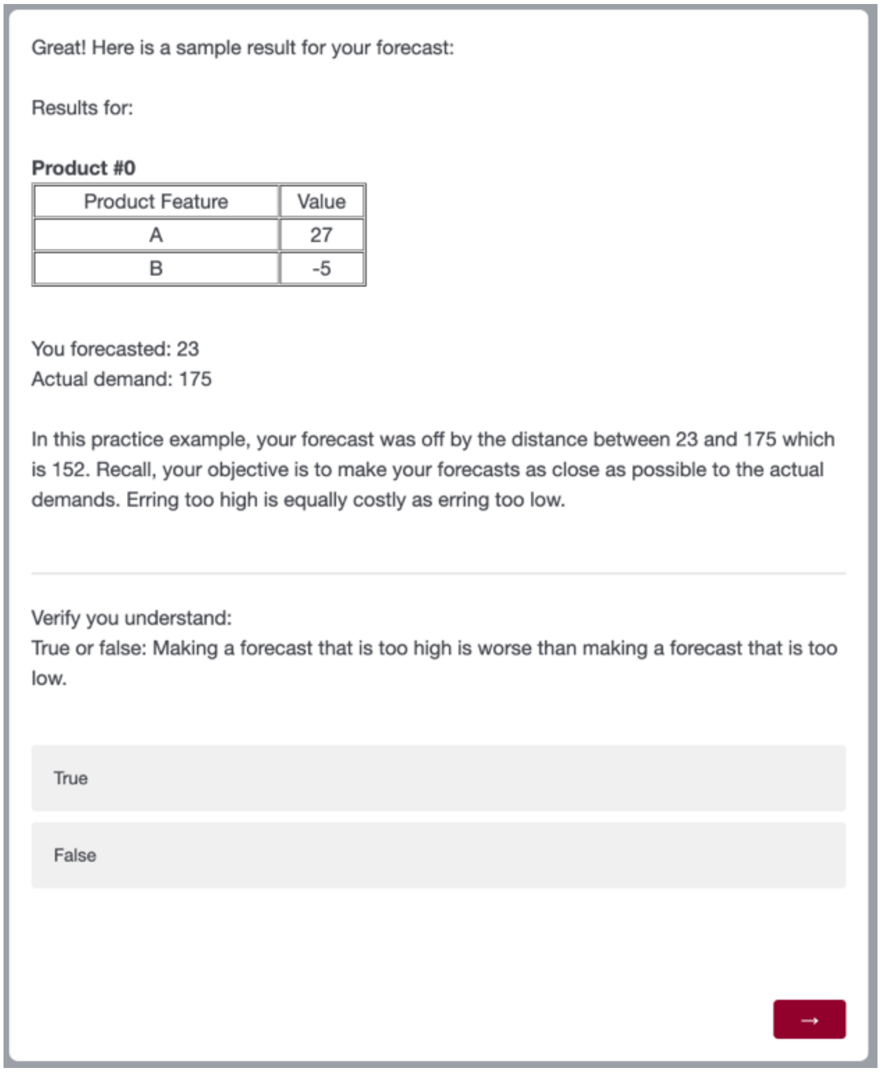
For each product, your task as the forecaster will be to provide your best guess for what demand will be based on these product features A and B. For example, your task will look something like this:

Product #0

Product Feature	Value
A	27
B	-5

What is your demand forecast for this product?

For practice, go ahead and put any number between 0 and 600 to try it out.



Great! Here is a sample result for your forecast:

Results for:

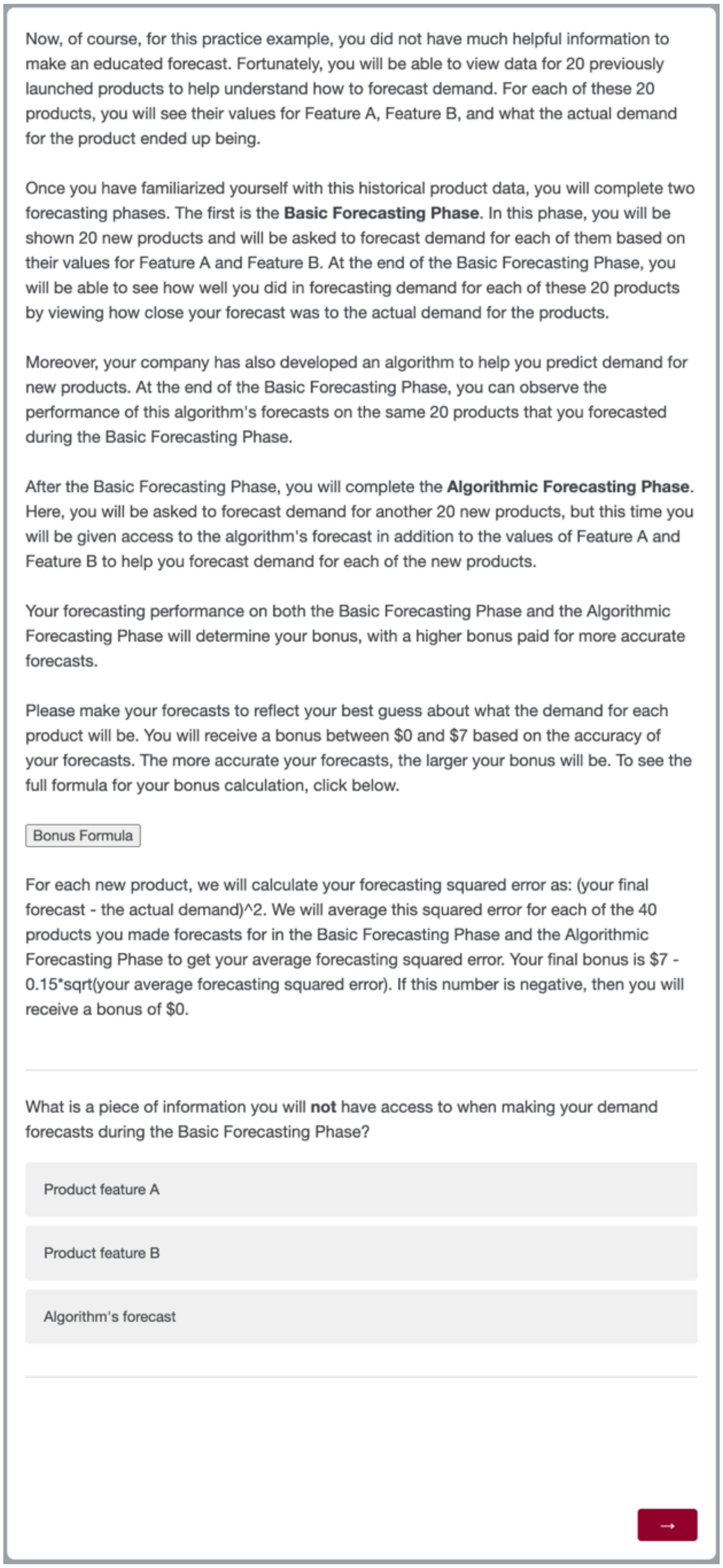
Product #0

Product Feature	Value
A	27
B	-5

You forecasted: 23
Actual demand: 175

In this practice example, your forecast was off by the distance between 23 and 175 which is 152. Recall, your objective is to make your forecasts as close as possible to the actual demands. Erring too high is equally costly as erring too low.

Verify you understand:
True or false: Making a forecast that is too high is worse than making a forecast that is too low.



Now, of course, for this practice example, you did not have much helpful information to make an educated forecast. Fortunately, you will be able to view data for 20 previously launched products to help understand how to forecast demand. For each of these 20 products, you will see their values for Feature A, Feature B, and what the actual demand for the product ended up being.

Once you have familiarized yourself with this historical product data, you will complete two forecasting phases. The first is the **Basic Forecasting Phase**. In this phase, you will be shown 20 new products and will be asked to forecast demand for each of them based on their values for Feature A and Feature B. At the end of the Basic Forecasting Phase, you will be able to see how well you did in forecasting demand for each of these 20 products by viewing how close your forecast was to the actual demand for the products.

Moreover, your company has also developed an algorithm to help you predict demand for new products. At the end of the Basic Forecasting Phase, you can observe the performance of this algorithm's forecasts on the same 20 products that you forecasted during the Basic Forecasting Phase.

After the Basic Forecasting Phase, you will complete the **Algorithmic Forecasting Phase**. Here, you will be asked to forecast demand for another 20 new products, but this time you will be given access to the algorithm's forecast in addition to the values of Feature A and Feature B to help you forecast demand for each of the new products.

Your forecasting performance on both the Basic Forecasting Phase and the Algorithmic Forecasting Phase will determine your bonus, with a higher bonus paid for more accurate forecasts.

Please make your forecasts to reflect your best guess about what the demand for each product will be. You will receive a bonus between $0 and $7 based on the accuracy of your forecasts. The more accurate your forecasts, the larger your bonus will be. To see the full formula for your bonus calculation, click below.

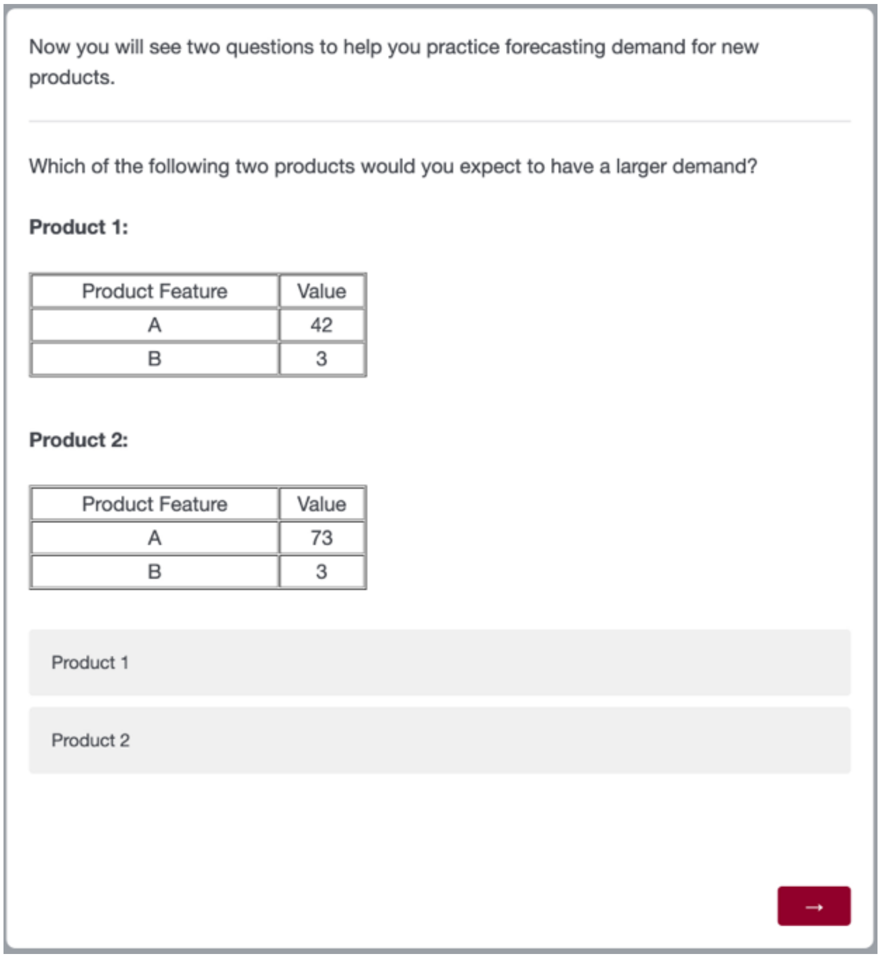
For each new product, we will calculate your forecasting squared error as: (your final forecast - the actual demand)^2. We will average this squared error for each of the 40 products you made forecasts for in the Basic Forecasting Phase and the Algorithmic Forecasting Phase to get your average forecasting squared error. Your final bonus is $7 - 0.15*sqrt(your average forecasting squared error). If this number is negative, then you will receive a bonus of $0.

What is a piece of information you will **not** have access to when making your demand forecasts during the Basic Forecasting Phase?

Product feature A

Product feature B

Algorithm's forecast



Now you will see two questions to help you practice forecasting demand for new products.

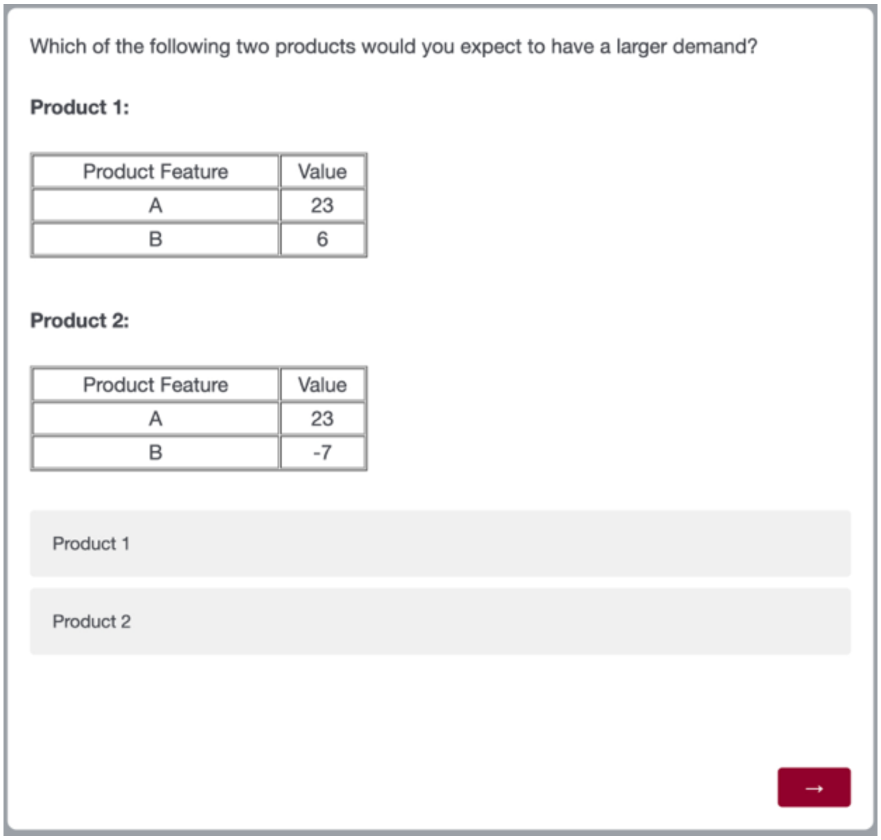
Which of the following two products would you expect to have a larger demand?

Product 1:

Product Feature	Value
A	42
B	3

Product 2:

Product Feature	Value
A	73
B	3



Which of the following two products would you expect to have a larger demand?

Product 1:

Product Feature	Value
A	23
B	6

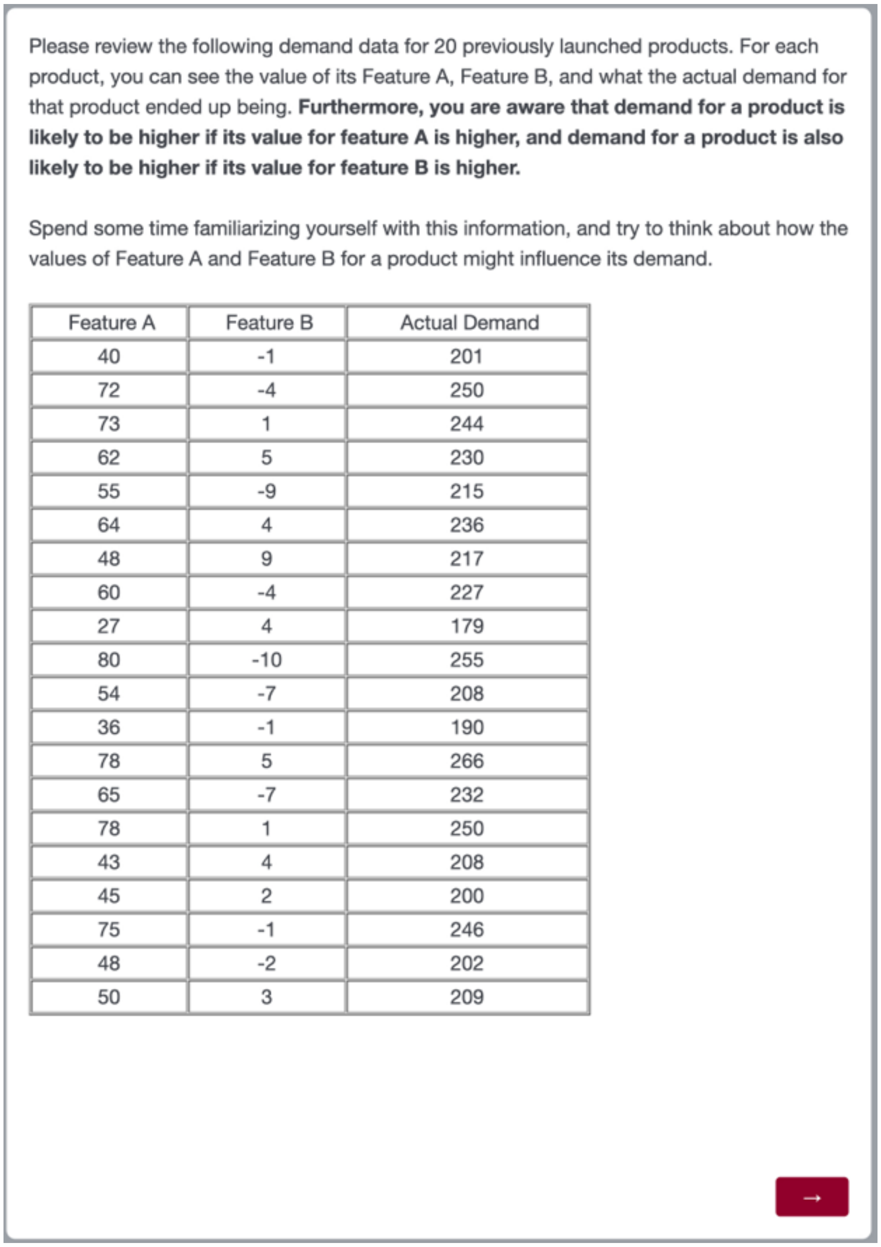
Product 2:

Product Feature	Value
A	23
B	-7

Product 1

Product 2

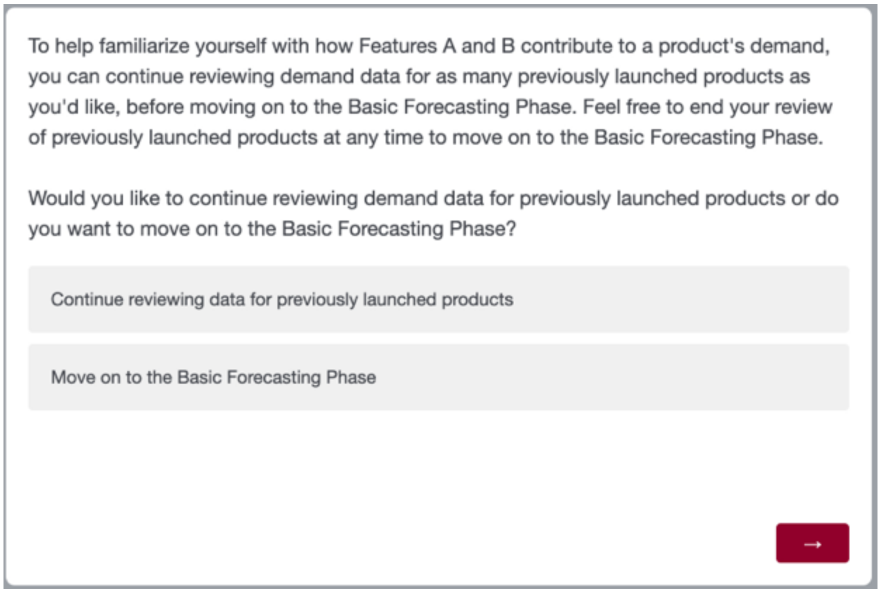
E.2. Step 2: Historical Data Review



Please review the following demand data for 20 previously launched products. For each product, you can see the value of its Feature A, Feature B, and what the actual demand for that product ended up being. **Furthermore, you are aware that demand for a product is likely to be higher if its value for feature A is higher, and demand for a product is also likely to be higher if its value for feature B is higher.**

Spend some time familiarizing yourself with this information, and try to think about how the values of Feature A and Feature B for a product might influence its demand.

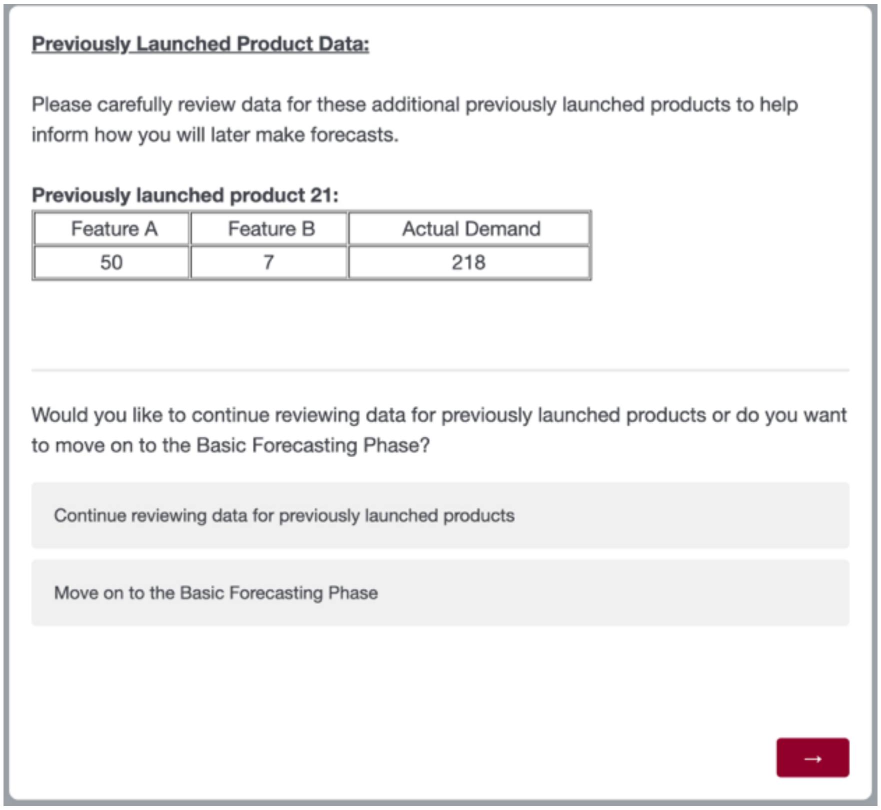
Feature A	Feature B	Actual Demand
40	-1	201
72	-4	250
73	1	244
62	5	230
55	-9	215
64	4	236
48	9	217
60	-4	227
27	4	179
80	-10	255
54	-7	208
36	-1	190
78	5	266
65	-7	232
78	1	250
43	4	208
45	2	200
75	-1	246
48	-2	202
50	3	209



To help familiarize yourself with how Features A and B contribute to a product's demand, you can continue reviewing demand data for as many previously launched products as you'd like, before moving on to the Basic Forecasting Phase. Feel free to end your review of previously launched products at any time to move on to the Basic Forecasting Phase.

Would you like to continue reviewing demand data for previously launched products or do you want to move on to the Basic Forecasting Phase?

Continue reviewing data for previously launched products
Move on to the Basic Forecasting Phase
→



Previously Launched Product Data:

Please carefully review data for these additional previously launched products to help inform how you will later make forecasts.

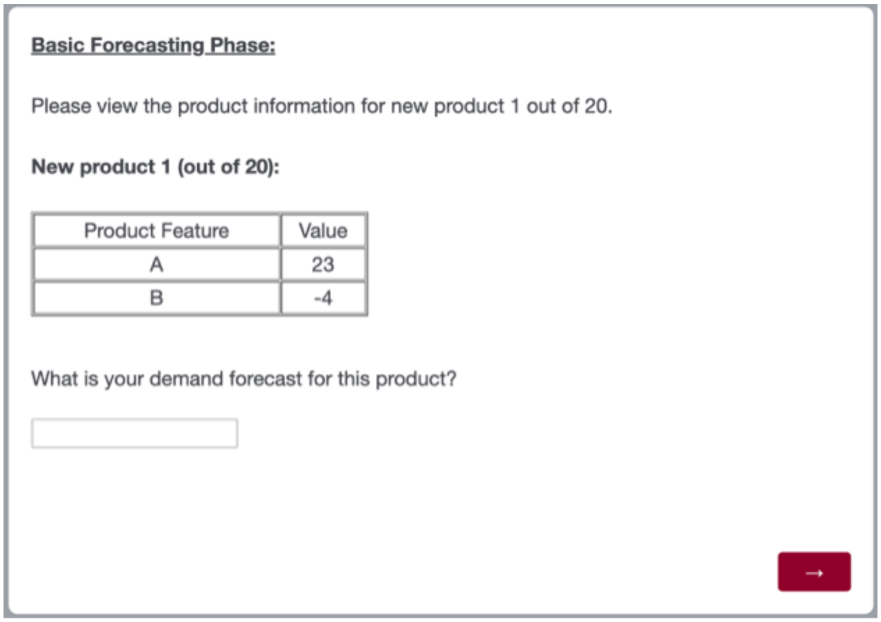
Previously launched product 21:

Feature A	Feature B	Actual Demand
50	7	218

Would you like to continue reviewing data for previously launched products or do you want to move on to the Basic Forecasting Phase?

Continue reviewing data for previously launched products
Move on to the Basic Forecasting Phase
→

E.3. Step 3: Demand Predictions without Algorithm



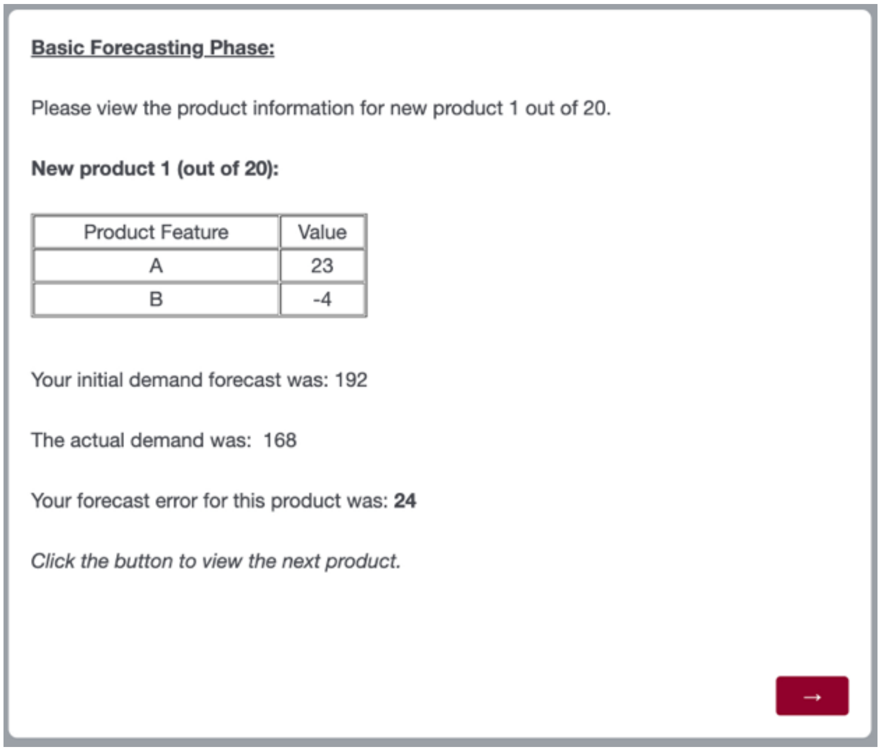
Basic Forecasting Phase:

Please view the product information for new product 1 out of 20.

New product 1 (out of 20):

Product Feature	Value
A	23
B	-4

What is your demand forecast for this product?



Basic Forecasting Phase:

Please view the product information for new product 1 out of 20.

New product 1 (out of 20):

Product Feature	Value
A	23
B	-4

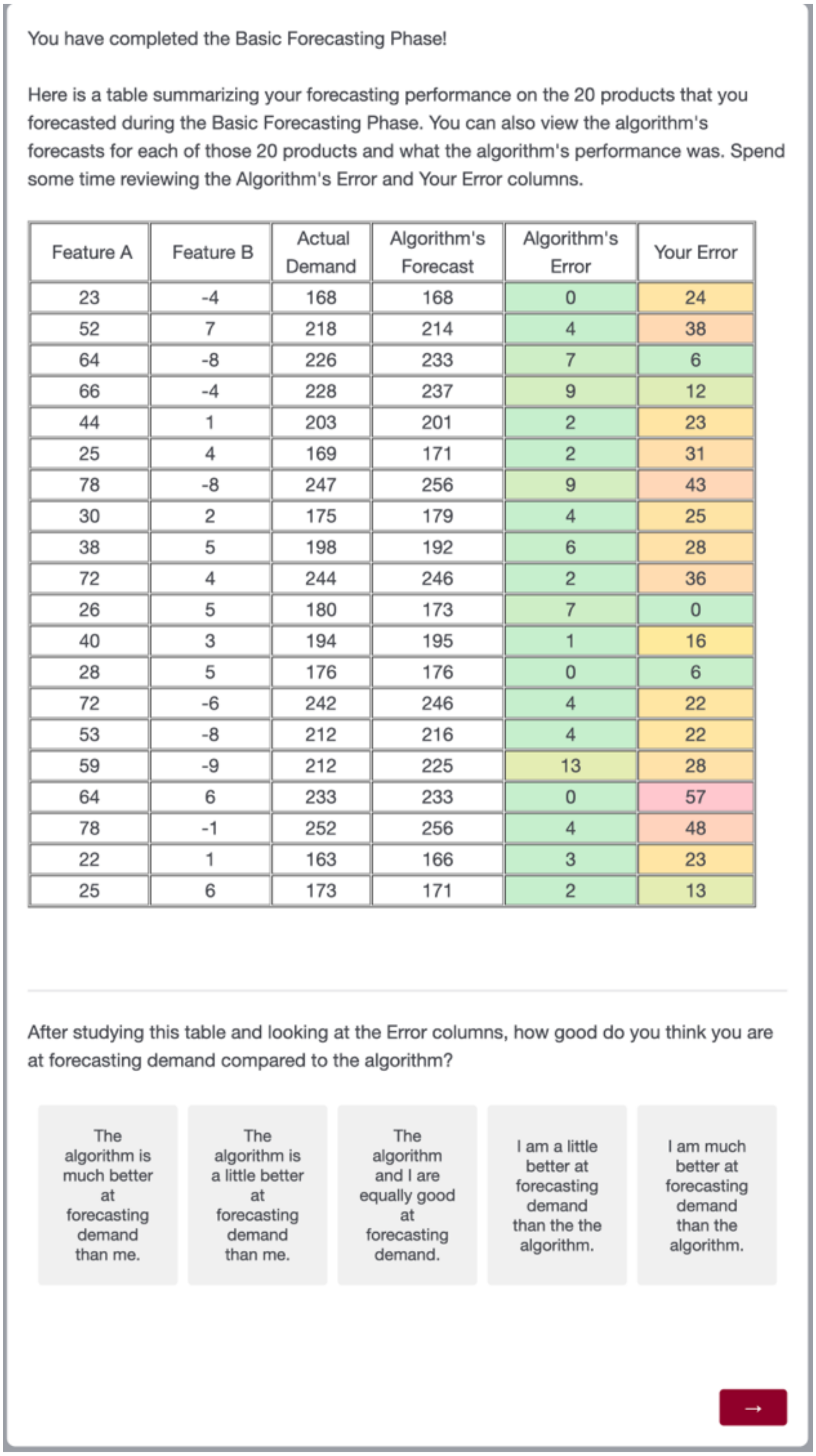
Your initial demand forecast was: 192

The actual demand was: 168

Your forecast error for this product was: **24**

Click the button to view the next product.

E.4. Step 4: Algorithm Introduction



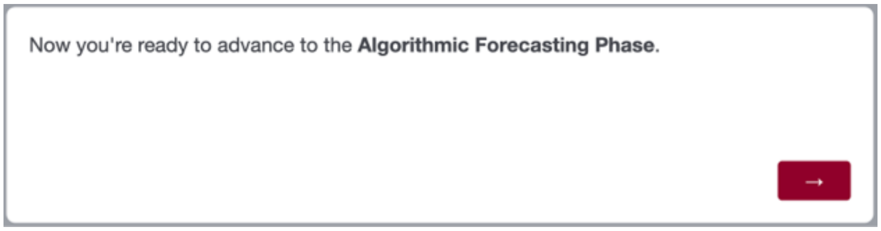
You have completed the Basic Forecasting Phase!

Here is a table summarizing your forecasting performance on the 20 products that you forecasted during the Basic Forecasting Phase. You can also view the algorithm's forecasts for each of those 20 products and what the algorithm's performance was. Spend some time reviewing the Algorithm's Error and Your Error columns.

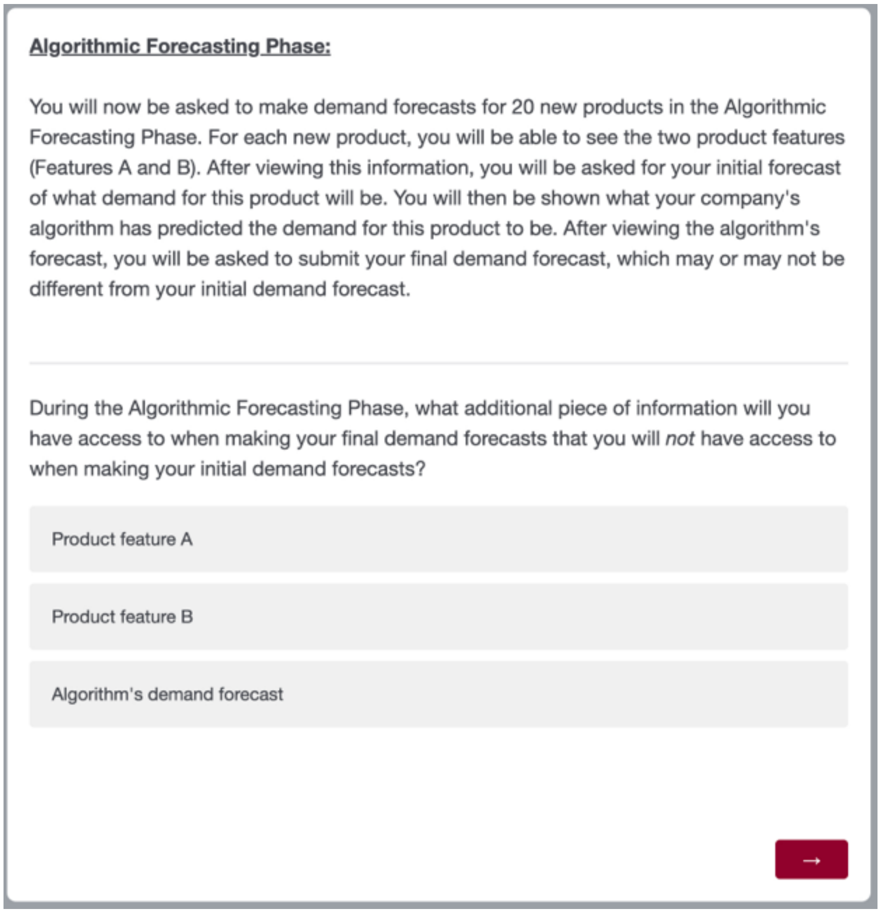
Feature A	Feature B	Actual Demand	Algorithm's Forecast	Algorithm's Error	Your Error
23	-4	168	168	0	24
52	7	218	214	4	38
64	-8	226	233	7	6
66	-4	228	237	9	12
44	1	203	201	2	23
25	4	169	171	2	31
78	-8	247	256	9	43
30	2	175	179	4	25
38	5	198	192	6	28
72	4	244	246	2	36
26	5	180	173	7	0
40	3	194	195	1	16
28	5	176	176	0	6
72	-6	242	246	4	22
53	-8	212	216	4	22
59	-9	212	225	13	28
64	6	233	233	0	57
78	-1	252	256	4	48
22	1	163	166	3	23
25	6	173	171	2	13

After studying this table and looking at the Error columns, how good do you think you are at forecasting demand compared to the algorithm?

The algorithm is much better at forecasting demand than me.
The algorithm is a little better at forecasting demand than me.
The algorithm and I are equally good at forecasting demand.
I am a little better at forecasting demand than the the algorithm.
I am much better at forecasting demand than the algorithm.
→



Now you're ready to advance to the **Algorithmic Forecasting Phase**.



Algorithmic Forecasting Phase:

You will now be asked to make demand forecasts for 20 new products in the Algorithmic Forecasting Phase. For each new product, you will be able to see the two product features (Features A and B). After viewing this information, you will be asked for your initial forecast of what demand for this product will be. You will then be shown what your company's algorithm has predicted the demand for this product to be. After viewing the algorithm's forecast, you will be asked to submit your final demand forecast, which may or may not be different from your initial demand forecast.

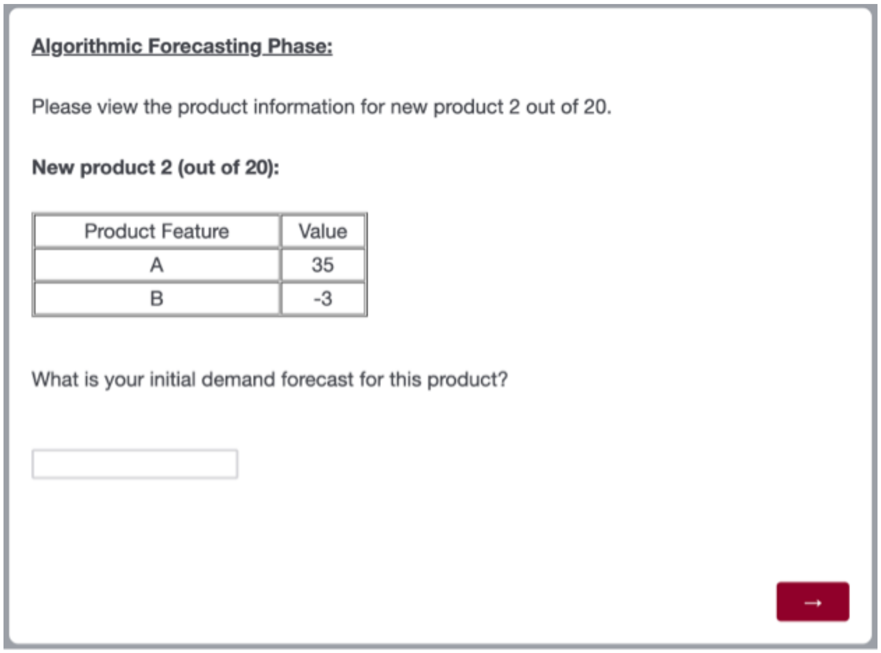
During the Algorithmic Forecasting Phase, what additional piece of information will you have access to when making your final demand forecasts that you will *not* have access to when making your initial demand forecasts?

Product feature A

Product feature B

Algorithm's demand forecast

E.5. Step 5: Demand Predictions with Algorithm



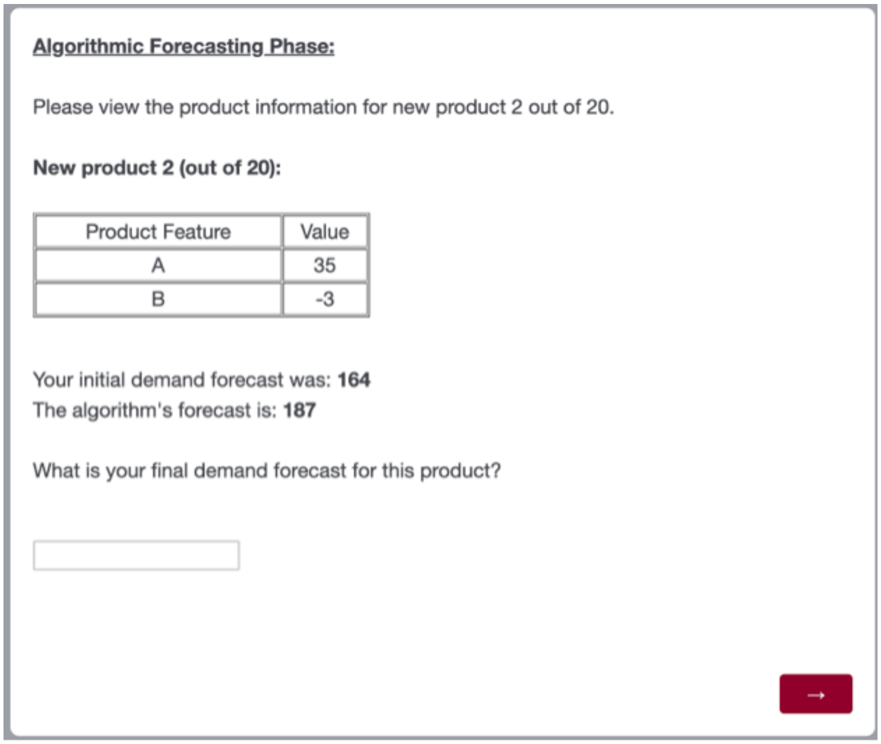
Algorithmic Forecasting Phase:

Please view the product information for new product 2 out of 20.

New product 2 (out of 20):

Product Feature	Value
A	35
B	-3

What is your initial demand forecast for this product?



Algorithmic Forecasting Phase:

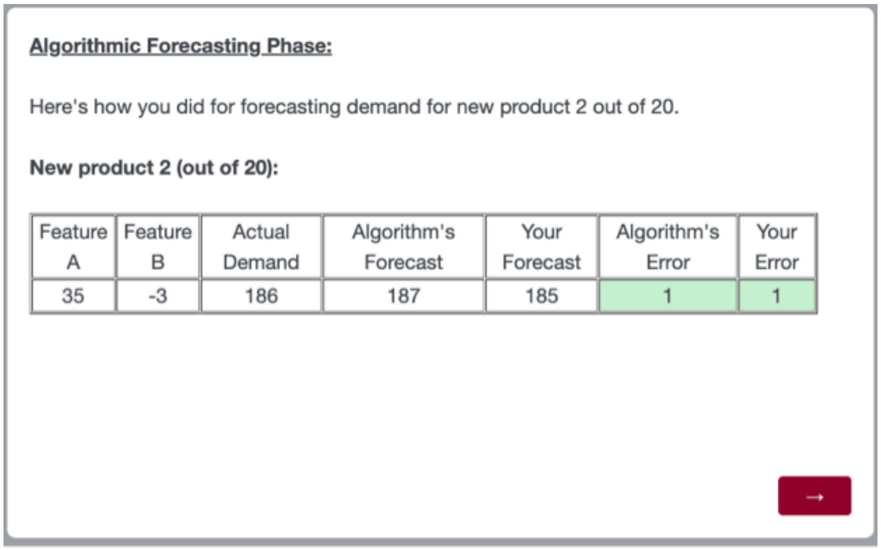
Please view the product information for new product 2 out of 20.

New product 2 (out of 20):

Product Feature	Value
A	35
B	-3

Your initial demand forecast was: **164**
The algorithm's forecast is: **187**

What is your final demand forecast for this product?



Algorithmic Forecasting Phase:

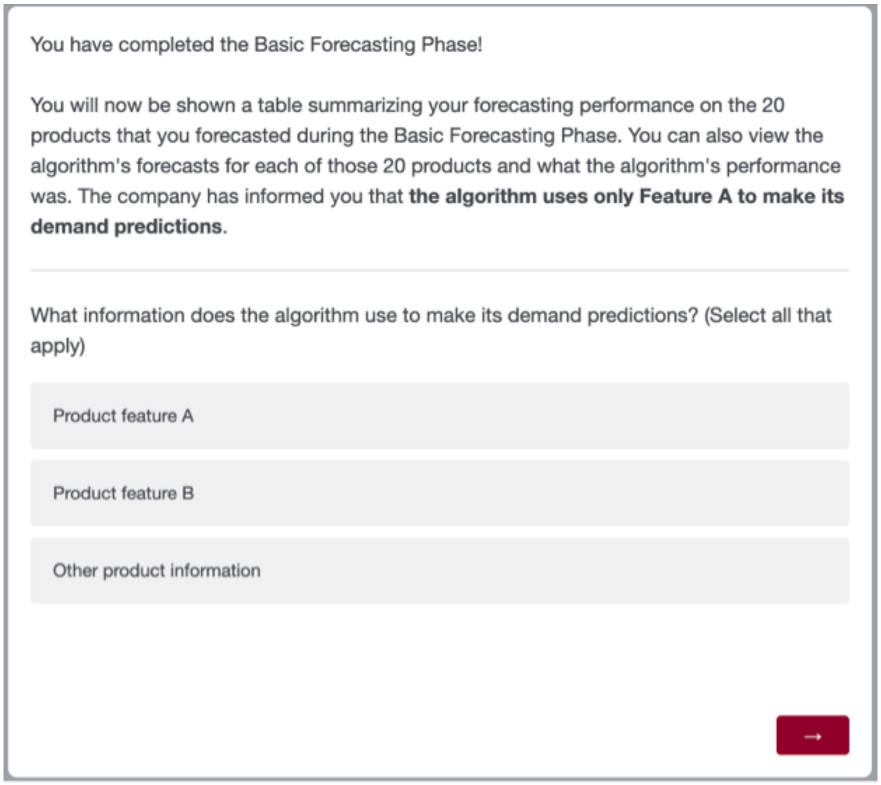
Here's how you did for forecasting demand for new product 2 out of 20.

New product 2 (out of 20):

Feature A	Feature B	Actual Demand	Algorithm's Forecast	Your Forecast	Algorithm's Error	Your Error
35	-3	186	187	185	1	1

Appendix F: Experiment 2: Participant Experience Changes

F.1. Feature Transparency

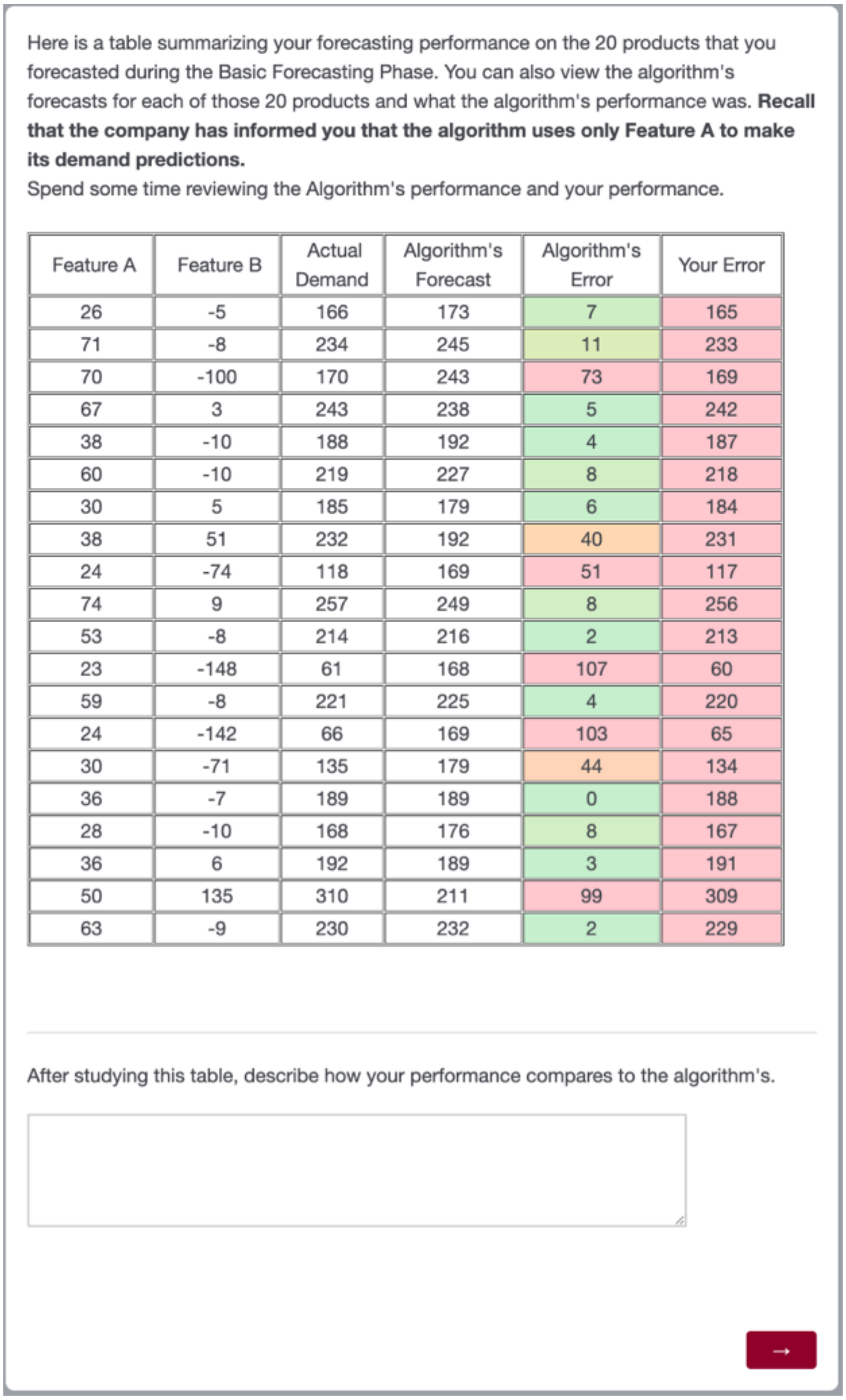


You have completed the Basic Forecasting Phase!

You will now be shown a table summarizing your forecasting performance on the 20 products that you forecasted during the Basic Forecasting Phase. You can also view the algorithm's forecasts for each of those 20 products and what the algorithm's performance was. The company has informed you that **the algorithm uses only Feature A to make its demand predictions**.

What information does the algorithm use to make its demand predictions? (Select all that apply)

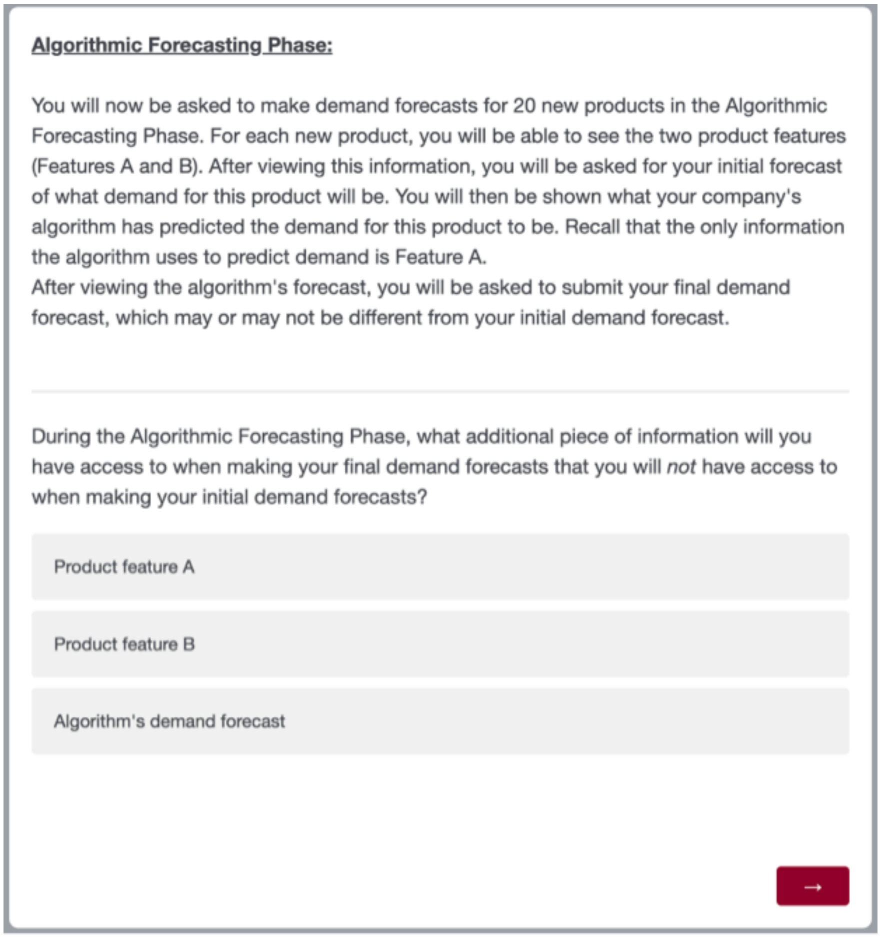
- Product feature A
- Product feature B
- Other product information



Here is a table summarizing your forecasting performance on the 20 products that you forecasted during the Basic Forecasting Phase. You can also view the algorithm's forecasts for each of those 20 products and what the algorithm's performance was. **Recall that the company has informed you that the algorithm uses only Feature A to make its demand predictions.**
Spend some time reviewing the Algorithm's performance and your performance.

Feature A	Feature B	Actual Demand	Algorithm's Forecast	Algorithm's Error	Your Error
26	-5	166	173	7	165
71	-8	234	245	11	233
70	-100	170	243	73	169
67	3	243	238	5	242
38	-10	188	192	4	187
60	-10	219	227	8	218
30	5	185	179	6	184
38	51	232	192	40	231
24	-74	118	169	51	117
74	9	257	249	8	256
53	-8	214	216	2	213
23	-148	61	168	107	60
59	-8	221	225	4	220
24	-142	66	169	103	65
30	-71	135	179	44	134
36	-7	189	189	0	188
28	-10	168	176	8	167
36	6	192	189	3	191
50	135	310	211	99	309
63	-9	230	232	2	229

After studying this table, describe how your performance compares to the algorithm's.

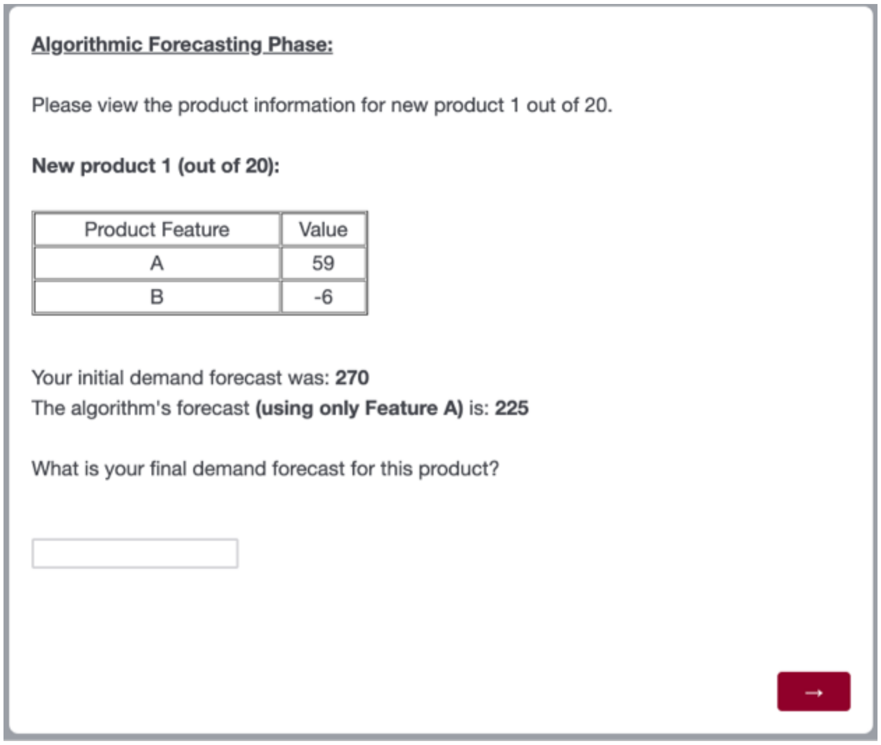


Algorithmic Forecasting Phase:

You will now be asked to make demand forecasts for 20 new products in the Algorithmic Forecasting Phase. For each new product, you will be able to see the two product features (Features A and B). After viewing this information, you will be asked for your initial forecast of what demand for this product will be. You will then be shown what your company's algorithm has predicted the demand for this product to be. Recall that the only information the algorithm uses to predict demand is Feature A.
After viewing the algorithm's forecast, you will be asked to submit your final demand forecast, which may or may not be different from your initial demand forecast.

During the Algorithmic Forecasting Phase, what additional piece of information will you have access to when making your final demand forecasts that you will *not* have access to when making your initial demand forecasts?

- Product feature A
- Product feature B
- Algorithm's demand forecast



Algorithmic Forecasting Phase:

Please view the product information for new product 1 out of 20.

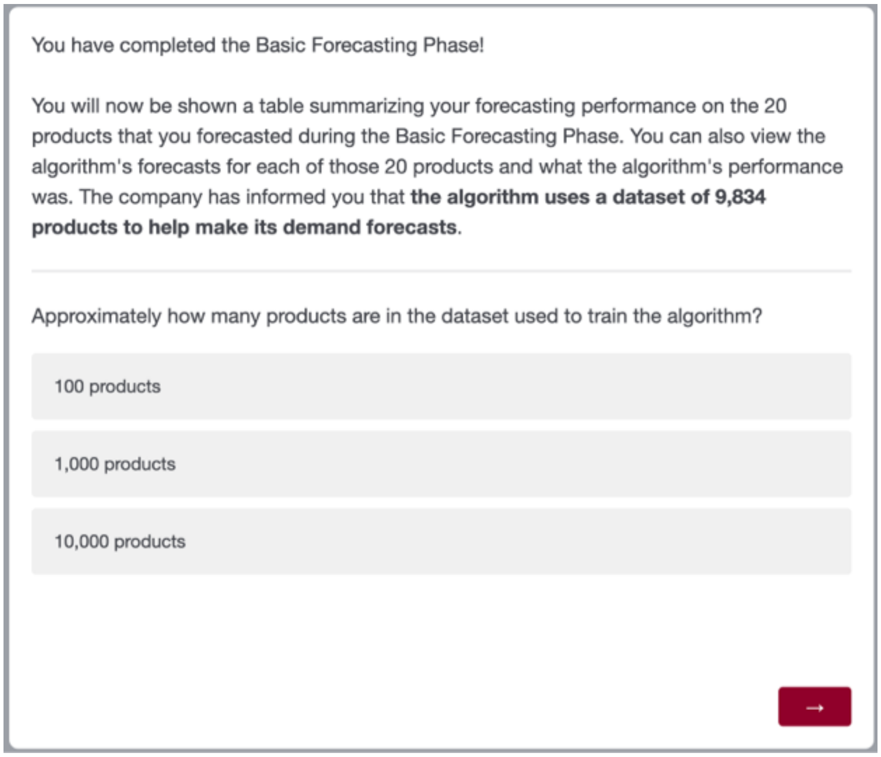
New product 1 (out of 20):

Product Feature	Value
A	59
B	-6

Your initial demand forecast was: **270**
The algorithm's forecast **(using only Feature A)** is: **225**

What is your final demand forecast for this product?

F.2. Training Data Transparency



You have completed the Basic Forecasting Phase!

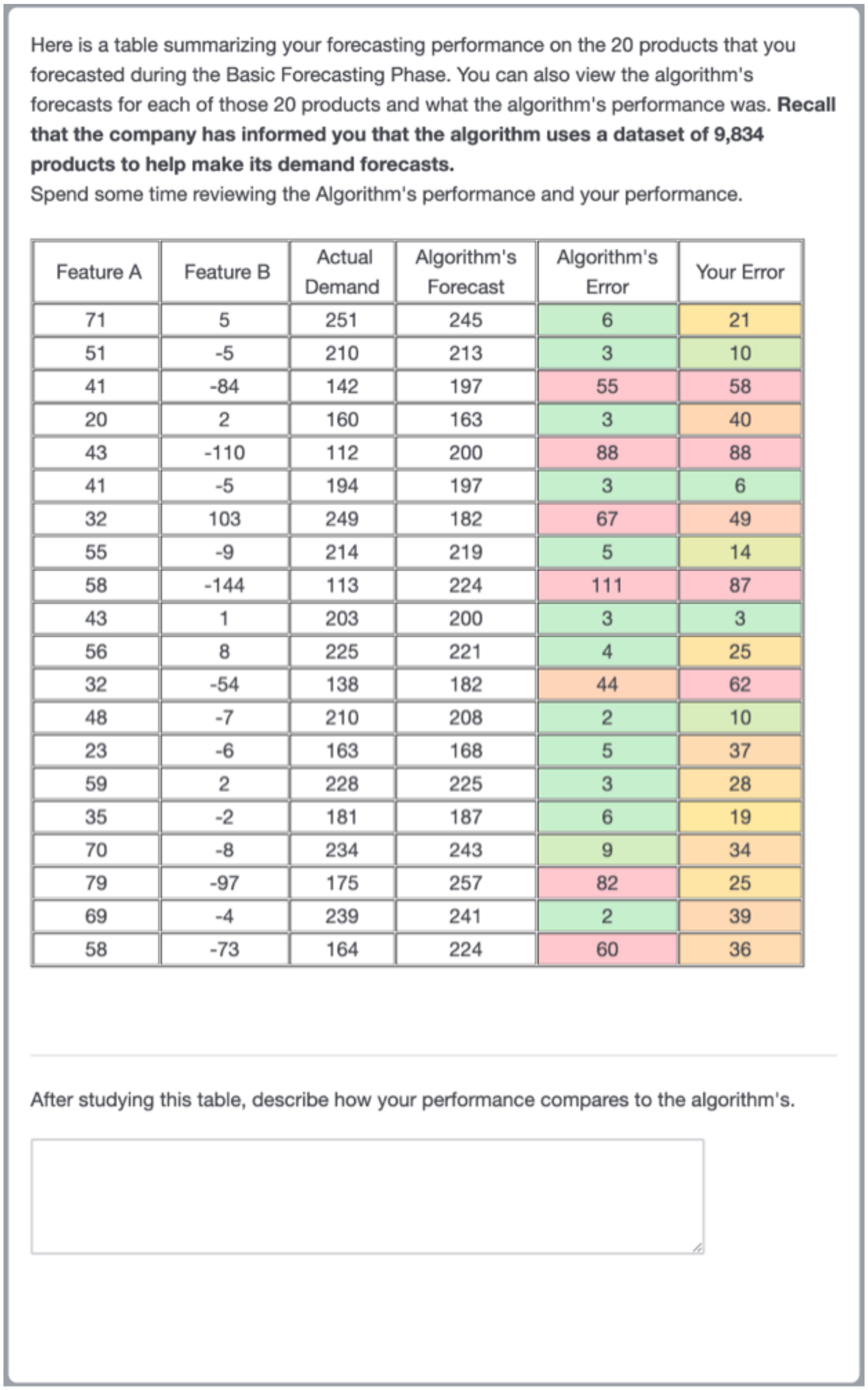
You will now be shown a table summarizing your forecasting performance on the 20 products that you forecasted during the Basic Forecasting Phase. You can also view the algorithm's forecasts for each of those 20 products and what the algorithm's performance was. The company has informed you that **the algorithm uses a dataset of 9,834 products to help make its demand forecasts**.

Approximately how many products are in the dataset used to train the algorithm?

100 products

1,000 products

10,000 products

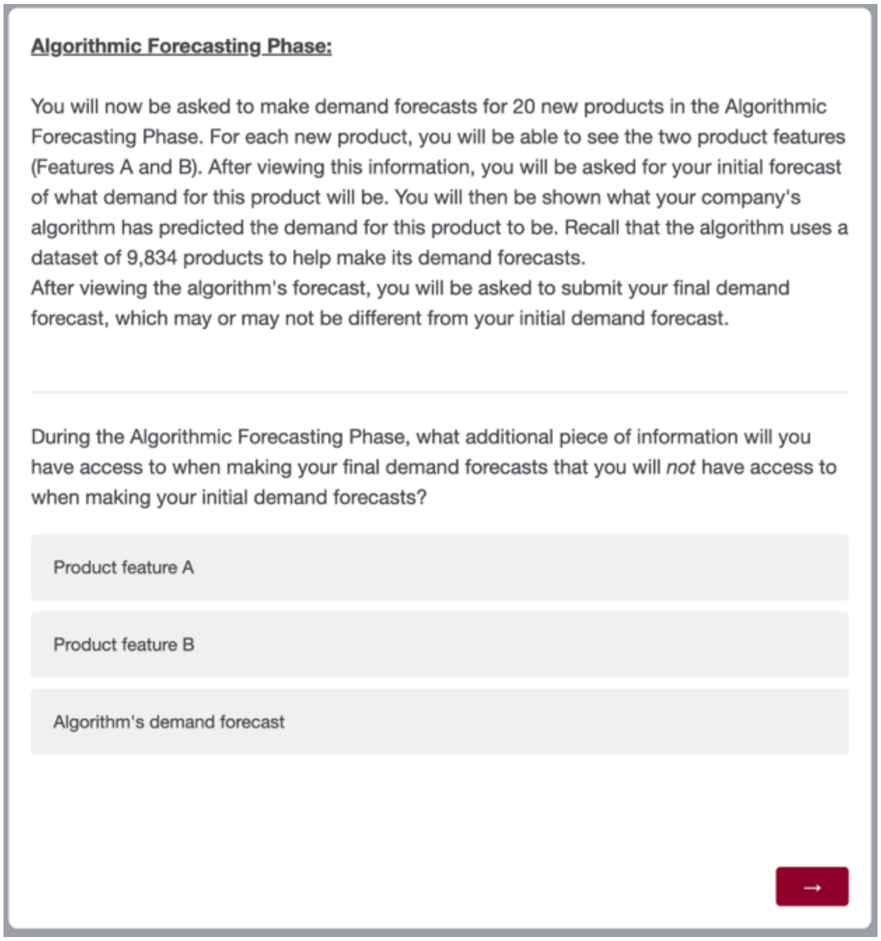


Here is a table summarizing your forecasting performance on the 20 products that you forecasted during the Basic Forecasting Phase. You can also view the algorithm's forecasts for each of those 20 products and what the algorithm's performance was. **Recall that the company has informed you that the algorithm uses a dataset of 9,834 products to help make its demand forecasts.**

Spend some time reviewing the Algorithm's performance and your performance.

Feature A	Feature B	Actual Demand	Algorithm's Forecast	Algorithm's Error	Your Error
71	5	251	245	6	21
51	-5	210	213	3	10
41	-84	142	197	55	58
20	2	160	163	3	40
43	-110	112	200	88	88
41	-5	194	197	3	6
32	103	249	182	67	49
55	-9	214	219	5	14
58	-144	113	224	111	87
43	1	203	200	3	3
56	8	225	221	4	25
32	-54	138	182	44	62
48	-7	210	208	2	10
23	-6	163	168	5	37
59	2	228	225	3	28
35	-2	181	187	6	19
70	-8	234	243	9	34
79	-97	175	257	82	25
69	-4	239	241	2	39
58	-73	164	224	60	36

After studying this table, describe how your performance compares to the algorithm's.



Algorithmic Forecasting Phase:

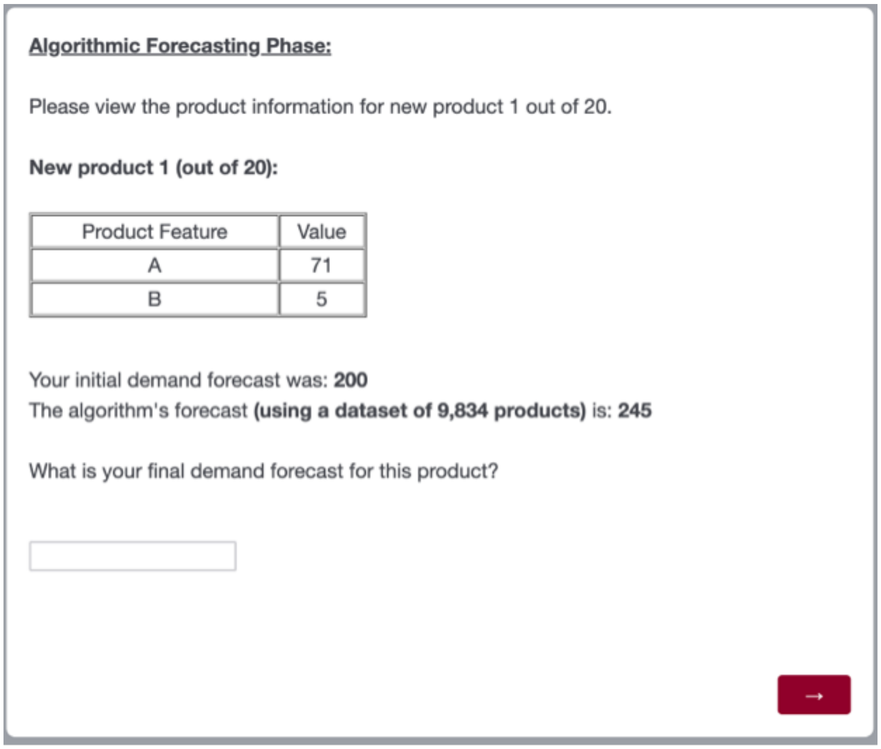
You will now be asked to make demand forecasts for 20 new products in the Algorithmic Forecasting Phase. For each new product, you will be able to see the two product features (Features A and B). After viewing this information, you will be asked for your initial forecast of what demand for this product will be. You will then be shown what your company's algorithm has predicted the demand for this product to be. Recall that the algorithm uses a dataset of 9,834 products to help make its demand forecasts.
After viewing the algorithm's forecast, you will be asked to submit your final demand forecast, which may or may not be different from your initial demand forecast.

During the Algorithmic Forecasting Phase, what additional piece of information will you have access to when making your final demand forecasts that you will *not* have access to when making your initial demand forecasts?

Product feature A

Product feature B

Algorithm's demand forecast



Algorithmic Forecasting Phase:

Please view the product information for new product 1 out of 20.

New product 1 (out of 20):

Product Feature	Value
A	71
B	5

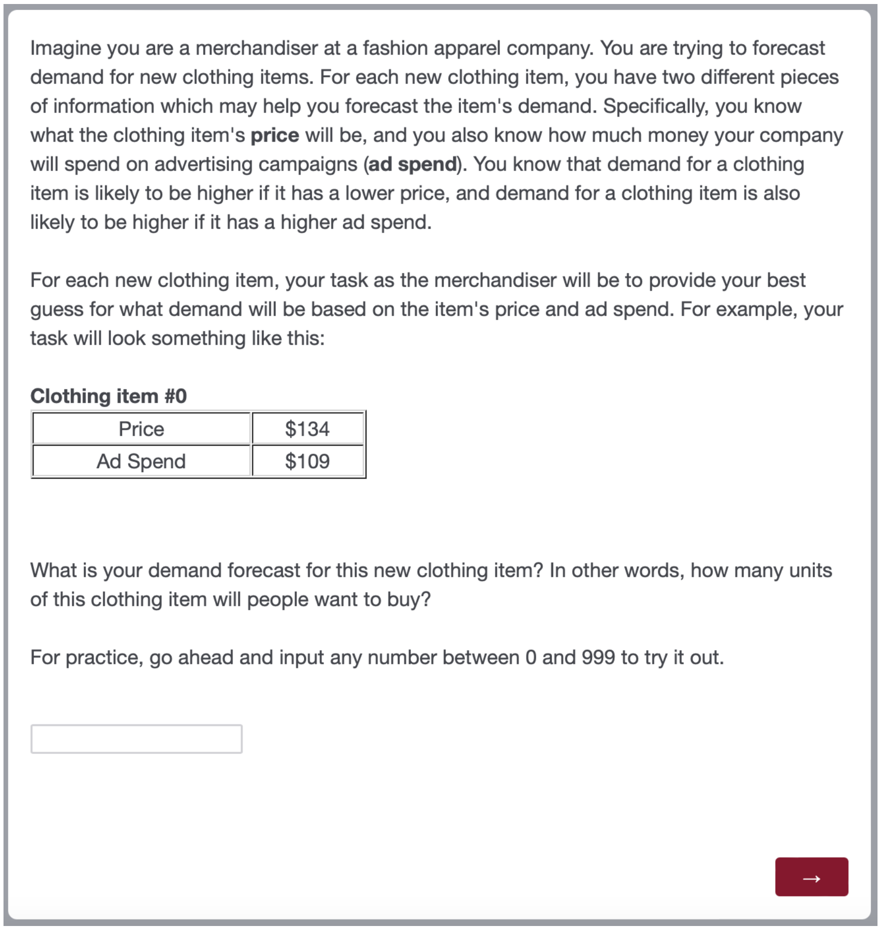
Your initial demand forecast was: **200**
The algorithm's forecast **(using a dataset of 9,834 products)** is: **245**

What is your final demand forecast for this product?

Appendix G: Experiment 3: Participant Experience

G.1. Step 1: Instructions and Comprehension Checks

The following screenshots from Step 1 are identical across all three treatment conditions.



Imagine you are a merchandiser at a fashion apparel company. You are trying to forecast demand for new clothing items. For each new clothing item, you have two different pieces of information which may help you forecast the item's demand. Specifically, you know what the clothing item's **price** will be, and you also know how much money your company will spend on advertising campaigns (**ad spend**). You know that demand for a clothing item is likely to be higher if it has a lower price, and demand for a clothing item is also likely to be higher if it has a higher ad spend.

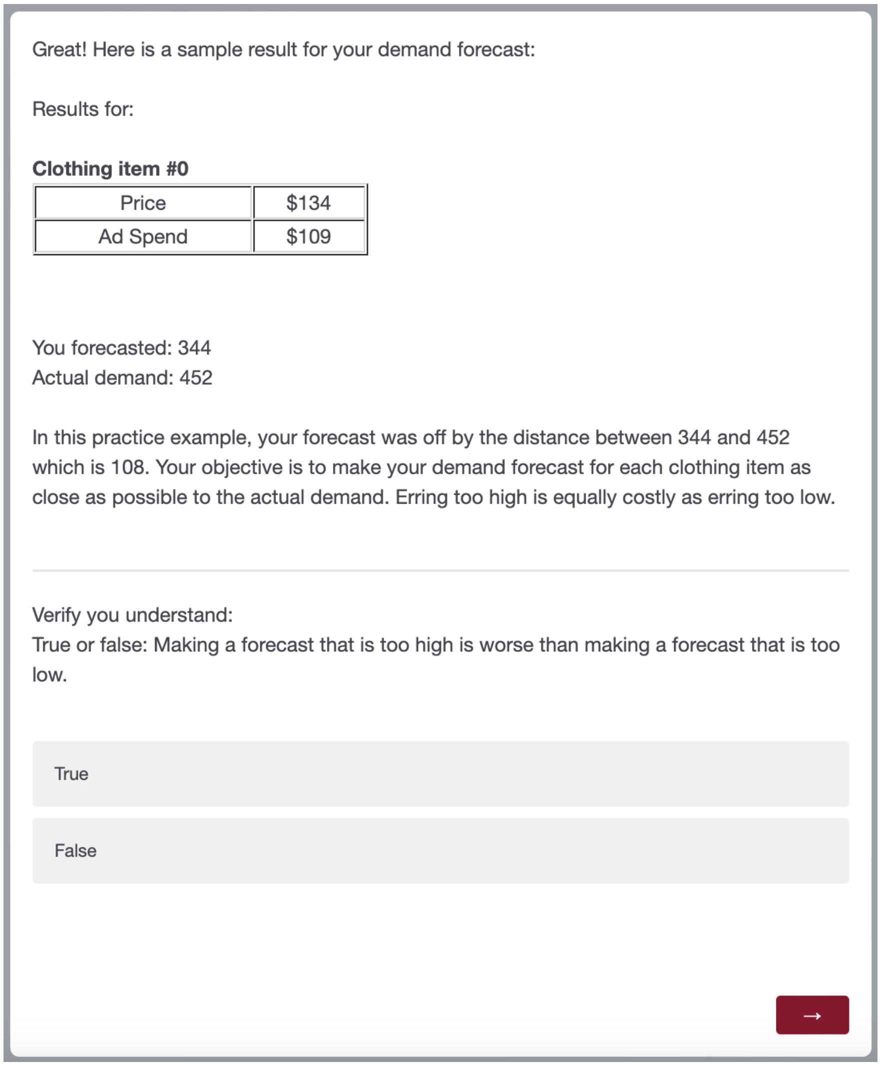
For each new clothing item, your task as the merchandiser will be to provide your best guess for what demand will be based on the item's price and ad spend. For example, your task will look something like this:

Clothing item #0

Price	$134
Ad Spend	$109

What is your demand forecast for this new clothing item? In other words, how many units of this clothing item will people want to buy?

For practice, go ahead and input any number between 0 and 999 to try it out.



Great! Here is a sample result for your demand forecast:

Results for:

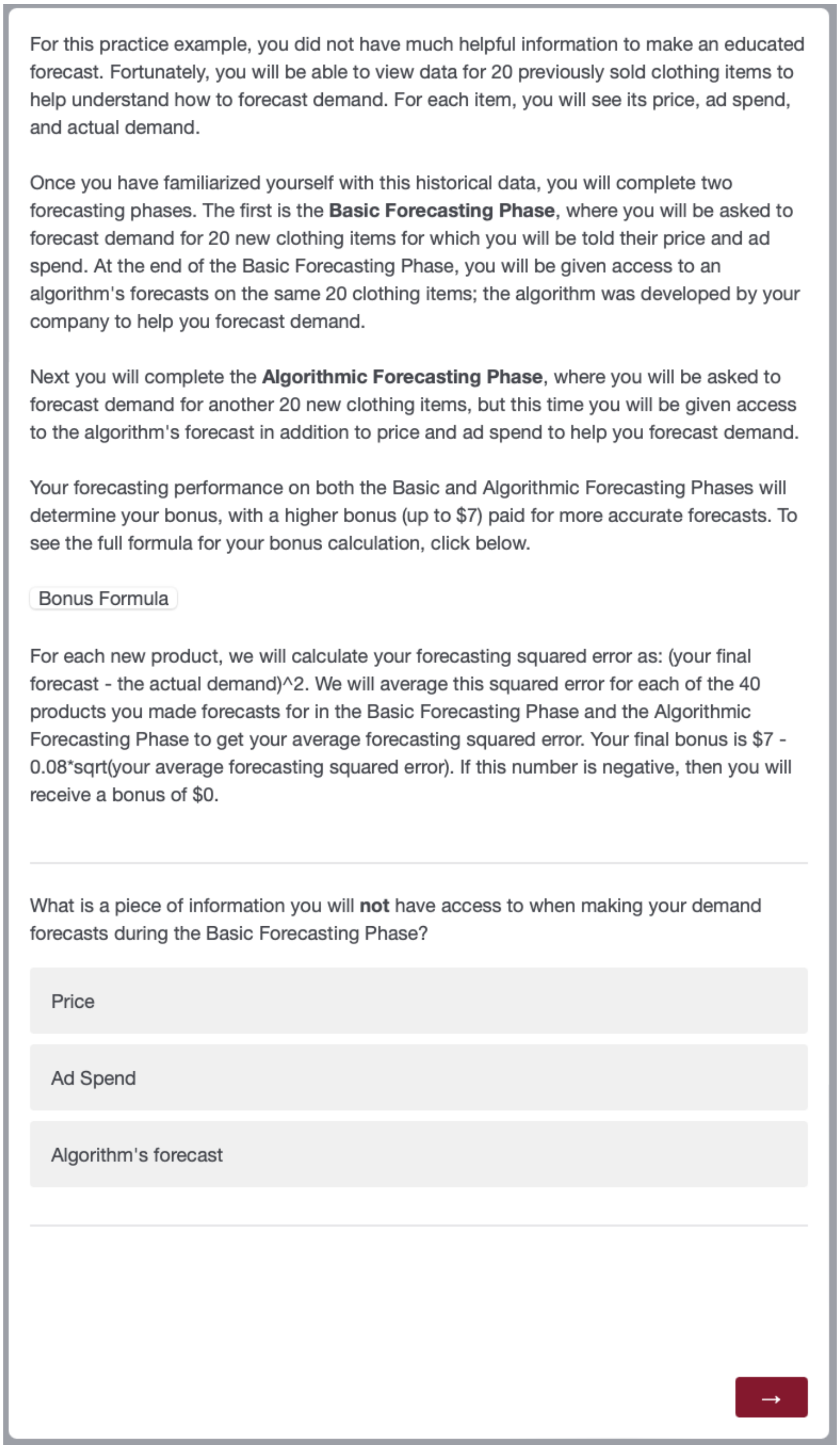
Clothing item #0

Price	$134
Ad Spend	$109

You forecasted: 344
Actual demand: 452

In this practice example, your forecast was off by the distance between 344 and 452 which is 108. Your objective is to make your demand forecast for each clothing item as close as possible to the actual demand. Erring too high is equally costly as erring too low.

Verify you understand:
True or false: Making a forecast that is too high is worse than making a forecast that is too low.



For this practice example, you did not have much helpful information to make an educated forecast. Fortunately, you will be able to view data for 20 previously sold clothing items to help understand how to forecast demand. For each item, you will see its price, ad spend, and actual demand.

Once you have familiarized yourself with this historical data, you will complete two forecasting phases. The first is the **Basic Forecasting Phase**, where you will be asked to forecast demand for 20 new clothing items for which you will be told their price and ad spend. At the end of the Basic Forecasting Phase, you will be given access to an algorithm's forecasts on the same 20 clothing items; the algorithm was developed by your company to help you forecast demand.

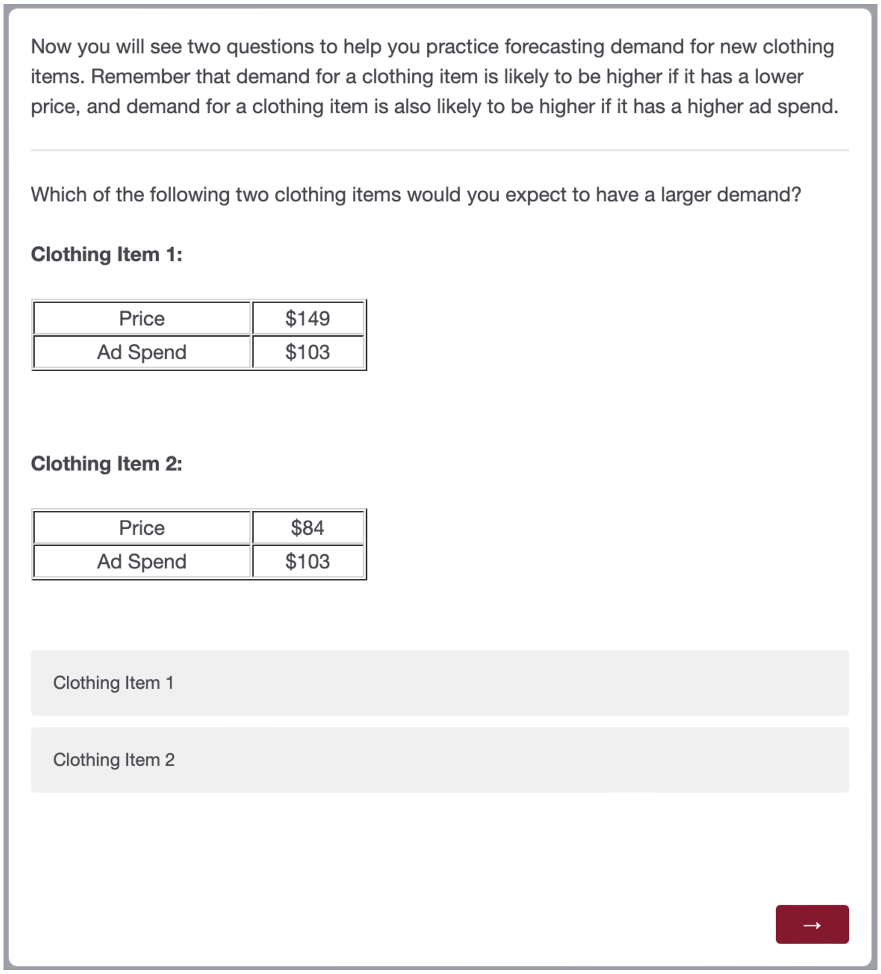
Next you will complete the **Algorithmic Forecasting Phase**, where you will be asked to forecast demand for another 20 new clothing items, but this time you will be given access to the algorithm's forecast in addition to price and ad spend to help you forecast demand.

Your forecasting performance on both the Basic and Algorithmic Forecasting Phases will determine your bonus, with a higher bonus (up to $7) paid for more accurate forecasts. To see the full formula for your bonus calculation, click below.

For each new product, we will calculate your forecasting squared error as: (your final forecast - the actual demand)^2. We will average this squared error for each of the 40 products you made forecasts for in the Basic Forecasting Phase and the Algorithmic Forecasting Phase to get your average forecasting squared error. Your final bonus is $7 - 0.08*sqrt(your average forecasting squared error). If this number is negative, then you will receive a bonus of $0.

What is a piece of information you will **not** have access to when making your demand forecasts during the Basic Forecasting Phase?

- Price
- Ad Spend
- Algorithm's forecast



Now you will see two questions to help you practice forecasting demand for new clothing items. Remember that demand for a clothing item is likely to be higher if it has a lower price, and demand for a clothing item is also likely to be higher if it has a higher ad spend.

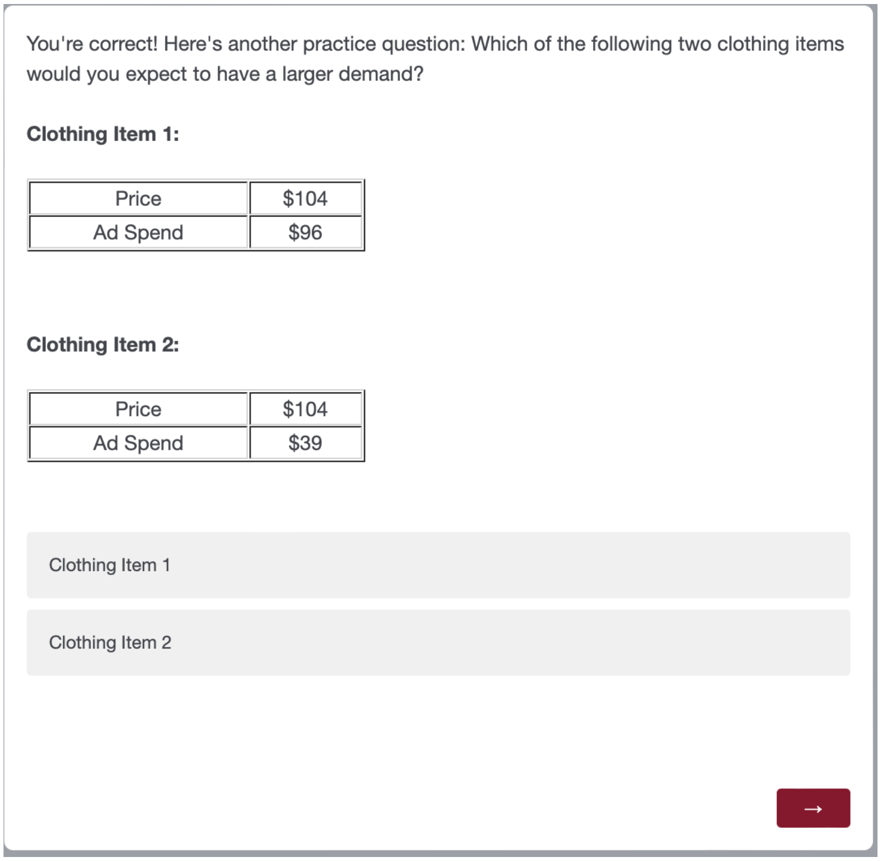
Which of the following two clothing items would you expect to have a larger demand?

Clothing Item 1:

Price	$149
Ad Spend	$103

Clothing Item 2:

Price	$84
Ad Spend	$103



You're correct! Here's another practice question: Which of the following two clothing items would you expect to have a larger demand?

Clothing Item 1:

Price	$104
Ad Spend	$96

Clothing Item 2:

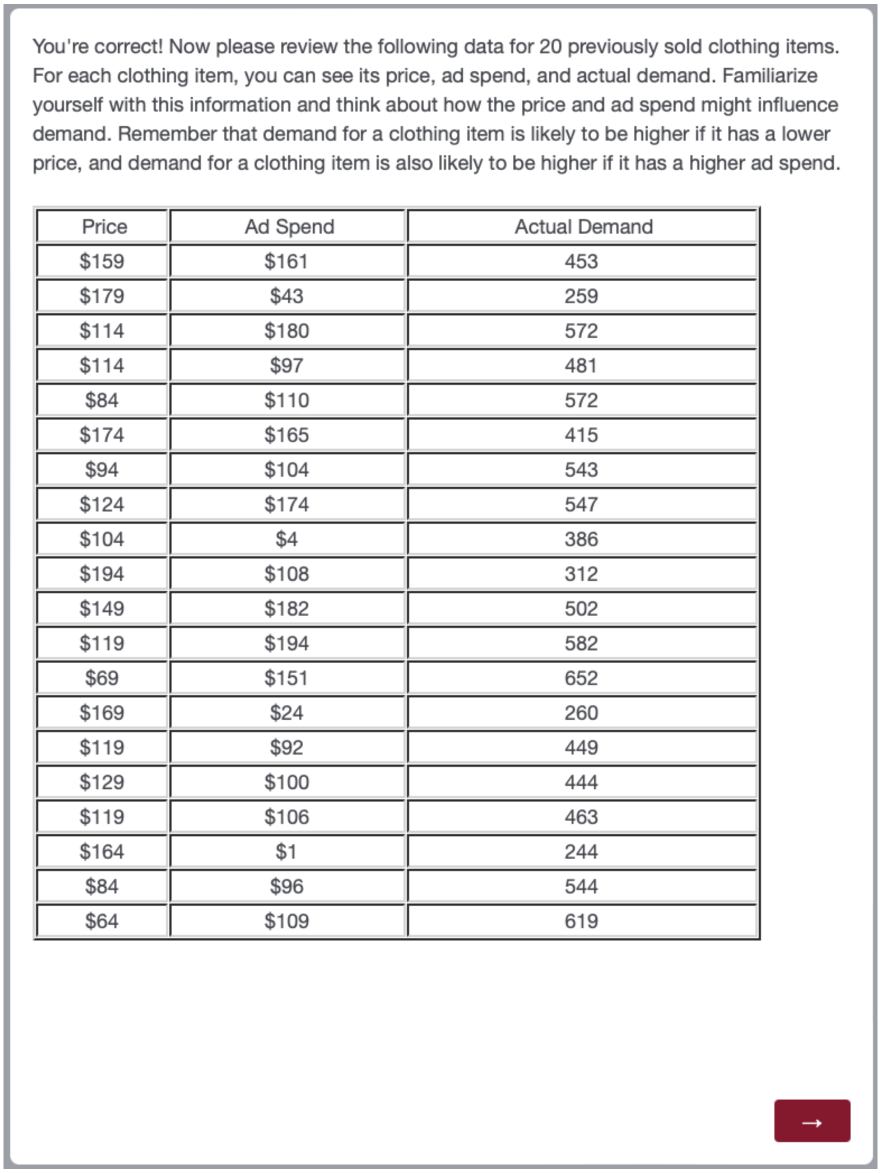
Price	$104
Ad Spend	$39

Clothing Item 1

Clothing Item 2

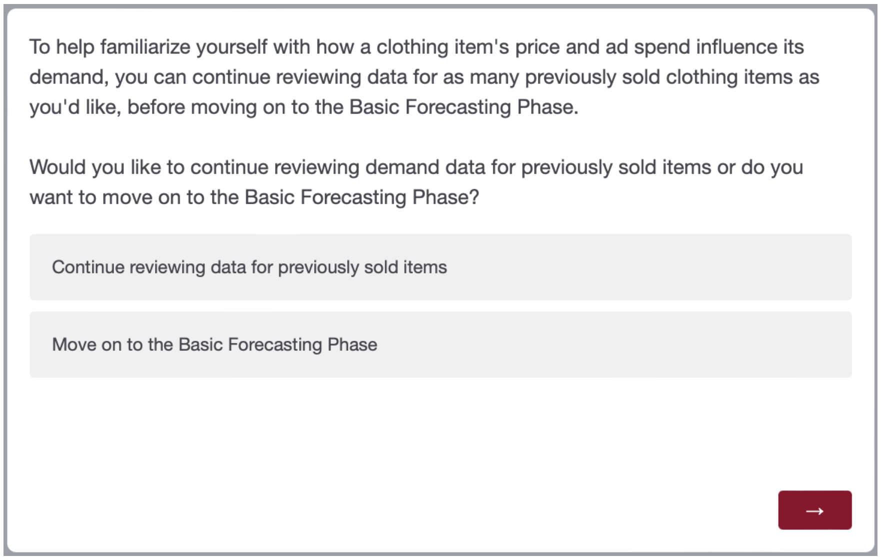
G.2. Step 2: Historical Data Review

The following screenshots from Step 2 are identical across all three treatment conditions.



You're correct! Now please review the following data for 20 previously sold clothing items. For each clothing item, you can see its price, ad spend, and actual demand. Familiarize yourself with this information and think about how the price and ad spend might influence demand. Remember that demand for a clothing item is likely to be higher if it has a lower price, and demand for a clothing item is also likely to be higher if it has a higher ad spend.

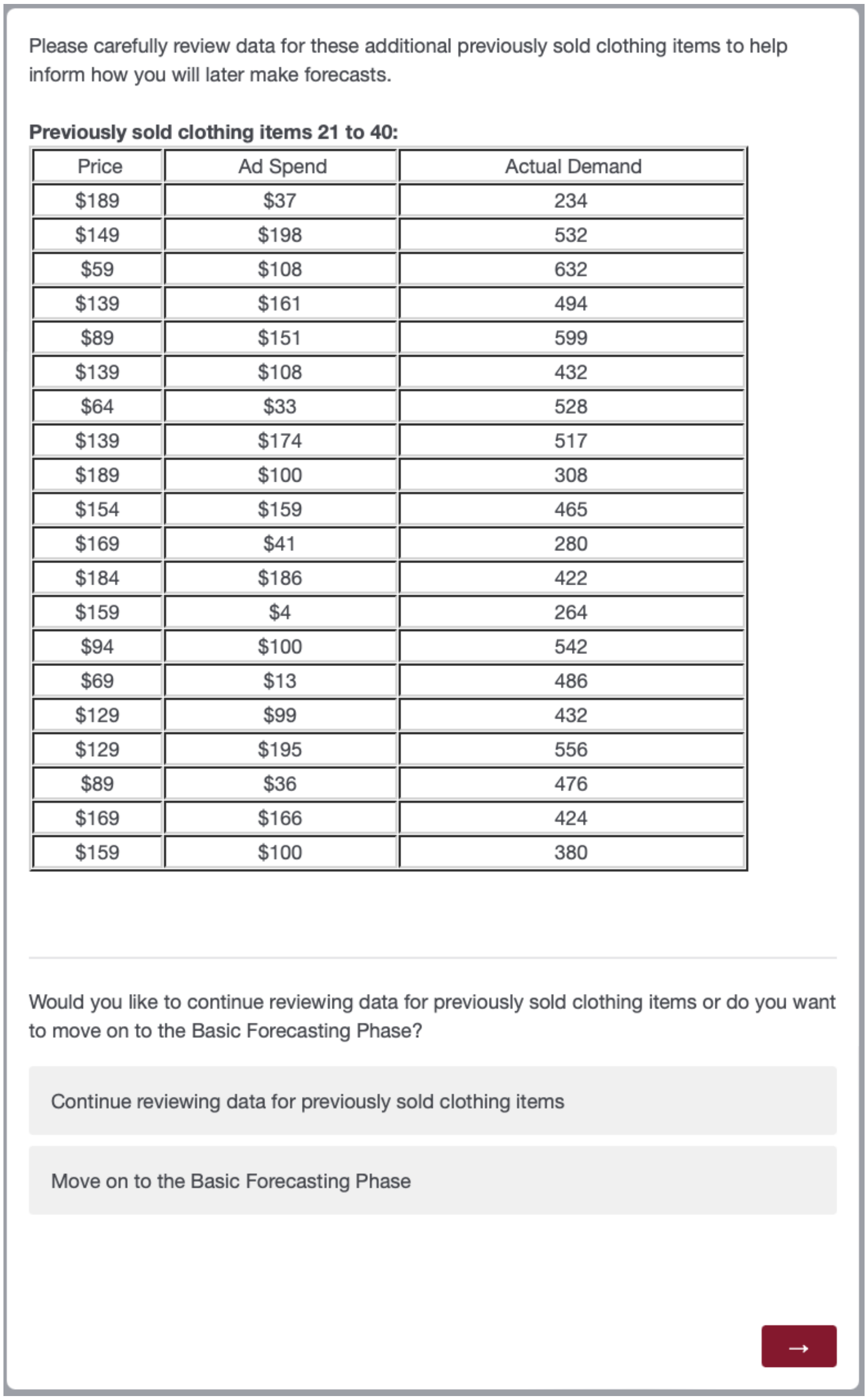
Price	Ad Spend	Actual Demand
$159	$161	453
$179	$43	259
$114	$180	572
$114	$97	481
$84	$110	572
$174	$165	415
$94	$104	543
$124	$174	547
$104	$4	386
$194	$108	312
$149	$182	502
$119	$194	582
$69	$151	652
$169	$24	260
$119	$92	449
$129	$100	444
$119	$106	463
$164	$1	244
$84	$96	544
$64	$109	619



To help familiarize yourself with how a clothing item's price and ad spend influence its demand, you can continue reviewing data for as many previously sold clothing items as you'd like, before moving on to the Basic Forecasting Phase.

Would you like to continue reviewing demand data for previously sold items or do you want to move on to the Basic Forecasting Phase?

Continue reviewing data for previously sold items
Move on to the Basic Forecasting Phase
→



Please carefully review data for these additional previously sold clothing items to help inform how you will later make forecasts.

Previously sold clothing items 21 to 40:

Price	Ad Spend	Actual Demand
$189	$37	234
$149	$198	532
$59	$108	632
$139	$161	494
$89	$151	599
$139	$108	432
$64	$33	528
$139	$174	517
$189	$100	308
$154	$159	465
$169	$41	280
$184	$186	422
$159	$4	264
$94	$100	542
$69	$13	486
$129	$99	432
$129	$195	556
$89	$36	476
$169	$166	424
$159	$100	380

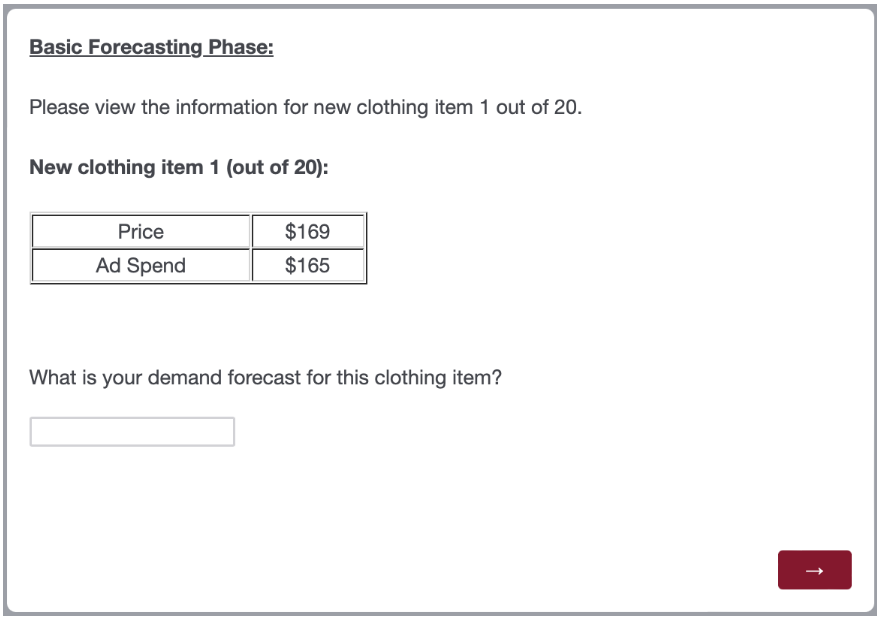
Would you like to continue reviewing data for previously sold clothing items or do you want to move on to the Basic Forecasting Phase?

Continue reviewing data for previously sold clothing items

Move on to the Basic Forecasting Phase

G.3. Step 3: Demand Predictions without Algorithm

The following screenshots from Step 3 are identical across all three treatment conditions.



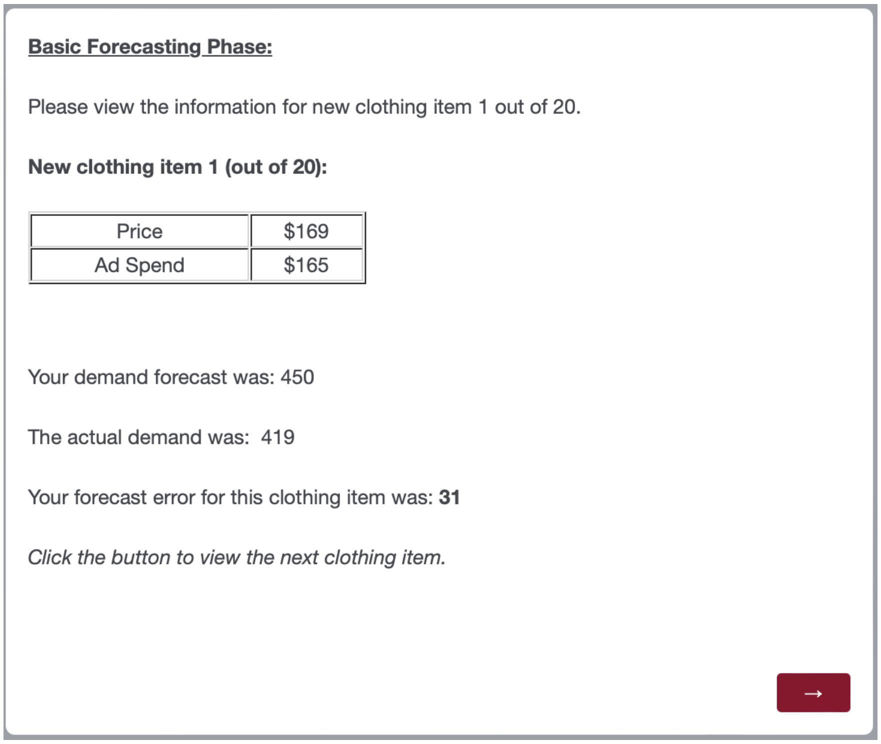
Basic Forecasting Phase:

Please view the information for new clothing item 1 out of 20.

New clothing item 1 (out of 20):

Price	$169
Ad Spend	$165

What is your demand forecast for this clothing item?



Basic Forecasting Phase:

Please view the information for new clothing item 1 out of 20.

New clothing item 1 (out of 20):

Price	$169
Ad Spend	$165

Your demand forecast was: 450

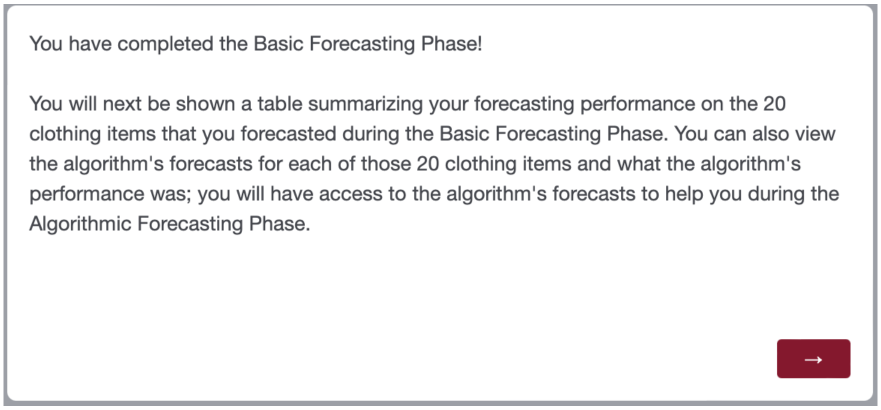
The actual demand was: 419

Your forecast error for this clothing item was: **31**

Click the button to view the next clothing item.

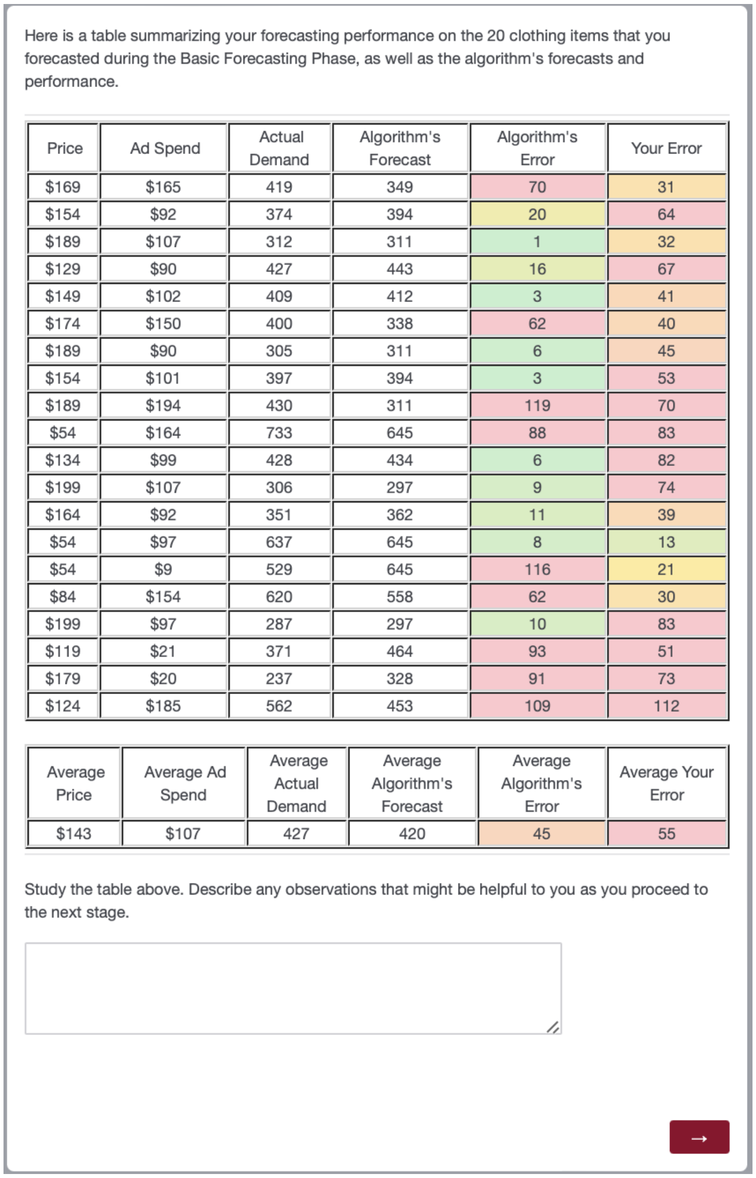
G.4. Step 4: Algorithm Introduction - No Transparency

The following screenshots from Step 4 are presented for the *No Transparency* treatment condition.



You have completed the Basic Forecasting Phase!

You will next be shown a table summarizing your forecasting performance on the 20 clothing items that you forecasted during the Basic Forecasting Phase. You can also view the algorithm's forecasts for each of those 20 clothing items and what the algorithm's performance was; you will have access to the algorithm's forecasts to help you during the Algorithmic Forecasting Phase.

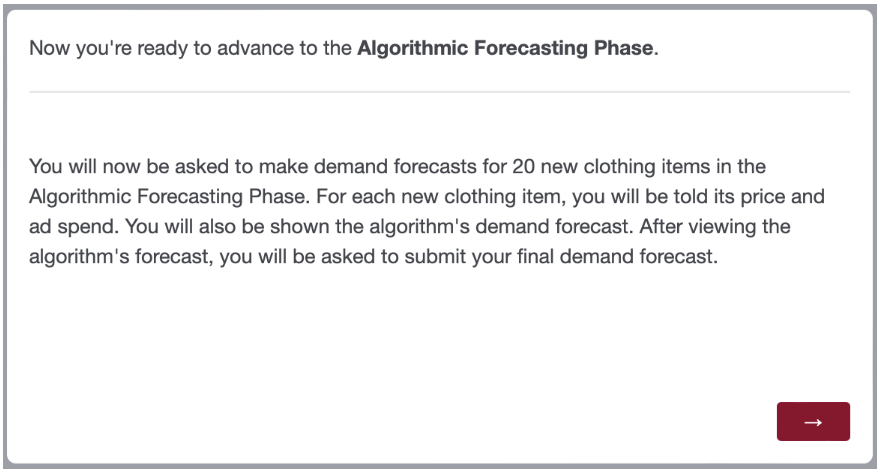


Here is a table summarizing your forecasting performance on the 20 clothing items that you forecasted during the Basic Forecasting Phase, as well as the algorithm's forecasts and performance.

Price	Ad Spend	Actual Demand	Algorithm's Forecast	Algorithm's Error	Your Error
$169	$165	419	349	70	31
$154	$92	374	394	20	64
$189	$107	312	311	1	32
$129	$90	427	443	16	67
$149	$102	409	412	3	41
$174	$150	400	338	62	40
$189	$90	305	311	6	45
$154	$101	397	394	3	53
$189	$194	430	311	119	70
$54	$164	733	645	88	83
$134	$99	428	434	6	82
$199	$107	306	297	9	74
$164	$92	351	362	11	39
$54	$97	637	645	8	13
$54	$9	529	645	116	21
$84	$154	620	558	62	30
$199	$97	287	297	10	83
$119	$21	371	464	93	51
$179	$20	237	328	91	73
$124	$185	562	453	109	112

Average Price	Average Ad Spend	Average Actual Demand	Average Algorithm's Forecast	Average Algorithm's Error	Average Your Error
$143	$107	427	420	45	55

Study the table above. Describe any observations that might be helpful to you as you proceed to the next stage.

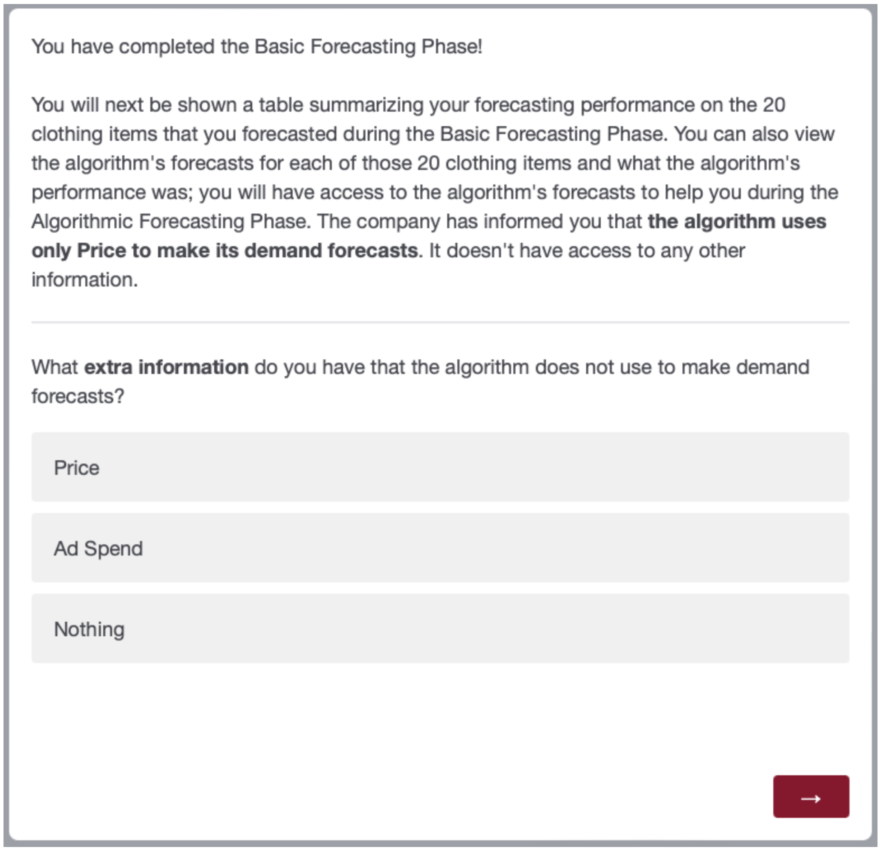


Now you're ready to advance to the **Algorithmic Forecasting Phase**.

You will now be asked to make demand forecasts for 20 new clothing items in the Algorithmic Forecasting Phase. For each new clothing item, you will be told its price and ad spend. You will also be shown the algorithm's demand forecast. After viewing the algorithm's forecast, you will be asked to submit your final demand forecast.

G.5. Step 4: Algorithm Introduction - Feature Transparency

The following screenshots from Step 4 are presented for the *Feature Transparency* treatment condition.



You have completed the Basic Forecasting Phase!

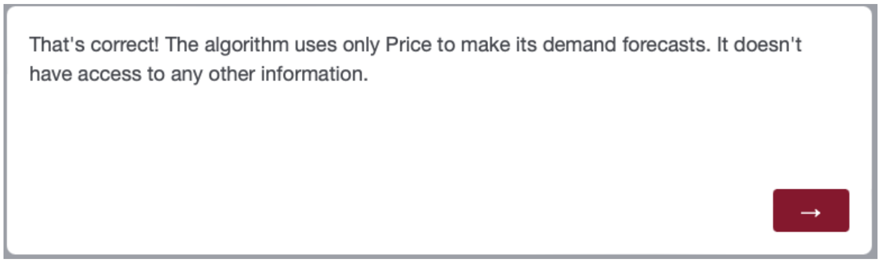
You will next be shown a table summarizing your forecasting performance on the 20 clothing items that you forecasted during the Basic Forecasting Phase. You can also view the algorithm's forecasts for each of those 20 clothing items and what the algorithm's performance was; you will have access to the algorithm's forecasts to help you during the Algorithmic Forecasting Phase. The company has informed you that **the algorithm uses only Price to make its demand forecasts**. It doesn't have access to any other information.

What **extra information** do you have that the algorithm does not use to make demand forecasts?

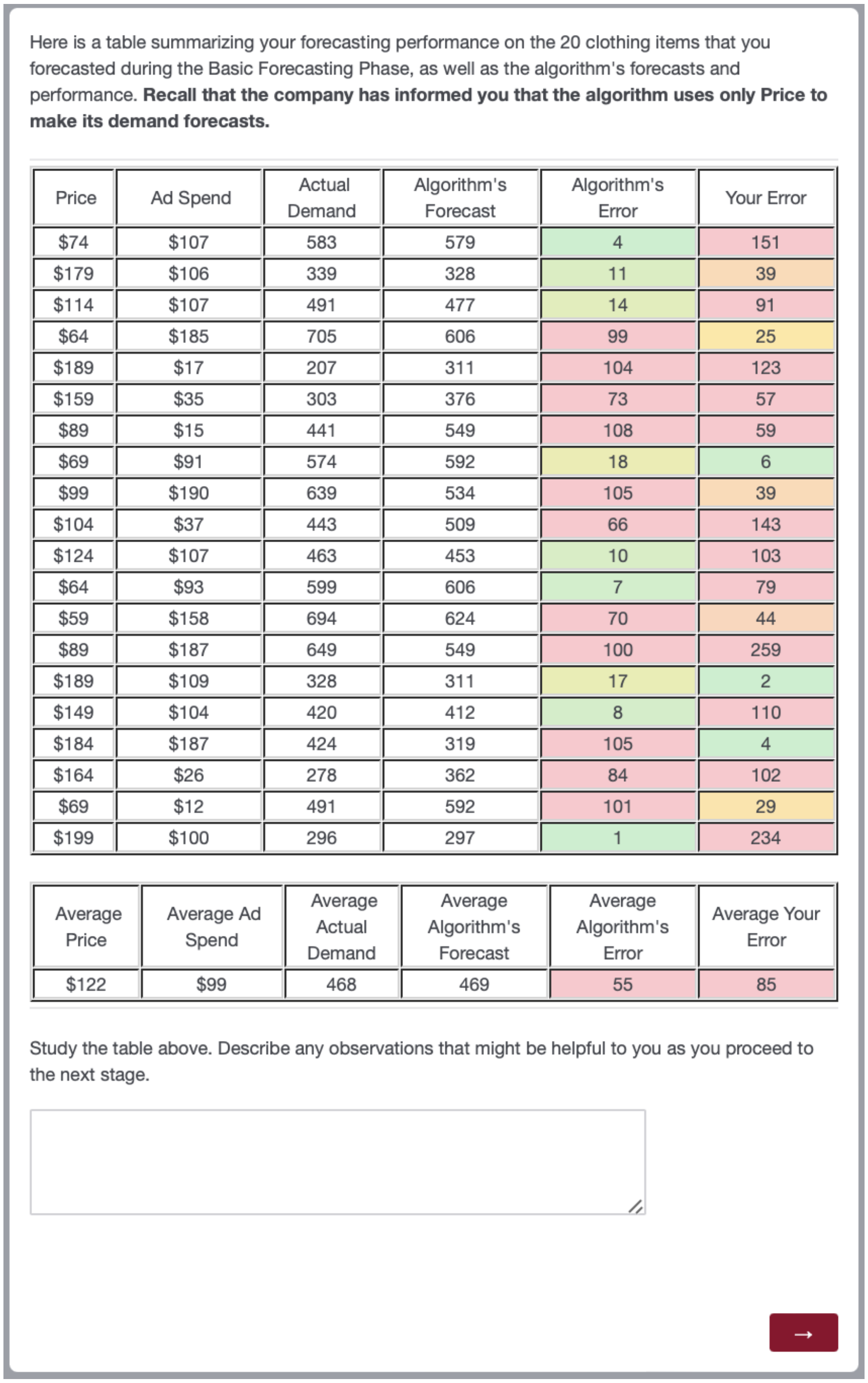
Price

Ad Spend

Nothing



That's correct! The algorithm uses only Price to make its demand forecasts. It doesn't have access to any other information.

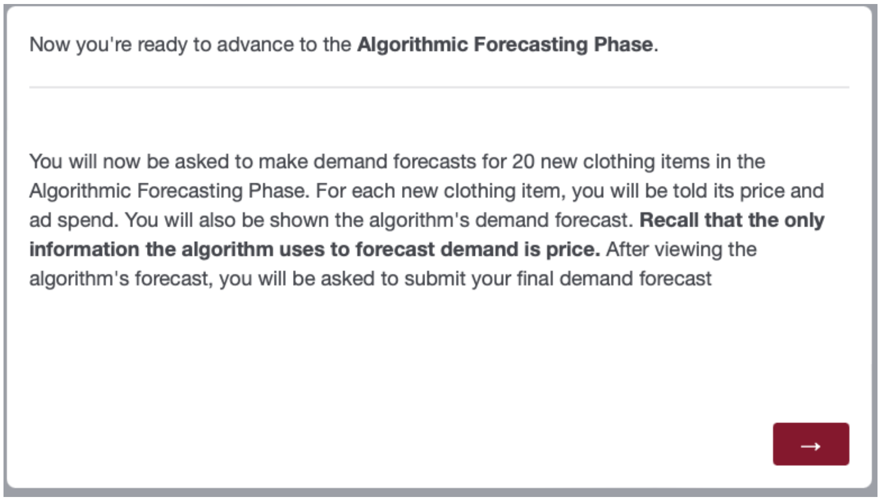


Here is a table summarizing your forecasting performance on the 20 clothing items that you forecasted during the Basic Forecasting Phase, as well as the algorithm's forecasts and performance. **Recall that the company has informed you that the algorithm uses only Price to make its demand forecasts.**

Price	Ad Spend	Actual Demand	Algorithm's Forecast	Algorithm's Error	Your Error
$74	$107	583	579	4	151
$179	$106	339	328	11	39
$114	$107	491	477	14	91
$64	$185	705	606	99	25
$189	$17	207	311	104	123
$159	$35	303	376	73	57
$89	$15	441	549	108	59
$69	$91	574	592	18	6
$99	$190	639	534	105	39
$104	$37	443	509	66	143
$124	$107	463	453	10	103
$64	$93	599	606	7	79
$59	$158	694	624	70	44
$89	$187	649	549	100	259
$189	$109	328	311	17	2
$149	$104	420	412	8	110
$184	$187	424	319	105	4
$164	$26	278	362	84	102
$69	$12	491	592	101	29
$199	$100	296	297	1	234

Average Price	Average Ad Spend	Average Actual Demand	Average Algorithm's Forecast	Average Algorithm's Error	Average Your Error
$122	$99	468	469	55	85

Study the table above. Describe any observations that might be helpful to you as you proceed to the next stage.

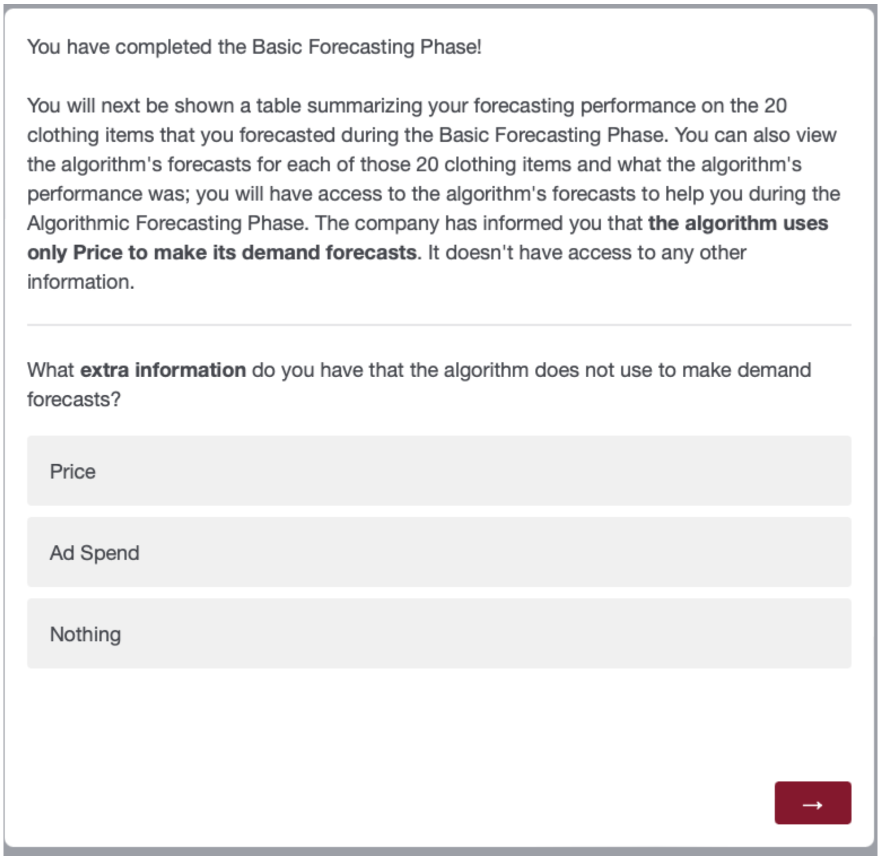


Now you're ready to advance to the **Algorithmic Forecasting Phase**.

You will now be asked to make demand forecasts for 20 new clothing items in the Algorithmic Forecasting Phase. For each new clothing item, you will be told its price and ad spend. You will also be shown the algorithm's demand forecast. **Recall that the only information the algorithm uses to forecast demand is price.** After viewing the algorithm's forecast, you will be asked to submit your final demand forecast

G.6. Step 4: Algorithm Introduction - Adjusting Nudge

The following screenshots from Step 4 are presented for the *Adjusting Nudge* treatment condition.



You have completed the Basic Forecasting Phase!

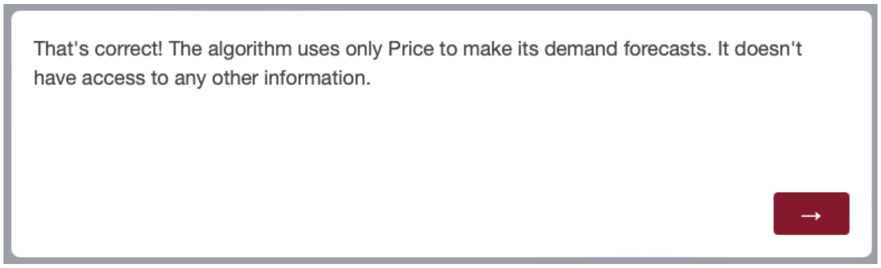
You will next be shown a table summarizing your forecasting performance on the 20 clothing items that you forecasted during the Basic Forecasting Phase. You can also view the algorithm's forecasts for each of those 20 clothing items and what the algorithm's performance was; you will have access to the algorithm's forecasts to help you during the Algorithmic Forecasting Phase. The company has informed you that **the algorithm uses only Price to make its demand forecasts**. It doesn't have access to any other information.

What **extra information** do you have that the algorithm does not use to make demand forecasts?

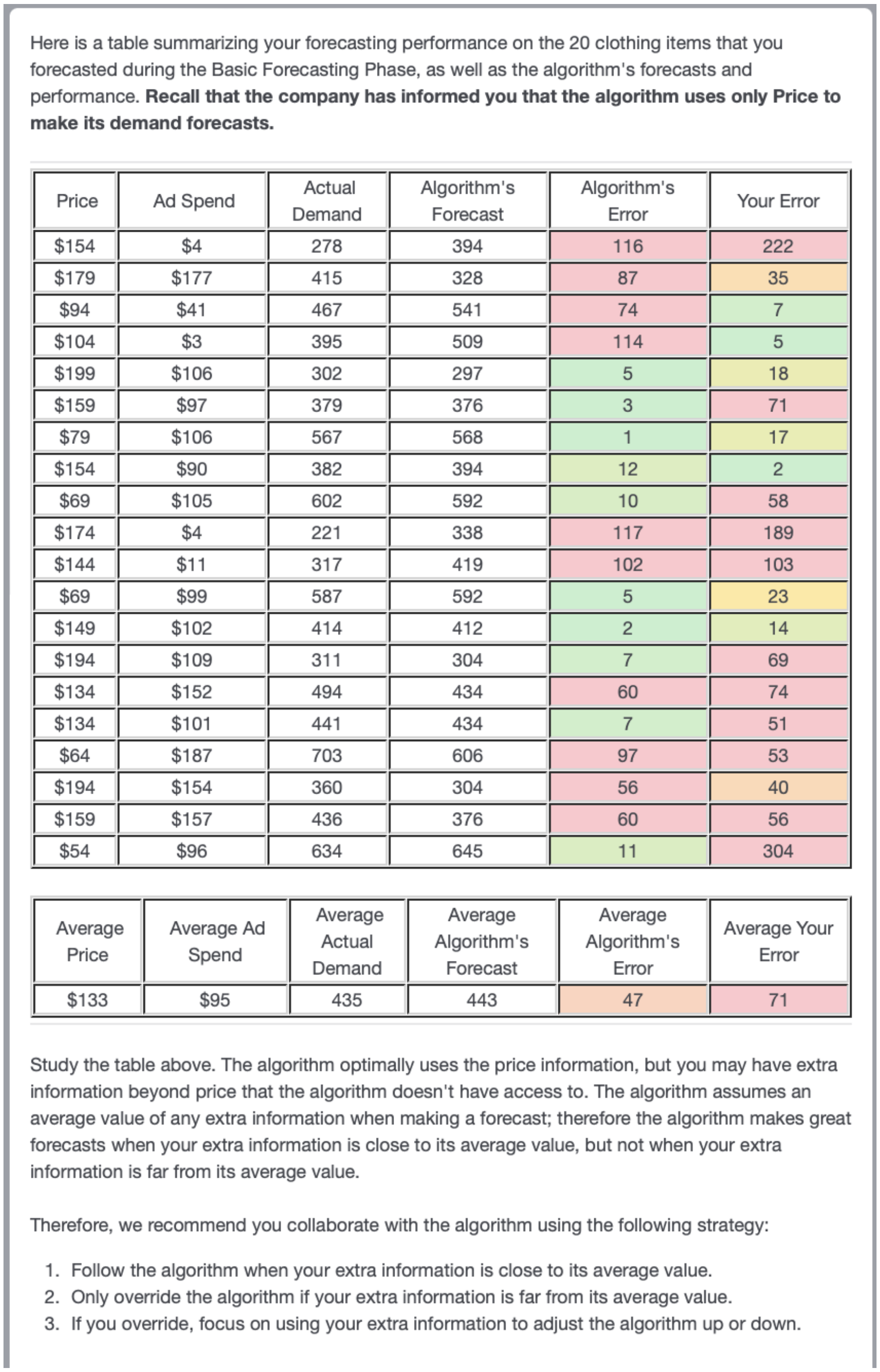
Price

Ad Spend

Nothing



That's correct! The algorithm uses only Price to make its demand forecasts. It doesn't have access to any other information.



Here is a table summarizing your forecasting performance on the 20 clothing items that you forecasted during the Basic Forecasting Phase, as well as the algorithm's forecasts and performance. **Recall that the company has informed you that the algorithm uses only Price to make its demand forecasts.**

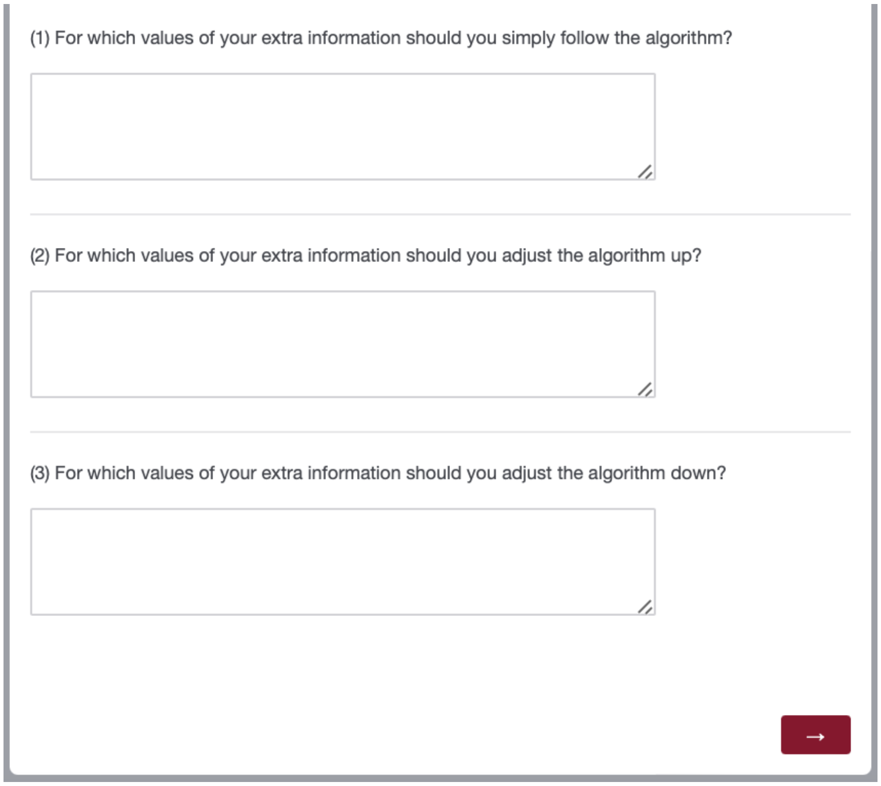
Price	Ad Spend	Actual Demand	Algorithm's Forecast	Algorithm's Error	Your Error
$154	$4	278	394	116	222
$179	$177	415	328	87	35
$94	$41	467	541	74	7
$104	$3	395	509	114	5
$199	$106	302	297	5	18
$159	$97	379	376	3	71
$79	$106	567	568	1	17
$154	$90	382	394	12	2
$69	$105	602	592	10	58
$174	$4	221	338	117	189
$144	$11	317	419	102	103
$69	$99	587	592	5	23
$149	$102	414	412	2	14
$194	$109	311	304	7	69
$134	$152	494	434	60	74
$134	$101	441	434	7	51
$64	$187	703	606	97	53
$194	$154	360	304	56	40
$159	$157	436	376	60	56
$54	$96	634	645	11	304

Average Price	Average Ad Spend	Average Actual Demand	Average Algorithm's Forecast	Average Algorithm's Error	Average Your Error
$133	$95	435	443	47	71

Study the table above. The algorithm optimally uses the price information, but you may have extra information beyond price that the algorithm doesn't have access to. The algorithm assumes an average value of any extra information when making a forecast; therefore the algorithm makes great forecasts when your extra information is close to its average value, but not when your extra information is far from its average value.

Therefore, we recommend you collaborate with the algorithm using the following strategy:

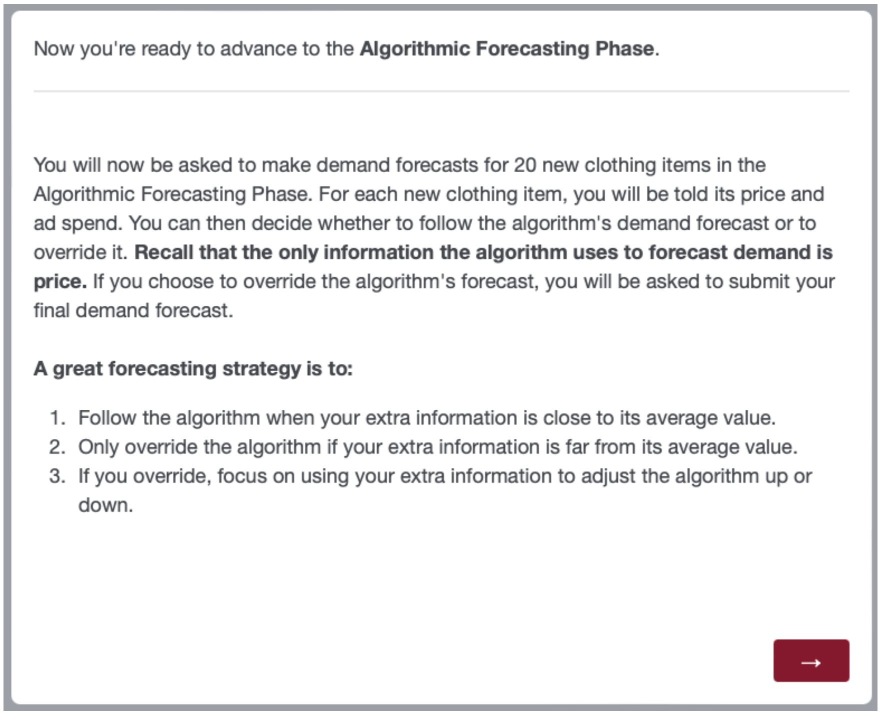
1. Follow the algorithm when your extra information is close to its average value.
2. Only override the algorithm if your extra information is far from its average value.
3. If you override, focus on using your extra information to adjust the algorithm up or down.



(1) For which values of your extra information should you simply follow the algorithm?

(2) For which values of your extra information should you adjust the algorithm up?

(3) For which values of your extra information should you adjust the algorithm down?



Now you're ready to advance to the **Algorithmic Forecasting Phase**.

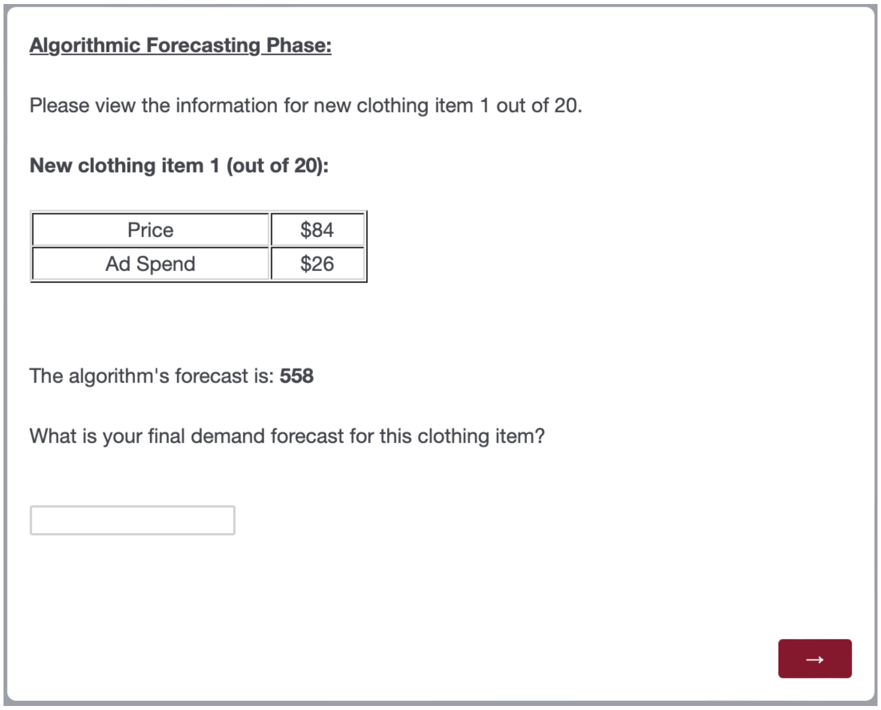
You will now be asked to make demand forecasts for 20 new clothing items in the Algorithmic Forecasting Phase. For each new clothing item, you will be told its price and ad spend. You can then decide whether to follow the algorithm's demand forecast or to override it. **Recall that the only information the algorithm uses to forecast demand is price.** If you choose to override the algorithm's forecast, you will be asked to submit your final demand forecast.

A great forecasting strategy is to:

1. Follow the algorithm when your extra information is close to its average value.
2. Only override the algorithm if your extra information is far from its average value.
3. If you override, focus on using your extra information to adjust the algorithm up or down.

G.7. Step 5: Demand Predictions with Algorithm - No Transparency

The following screenshots from Step 5 are presented for the *No Transparency* treatment condition.



Algorithmic Forecasting Phase:

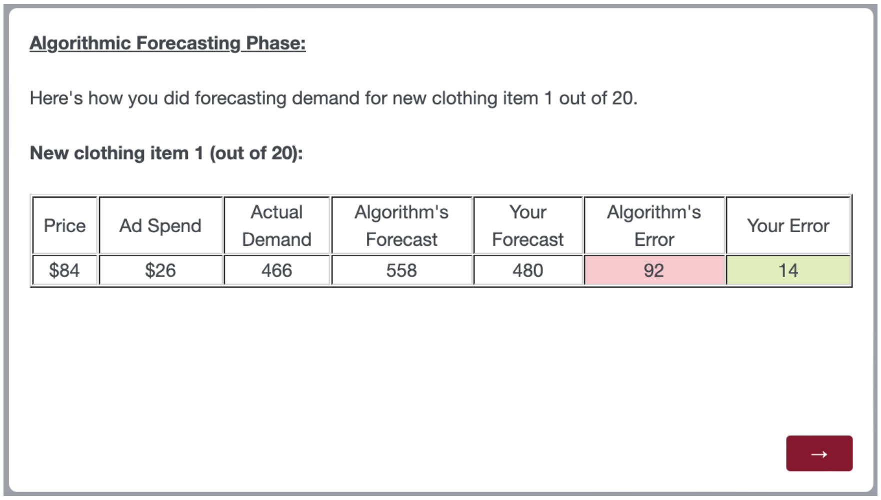
Please view the information for new clothing item 1 out of 20.

New clothing item 1 (out of 20):

Price	$84
Ad Spend	$26

The algorithm's forecast is: **558**

What is your final demand forecast for this clothing item?



Algorithmic Forecasting Phase:

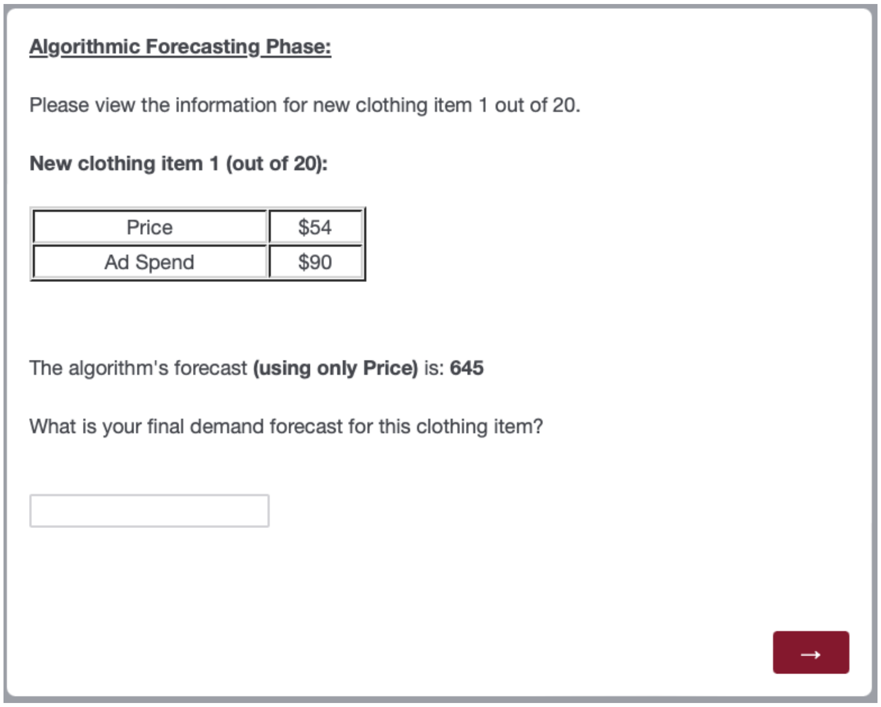
Here's how you did forecasting demand for new clothing item 1 out of 20.

New clothing item 1 (out of 20):

Price	Ad Spend	Actual Demand	Algorithm's Forecast	Your Forecast	Algorithm's Error	Your Error
$84	$26	466	558	480	92	14

G.8. Step 5: Demand Predictions with Algorithm - Feature Transparency

The following screenshots from Step 5 are presented for the *Feature Transparency* treatment condition.



Algorithmic Forecasting Phase:

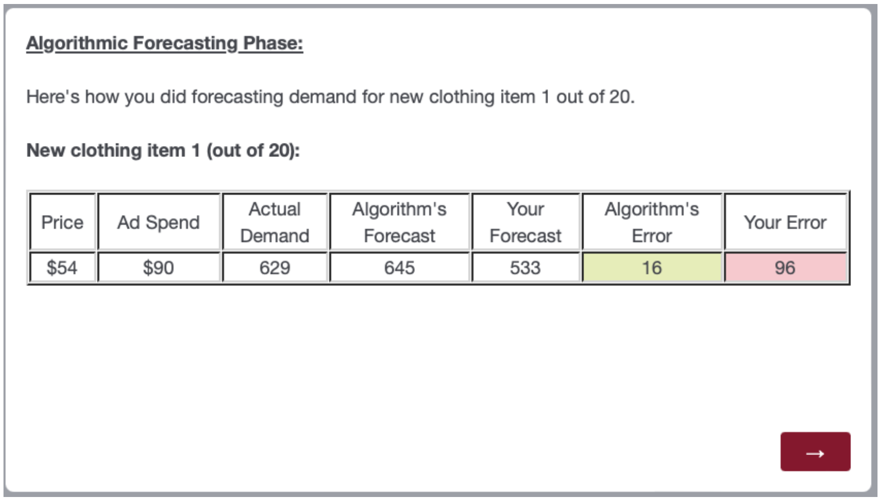
Please view the information for new clothing item 1 out of 20.

New clothing item 1 (out of 20):

Price	$54
Ad Spend	$90

The algorithm's forecast **(using only Price)** is: **645**

What is your final demand forecast for this clothing item?



Algorithmic Forecasting Phase:

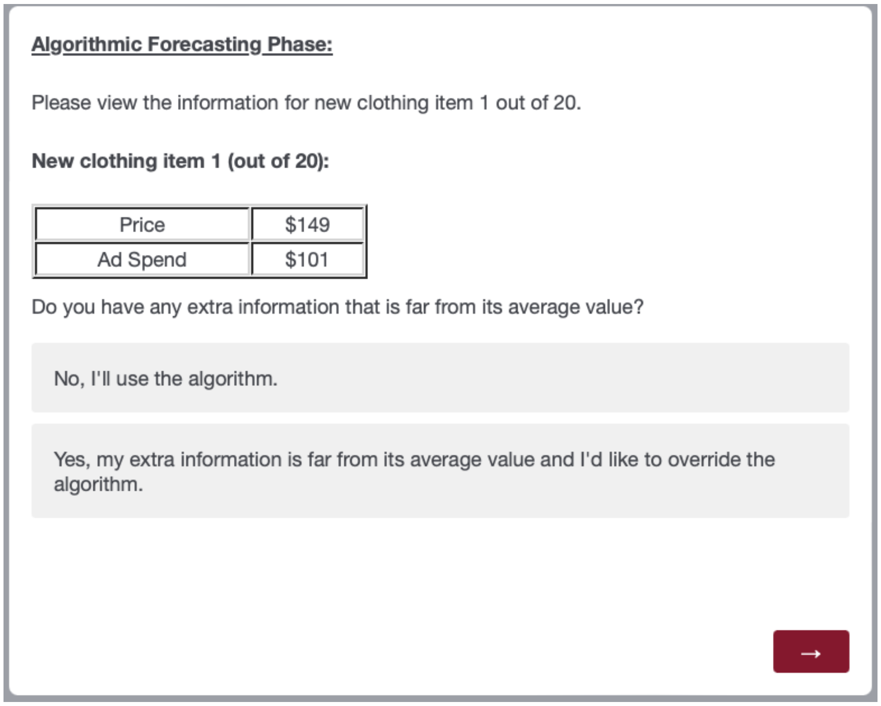
Here's how you did forecasting demand for new clothing item 1 out of 20.

New clothing item 1 (out of 20):

Price	Ad Spend	Actual Demand	Algorithm's Forecast	Your Forecast	Algorithm's Error	Your Error
$54	$90	629	645	533	16	96

G.9. Step 5: Demand Predictions with Algorithm - Adjusting Nudge

The following screenshots from Step 5 are presented for the *Adjusting Nudge* treatment condition.



Algorithmic Forecasting Phase:

Please view the information for new clothing item 1 out of 20.

New clothing item 1 (out of 20):

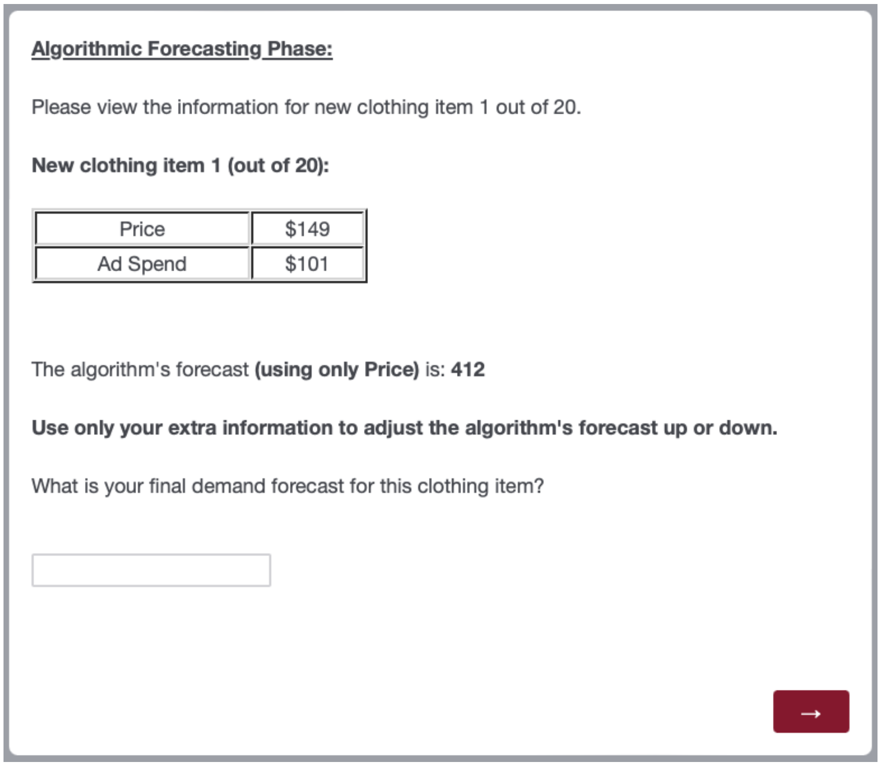
Price	$149
Ad Spend	$101

Do you have any extra information that is far from its average value?

No, I'll use the algorithm.

Yes, my extra information is far from its average value and I'd like to override the algorithm.

If the participant selects "No, I'll use the algorithm," then the algorithm's demand prediction, $\hat{y}_i^{alg}$, is recorded as their final demand prediction, $\hat{y}_i^{final}$. If they select "Yes, my extra information is far from its average value and I'd like to override the algorithm," then they see the following screen:



Algorithmic Forecasting Phase:

Please view the information for new clothing item 1 out of 20.

New clothing item 1 (out of 20):

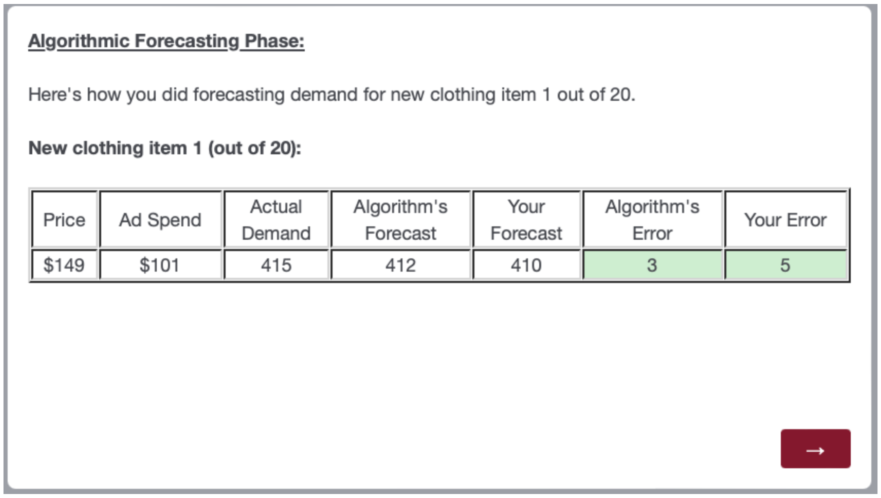
Price	$149
Ad Spend	$101

The algorithm's forecast **(using only Price)** is: **412**

Use only your extra information to adjust the algorithm's forecast up or down.

What is your final demand forecast for this clothing item?

Regardless of whether the participant selects “No, I’ll use the algorithm” or “Yes, my extra information is far from its average value and I’d like to override the algorithm,” they will see the following screen:



Algorithmic Forecasting Phase:

Here's how you did forecasting demand for new clothing item 1 out of 20.

New clothing item 1 (out of 20):

Price	Ad Spend	Actual Demand	Algorithm's Forecast	Your Forecast	Algorithm's Error	Your Error
$149	$101	415	412	410	3	5