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

December 8, 2022

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 a lab experiment where participants are tasked with making demand predictions for 20 products while having access to an algorithm’s recommendations, we confirm this bias towards NAW and find that it leads to a 20-61% increase in prediction error. In a follow-up experiment, we find that feature transparency – even when the underlying algorithm is a black box – helps users more effectively discriminate when and how to deviate from algorithms, resulting in a 25% reduction in prediction error.

Key words: human-algorithm interaction, forecasting, behavioral operations, cognitive bias, algorithm transparency, information aggregation, wisdom of crowds, 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). These examples imply that organizations would be better off not letting humans override algorithms at all.

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 for a product) 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 then narrow our attention to 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* (see also §2.2).

We define such *private information* pragmatically: 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); judges making bail decisions see defendants in court, though algorithms only use information from before the appearance (Kleinberg et al. 2017). 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 has been studied extensively in decision analysis and judgment and decision making (see §2.2). We can understand the best possible overriding decision as the one that follows the optimal aggregation policy. And, we can leverage the established literature that examines how humans approach aggregation tasks 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 rational 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 optimal solution may not even lie between the algorithm's recommendation and the person's own initial prediction.

Next, we conduct controlled laboratory experiments that seek to test the over- and under-adherence pattern predicted by our theory. Laboratory experiments allow us direct access to and manipulation of people's private information, while controlling for any differences in the algorithm and rational benchmark. 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 effectively differentiate when their private information was valuable (though a rational participant could).

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* as well as another literature-inspired *training data transparency* aimed at increasing people's overall trust in algorithms. 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*. In contrast, while *training data transparency* marginally increased adherence, it did so both when it is helpful and when it is not, leading to no significant improvement in performance. Furthermore, 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 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 laboratory experiment 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 show it can significantly improve human-algorithm collaborative performance by addressing NAW in a manner that other algorithm transparency initiatives do not.

2. Literature Review

We describe our contribution to the literature by discussing research on algorithm overriding, aggregation strategies, and algorithm transparency.

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: manager overrides of a SKU replenishment algorithm increased inventory costs (van Donselaar et al. 2010), doctor overrides of task-scheduling algorithms decreased productivity (Ibanez et al. 2018), warehouse worker overrides of box-packing algorithms increased packing times (Sun et al. 2022), vending machine manager overrides of assortment algorithms decreased revenue Kawaguchi (2021), 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 if we are to help humans override in such a way that consistently yields improvement over the algorithm alone. We contribute by developing and testing behavioral theory describing override decision-making that explains how overrides tend to be suboptimal and helps prescribe interventions to increase their effectiveness.

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). Overriding may be an expression of people's preference for more control over a decision (Dietvorst et al. 2018). They may also (mistakenly) feel like the task itself is too subjective for an algorithm (Castelo et al. 2019). People may also be more tolerant of their own mistakes relative to an algorithm's mistakes (Dietvorst et al. 2015). They may also perceive an algorithm as too simple or too complex (Lehmann et al. 2022). Despite these many psychological reasons why people may sometimes be reluctant to fully accept algorithms, we note that there is *not* uniform evidence that people prefer human judgment over algorithms (in fact, Logg et al. 2019 suggests the opposite). 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. Thus, we contribute to the psychology of overriding *not* by examining a psychological preference against algorithms, but by examining people's cognitive limitations in a setting with rational reasons to override.

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 or predictions. 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. 2022). 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 setting where the algorithm makes the final decision using the human's prediction as input; in contrast, we seek to address the shared information problem when the human makes the final judgment using the algorithm's prediction as input.

Relatedly, how do people make final decisions when they receive a recommended prediction from an external advisor? Several researchers have employed the "Judge-Advisor System" (JAS) paradigm to study this question, first primarily to examine human external advisors (e.g., Bonaccio and Dalal 2006, Soll et al. 2021), then also to examine algorithmic advisors (Logg et al. 2019, Lehmann et al. 2022). In JAS, a human judge first forms their initial prediction, then they receive advice from an advisor in the form of a recommended prediction, 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. These 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: a WOA between 0 and 1, inclusive. In other words, people's final predictions are generally consistent with taking some weighted average between their initial prediction and the external advice, or choosing one of the extremes. Like several of these papers, we examine how people take the advice of an algorithm and also report WOA measurements. However, we examine settings in which the participant has objective and measurable reasons to deviate from the algorithm's advice that may vary from question to question. Moreover, we examine how participants can improve their performance by deviating from an advice-weighting approach (i.e., a WOA greater than 1 or less than 0).

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 reduce the negative impact of human's random errors (e.g., see "bootstrapping" techniques, Dawes et al. 1989) or 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 (e.g., changing the algorithm in anticipation of overrides as in Sun et al. 2022), 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 for a given set of inputs (Lipton 2017). In contrast, the type of transparency we consider in this paper 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 for any given input (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 or not algorithm 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. Poursabzi-Sangdeh et al. (2021) find that transparency that helps people simulate a model's predictions did not necessarily increase their observed trust in it. 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

In this section, we first outline our model setting, including how two types of features – private and public – impact actual outcomes and algorithmic predictions. We then hypothesize how human decision-makers combine algorithmic predictions with their own initial predictions to make their final predictions, which describes a class of human behavior models we term "advice weighting". We present and compare two extreme behaviors within this class – "naïve" and "sophisticated" advice weighting – and use a simple example to build intuition and motivate our hypotheses tested in our experimental studies in §4 and §5.

3.1. Model

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 independent, zero-mean random noise ϵ for each instance i, i.e.,

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

This function is unknown to the human decision-maker, who is tasked with predicting outcome Y_i given feature vector $(\boldsymbol{x}_i^{pub}, \boldsymbol{x}_i^{priv})$. In addition, the decision-maker has access to an algorithm that she can use to help predict Y_i. This algorithm uses only public features as inputs, and outputs a prediction $\hat{y}_i^{alg}$, i.e.,

$$\hat{y}_i^{alg} = f_{alg}(\boldsymbol{x}_i^{pub}) \; . \tag{2}$$

Importantly, the gap between the expected outcome $\mathbb{E}[Y_i]$ and the algorithm's prediction $\hat{y}_i^{alg}$ represents the potential improvement that the human can make over the algorithm's prediction. Note that this improvement 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 $\boldsymbol{x}_i^{pub}$. Although not necessary for our theory, for ease of exposition and in line with our experiments, it is helpful to assume that the algorithm optimally uses $\boldsymbol{x}_i^{pub}$ when making its predictions, and thus any potential improvement that the human can make over the algorithm's prediction is due to the 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 instance i, v_i, is $v_i \triangleq \mathbb{E}[Y_i] - \hat{y}_i^{alg}$.*

Finally, we note that the best possible prediction of Y_i given feature vector $(\boldsymbol{x}_i^{pub}, \boldsymbol{x}_i^{priv})$ is simply $f_{actual}(\boldsymbol{x}_i^{pub}, \boldsymbol{x}_i^{priv}) = \mathbb{E}[Y_i]$. Thus, we will consider the benchmark of a *hyper-rational* human, who – with access to enough historical data linking outcomes to feature vectors – fully recovers $f_{actual}(\cdot)$ and uses $\mathbb{E}[Y_i]$ as her prediction for instance i. We believe that it is unlikely for humans to make predictions in such a hyper-rational way, and thus in the following subsection, we present a behavioral model depicting how humans may make predictions in practice.

3.2. Advice Weighting Behavior

Following the advice weighting literature summarized in §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. Specifically, we model human j's *initial prediction* for instance i as

$$\hat{Y}_{ij}^{init} = f_{init,j}(\boldsymbol{x}_i^{pub}, \boldsymbol{x}_i^{priv}) + \eta_j \; . \tag{3}$$

Here, η_j is a zero-mean, bounded random noise independent of $(\boldsymbol{x}_i^{pub}, \boldsymbol{x}_i^{priv})$, reflecting the idea that humans are boundedly rational and make noisy predictions (e.g., Su 2008, Kahneman et al. 2022). The human's final prediction is then

$$\hat{Y}_{ij}^{final} = \lambda_{ij}\hat{y}_i^{alg} + (1-\lambda_{ij})\hat{Y}_{ij}^{init}, \tag{4}$$

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.

3.2.1. Naïve Advice Weighting We believe many humans are biased towards *naïve advice weighting*. We define a *naïve advice weighter* as a human who places 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. Although a naïve advice weighter may choose any $\lambda \in [0,1]$, for the remainder of §3.2, we will consider a hypothetical, optimal choice of λ; this will help us identify the pitfalls of naïve advice weighting even when it achieves its best-case performance. In §4, we empirically study advice weighting without assuming an optimal choice of λ.

Given a set $\mathcal{S}$ of instances characterized by $(\boldsymbol{x}_k^{pub}, \boldsymbol{x}_k^{priv})\ \forall k \in \mathcal{S}$, the optimal λ for the naïve advice weighting strategy, $\lambda^{NAW}(\hat{\boldsymbol{y}}^{init})$, can be found as a function of the vector of realized initial predictions $\hat{\boldsymbol{y}}^{init}$ consisting of components $\hat{y}_k^{init}\ \forall k \in \mathcal{S}$ using the following optimization model:

$$NAW(\hat{\boldsymbol{y}}^{init}):\ \min_{\lambda \in [0,1]} \sum_{k \in \mathcal{S}} \left(\mathbb{E}[Y_k] - (\lambda \hat{y}_k^{alg} + (1-\lambda)\hat{y}_k^{init})\right)^2. \tag{5}$$

$NAW(\hat{\boldsymbol{y}}^{init})$ minimizes the sum of squared errors between the human's final predictions, $\hat{y}_k^{final}$, and the hyper-rational human's prediction, $\mathbb{E}[Y_k]$ for all $k \in \mathcal{S}$. Note that this solution is unachievable in practice since $\mathbb{E}[Y_k]$ is unknown; nonetheless, $\lambda^{NAW}(\hat{\boldsymbol{y}}^{init})$ represents the human's best-case weight on the algorithm's predictions given her own initial predictions if she were to use the naïve advice weighting strategy. Note that an optimal solution $\lambda^{NAW}(\hat{\boldsymbol{y}}^{init})$ can be determined for any realization $\hat{\boldsymbol{y}}^{init}$. Thus, we can characterize the optimal solution as a function of random variables $\hat{Y}_k^{init}\ \forall k \in \mathcal{S}$, which we will define as random variable Λ^{NAW}.

3.2.2. Sophisticated Advice Weighting To understand errors that could arise from using the naïve advice weighting strategy, we next consider the hypothetical, best-case advice weighting strategy when the weight on the algorithm's prediction is not constrained to be identical across all instances. Namely, we define a *sophisticated advice weighter* as a human who solves the following optimization model to determine the optimal weights $\lambda_k^{SAW}(\hat{\boldsymbol{y}}^{init})$ for each instance $k \in \mathcal{S}$:

$$SAW(\hat{\boldsymbol{y}}^{init}):\ \min_{\lambda_k \in [0,1] \forall k \in \mathcal{S}} \sum_{k \in \mathcal{S}} \left(\mathbb{E}[Y_k] - (\lambda_k \hat{y}_k^{alg} + (1-\lambda_k)\hat{y}_k^{init})\right)^2. \tag{6}$$

Since the optimal solution can be determined for any realization $\hat{\boldsymbol{y}}^{init}$, we can characterize the optimal solution $\Lambda_k^{SAW}\ \forall k \in \mathcal{S}$ as a function of random variables $\hat{Y}_k^{init}\ \forall k \in \mathcal{S}$.

The following proposition shows that the sophisticated advice weighting strategy is superior to the naïve advice weighting strategy.

Proposition 1. *Let $OPT^{SAW}(\hat{\boldsymbol{y}}^{init})$ and $OPT^{NAW}(\hat{\boldsymbol{y}}^{init})$ represent the optimal values of $SAW(\hat{\boldsymbol{y}}^{init})$ and $NAW(\hat{\boldsymbol{y}}^{init})$, respectively. Then $OPT^{SAW}(\hat{\boldsymbol{y}}^{init}) \leq OPT^{NAW}(\hat{\boldsymbol{y}}^{init})$.*

Proving Proposition 1 simply requires showing that $NAW(\hat{\boldsymbol{y}}^{init})$ is identical to $SAW(\hat{\boldsymbol{y}}^{init})$ except that it further constrains weights $\lambda_k \ \forall k \in \mathcal{S}$; formal proofs are in Appendix A. To build intuition and better understand why the sophisticated advice weighting strategy is superior, we consider a special case and compare the optimal weights identified for each strategy. This analysis will guide our experimental design and hypotheses.

Consider a special case with only four possible values of v_k for each instance k: $v_k \in \{v_L, -v_L, v_H, -v_H\}$, with $0 \leq v_L < v_H$. We define $\mathcal{S}_{L^+}, \mathcal{S}_{L^-}, \mathcal{S}_{H^+}, \mathcal{S}_{H^-}$ as the set of instances with those corresponding impacts of private features, e.g. $\mathcal{S}_{L^+} = \{k \in \mathcal{S} : v_k = v_L\}$. Furthermore, we define the human's initial prediction errors as $Z_k = \mathbb{E}[Y_k] - \hat{Y}_k^{init}$. To simplify the analysis, we assume that these errors are equal in distribution across sets, i.e., $Z_k =_d Z \ \forall k \in \mathcal{S}$.

Lemma 1. *For the special case where all instances $k \in \mathcal{S}$ can be partitioned into $\mathcal{S}_{L^+}, \mathcal{S}_{L^-}, \mathcal{S}_{H^+}$, and $\mathcal{S}_{H^-}$, and where $Z_k =_d Z \ \forall k \in \mathcal{S}$, we have (a) within each set, Λ_k^{SAW} is equal in distribution, i.e., $\Lambda_k^{SAW} =_d \Lambda_{L^+}^{SAW} \ \forall k \in \mathcal{S}_{L^+}$, $\Lambda_k^{SAW} =_d \Lambda_{L^-}^{SAW} \ \forall k \in \mathcal{S}_{L^-}$, $\Lambda_k^{SAW} =_d \Lambda_{H^+}^{SAW} \ \forall k \in \mathcal{S}_{H^+}$, and $\Lambda_k^{SAW} =_d \Lambda_{H^-}^{SAW} \ \forall k \in \mathcal{S}_{H^-}$ and (b) $\Lambda_{L^+}^{SAW} \succcurlyeq_S \Lambda_{H^+}^{SAW}$ and $\Lambda_{L^-}^{SAW} \succcurlyeq_S \Lambda_{H^-}^{SAW}$.*

Here, $=_d$ denotes equality in distribution and $\succcurlyeq_S$ denotes first-order stochastic dominance. Next, we define Λ_L^{SAW} (Λ_H^{SAW}) as the sophisticated advice weighter's average weight on the algorithm across instances where the absolute impact of private features is low (high), i.e.,

$$\Lambda_L^{SAW} = \frac{1}{|\mathcal{S}_{L^+}| + |\mathcal{S}_{L^-}|} \sum_{k \in \{\mathcal{S}_{L^+}, \mathcal{S}_{L^-}\}} \Lambda_k^{SAW}, \text{ and}$$

$$\Lambda_H^{SAW} = \frac{1}{|\mathcal{S}_{H^+}| + |\mathcal{S}_{H^-}|} \sum_{k \in \{\mathcal{S}_{H^+}, \mathcal{S}_{H^-}\}} \Lambda_k^{SAW}.$$

Proposition 2. *Consider the special case defined in Lemma 1 with $|\mathcal{S}_{L^-}| = |\mathcal{S}_{H^-}|$ and $|\mathcal{S}_{L^+}| = |\mathcal{S}_{H^+}|$. We have (a) $\Lambda_L^{SAW} \succcurlyeq_S \Lambda_H^{SAW}$, and (b) $\lim_{v_L \to 0} \mathbb{E}[\Lambda_L^{SAW}] = 1$ and $\lim_{v_H \to \infty} \mathbb{E}[\Lambda_H^{SAW}] = 0$.*

Proposition 2 shows that, on average, the sophisticated advice weighter adheres more to the algorithm across instances where the impact of their private features is low, but adheres less to the algorithm across instances where the impact of their private features is high. Moreover, if the impact of private features is low (high) enough, then they *fully* adhere to (ignore) the algorithm. Our next proposition builds intuition as to how the naïve advice weighter's optimal weight compares to the sophisticated advice weighter's optimal weights.

Proposition 3. *Consider the special case defined in Lemma 1 with $|\mathcal{S}_{L^-}| = |\mathcal{S}_{H^-}| = |\mathcal{S}_{L^+}| = |\mathcal{S}_{H^+}| = 1$, and where Z is a two-point distribution taking values $\{-c, c\}$ each with probability $\frac{1}{2}$ for some $v_L \leq c \leq v_H \leq 2v_L$. We have $\mathbb{E}[\Lambda_L^{SAW}] > \mathbb{E}[\Lambda^{NAW}] > \mathbb{E}[\Lambda_H^{SAW}]$.*

Proposition 3 shows that for our simple example, the naïve advice weighter's optimal weight falls between the sophisticated advice weighter's optimal weights for instances with low vs. high impact of private features. The naïve advice weighter does not differentiate amongst instances where there is less vs. more impact of private features, and instead applies an identical weight across all instances as though they all had an equal impact of private features. This leads to the naïve advice weighter under-weighting the algorithm's prediction when the absolute impact of private features is below average, and over-weighting the algorithm's prediction when the absolute impact of private features is above average.

3.2.3. Comparison to Hyper-Rational Benchmark Compared to the hyper-rational benchmark, the naïve advice weighting strategy is suboptimal for two reasons: (i) differential weights are not applied across instances with different impact of private features, i.e., $\lambda_i = \lambda$ for every instance i, and (ii) advice weighting in general – restricting weights λ_i to be in the interval $[0, 1]$ – may not contain the optimal solution. Reason (i) is well-captured by comparing the sophisticated and naïve advice-weighting models above. However, reason (ii) has to do with the entire class of advice-weighting policies.

Figure 1 illustrates what we call the *advice-weighting region* – the interval between a person's initial prediction and the algorithm, which corresponds to $\lambda_i \in [0, 1]$. It is important to recognize that the optimal final prediction $\mathbb{E}[Y_i]$ may not always lie within the advice-weighting region; when this is the case, even $SAW(\hat{\boldsymbol{y}}^{init})$ leads to suboptimal predictions. By removing the constraints $\lambda_i \in [0, 1]$, one can show that $SAW(\hat{\boldsymbol{y}}^{init})$ yields equivalent predictions to the hyper-rational benchmark.

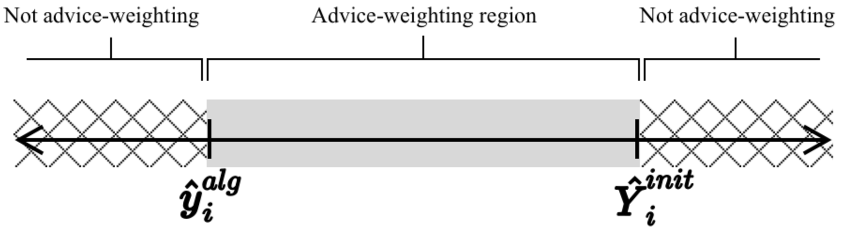


Figure 1 **Depiction of the advice-weighting region**

3.3. Hypotheses & Preview of Experiments

Our first study, presented in §4, is designed to test the key intuition developed from Propositions 1 - 3. To do this, we use three conditions in which participants experience (i) only instances 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 realized outcome 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 the weight that participants place on the algorithm's predictions. The only difference across conditions is the impact of private features that participants experience. For the first two conditions, since the participants experience an approximately identical impact of private features across instances (i.e., low or high), their average weight on the algorithm's predictions gives us an empirical estimate of the weight a sophisticated advice weighter – someone who differentially applies weights based on the impact of private features – would place on the algorithm. If the participants in the third condition use a sophisticated advice weighting strategy, then they would be able to distinguish between the instances with low and high impact of private features, differentially weighting the algorithm's predictions; specifically, their weight on the algorithm's prediction for instances with low (high) impact of private features should not be statistically different than the weight that participants in the first (second) condition apply. However, we believe that humans are unable to sufficiently distinguish between instances with low and high impact of private features and are instead biased towards naïve advice weighting, leading to the following two hypotheses.

Hypothesis 1. *Humans who experience instances of both low and high impact of private features under-weight the algorithm's predictions when the impact of private features is low and over-weight the algorithm's predictions when the impact of private features is high, compared to humans who experience either always low or always high impact of private features, respectively.*

Hypothesis 2. *For instances with low (high) impact of private features, the average prediction error made by humans who experience instances of both low and high impact of private features is larger than the average prediction error made by humans who experience only low (high) impact of private features.*

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 naïve advice weighting behavior would be to specify which features are public vs. private. Since it is often not possible to tell humans what their private features are and when they are important, we instead study the impact of 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. *When the impact of private features is low (high), humans who are provided feature transparency place more (less) weight on the algorithm's predictions than humans who are provided no transparency.*

Hypothesis 4. *The average prediction error made by humans who are provided feature transparency is smaller than the average prediction error made by humans who are provided no transparency for both subsets of products with low and high impact of private features.*

4. Study 1: Humans Exhibit Bias Towards Naïve Advice Weighting, Degrading Prediction Accuracy

Study 1 tests Hypotheses 1 and 2 in a controlled lab 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 D.

1. *Instructions & Comprehension Checks.* Participants are introduced to the demand prediction task and objective of minimizing absolute prediction error, and are tested for comprehension.

[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.

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.
3. *Demand Predictions without Algorithm.* Sequentially for each of $i = 1, ..., 20$ new products, 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 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 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, 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 their absolute prediction error, $|\hat{y}_i^{final} - y_i|$, and the algorithm's error, $|\hat{y}_i^{alg} - y_i|$.

Participants' demand predictions in Step 3 ($\hat{y}_i^{init}$) and Step 5 ($\hat{y}_i^{final}$) were incentivized for accuracy. Namely, participants received a base compensation of \$7 for completing the experiment 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 each product i, actual demand is generated by the equation:

$$Y_i = 131 + 1.6x_i^{pub} + 0.75x_i^{priv} + \epsilon_i, \tag{7}$$

where ϵ_i is drawn from a normal distribution with mean 0 and standard deviation 4. The values of x_i^{pub} are drawn from a discrete uniform distribution with support $\{20, 80\}$. The values of x_i^{priv} are 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 by the equation:

$$\hat{y}_i^{alg} = 131 + 1.6x_i^{pub}. \tag{8}$$

Due to our data generation process, one can easily show that the algorithm makes the best possible demand predictions conditional on being constrained to only using x_i^{pub}. Thus the difference between the expected outcome, $\mathbb{E}[Y_i]$, and the algorithm's prediction, $\hat{y}_i^{alg}$, i.e., the impact of private feature, is simply $v_i = 0.75x_i^{priv}$. Notably, v_i is fully determined by x_i^{priv}.

Participants are not explicitly given equations (7) and (8), 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 hyper-rational participant could theoretically recover (7) and make optimal predictions $\mathbb{E}[Y_i] = 131 + 1.6x_i^{pub} + 0.75x_i^{priv}$. We will include this hypothetical, hyper-rational participant as a benchmark in our analyses.

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. 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 shows how our three conditions relate to this 2×2 mixed design.

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 (i.e., experience a mixed exposure set). We use participants in the other two conditions to represent 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. 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

$$WOA_{ij} = \min\Big(\max\Big(\frac{\hat{y}_{ij}^{final} - \hat{y}_{ij}^{init}}{\hat{y}_i^{alg} - \hat{y}_{ij}^{init}}, 0\Big), 1\Big). \tag{9}$$

We note that WOA_{ij} is an estimate for λ_{ij} defined for advice weighting behavior in (4), 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

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

$$MedWOA_j^H = \text{median}(WOA_{ij} \; \forall i \; s.t. \; x_i^{priv} \in \{-150, -50\} \cup \{50, 150\}). \tag{11}$$

$MedWOA_j^L$ ($MedWOA_j^H$) 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 participants in the *Mixed* $|v_i|$ condition will have values for both $MedWOA_j^L$ and $MedWOA_j^H$, whereas participants in the *Always Low* $|v_i|$ (*Always High* $|v_i|$) condition will have values only for $MedWOA_j^L$ ($MedWOA_j^H$) since they 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

$$RMedSE_j^L = \sqrt{\text{median}([\hat{y}_{ij}^{final} - y_i]^2 \; \forall i \; s.t. \; x_i^{priv} \in \{-10, 10\}}); \tag{12}$$

$$RMedSE_j^H = \sqrt{\text{median}([\hat{y}_{ij}^{final} - y_i]^2 \; \forall i \; s.t. \; x_i^{priv} \in \{-150, -50\} \cup \{50, 150\}}). \tag{13}$$

These *RMedSE* metrics are measures of the participant's prediction accuracy. As we did for *MedWOA*, we define *RMedSE* separately for each impact of private feature.

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 Impact of Private Feature* condition, 121 in the *Always High Impact of Private Feature* condition, and 119 in the *Mixed Impact of Private Feature* condition. The mean time to complete the study was 31.55 minutes (SD = 28.24), and the mean bonus payment was \$1.75 (SD = \$2.06).

4.2.1. Weight on Algorithm Results Figure 2 summarizes the results on participants' median weight on algorithm (*MedWOA*). Recall that participants in the *Always Low Impact of Private Feature* and *Always High Impact of Private Feature* conditions represent sophisticated advice weighters – humans who make predictions based on impact of private features – since, by construction, all of their predictions are based on a single exposure set. As one would expect, these sophisticated advice weighters 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 ($t(237.92) = -12.723, p < 0.0001$).

Our primary interest is studying behavior of participants in the *Mixed Impact of Private Feature* condition, as this is the most common condition in practice. As detailed in Hypothesis 1, we hypothesize that participants in this condition would be unable to sufficiently distinguish between instances with low and high impact of private features and would instead be biased towards naïve advice weighting. To evaluate Hypothesis 1, we perform two, one-sided t-tests comparing mean values of *MedWOA* across each row in Table 1. For convenience, we define $\mathcal{C}_L$, $\mathcal{C}_H$, and $\mathcal{C}_M$ to be

[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.

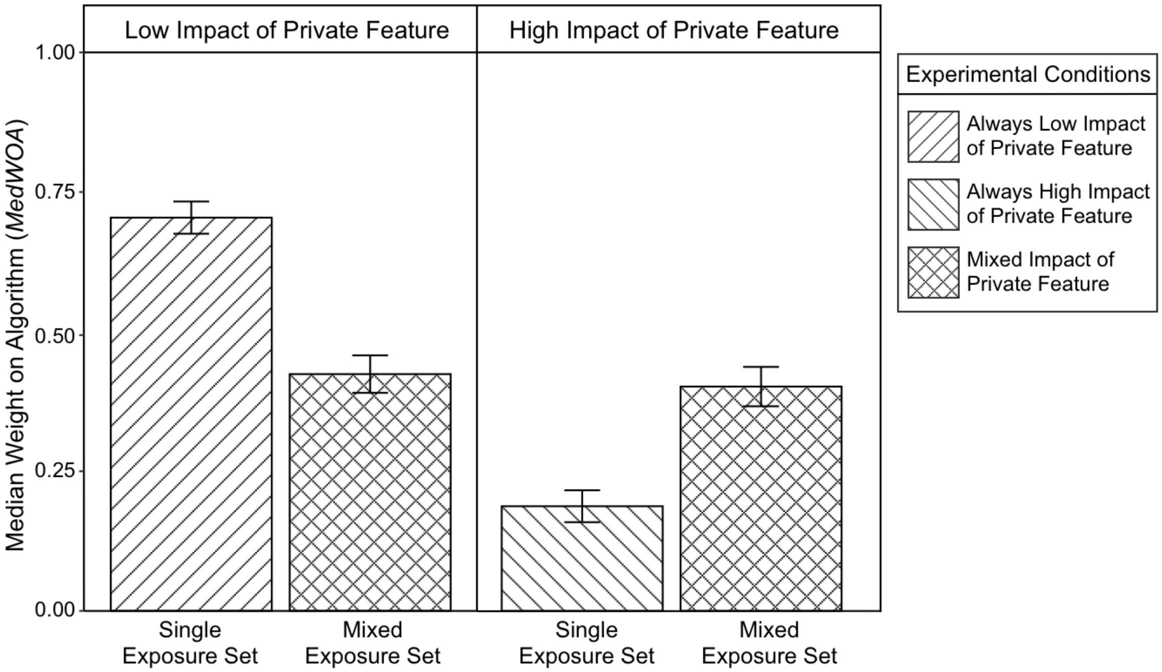


Figure 2 Median weight on algorithm results are averaged (mean) by exposure set, separately for low and high impact of private feature; standard error bars are shown.

the set of participants assigned to the *Always Low Impact of Private Feature*, *Always High Impact of Private Feature*, and *Mixed Impact of Private Feature* conditions, respectively.

For the first part of Hypothesis 1, we test whether

$$\frac{\sum_{j\in\mathcal{C}_M} MedWOA_j^L}{|\mathcal{C}_M|} \leq \frac{\sum_{j\in\mathcal{C}_L} MedWOA_j^L}{|\mathcal{C}_L|}, \tag{14}$$

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 Impact of Private Feature* condition had a significantly smaller mean *MedWOA*^{L} compared to participants in the *Always Low Impact of Private Feature* condition ($t(230.83) = -6.332, p < 0.0001$).

For the second part of Hypothesis 1, we test whether

$$\frac{\sum_{j\in\mathcal{C}_M} MedWOA_j^H}{|\mathcal{C}_M|} \geq \frac{\sum_{j\in\mathcal{C}_H} MedWOA_j^H}{|\mathcal{C}_H|}, \tag{15}$$

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 Impact of Private Feature* condition had a significantly larger mean *MedWOA*^{H} compared to participants in the *Always High Impact of Private Feature* 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.,

$$\frac{\sum_{j\in\mathcal{C}_M} MedWOA_j^L}{|\mathcal{C}_M|} - \frac{\sum_{j\in\mathcal{C}_M} MedWOA_j^H}{|\mathcal{C}_M|} \leq \frac{\sum_{j\in\mathcal{C}_L} MedWOA_j^L}{|\mathcal{C}_L|} - \frac{\sum_{j\in\mathcal{C}_H} MedWOA_j^H}{|\mathcal{C}_H|}. \tag{16}$$

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 what sophisticated advice weighters would do. 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 Impact of Private Feature* condition have larger prediction error compared to participants in the *Always Low Impact of Private Feature* 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 Impact of Private Feature* 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 ($t(235.66) = 4.733, p < 0.0001$). In fact, the average algorithm's *RMedSE* for products with a high impact of private feature is 75.263, compared to a substantially better average algorithm's *RMedSE* for products with a low impact of private feature of 4.496 (empirical results shown in Figure 3).

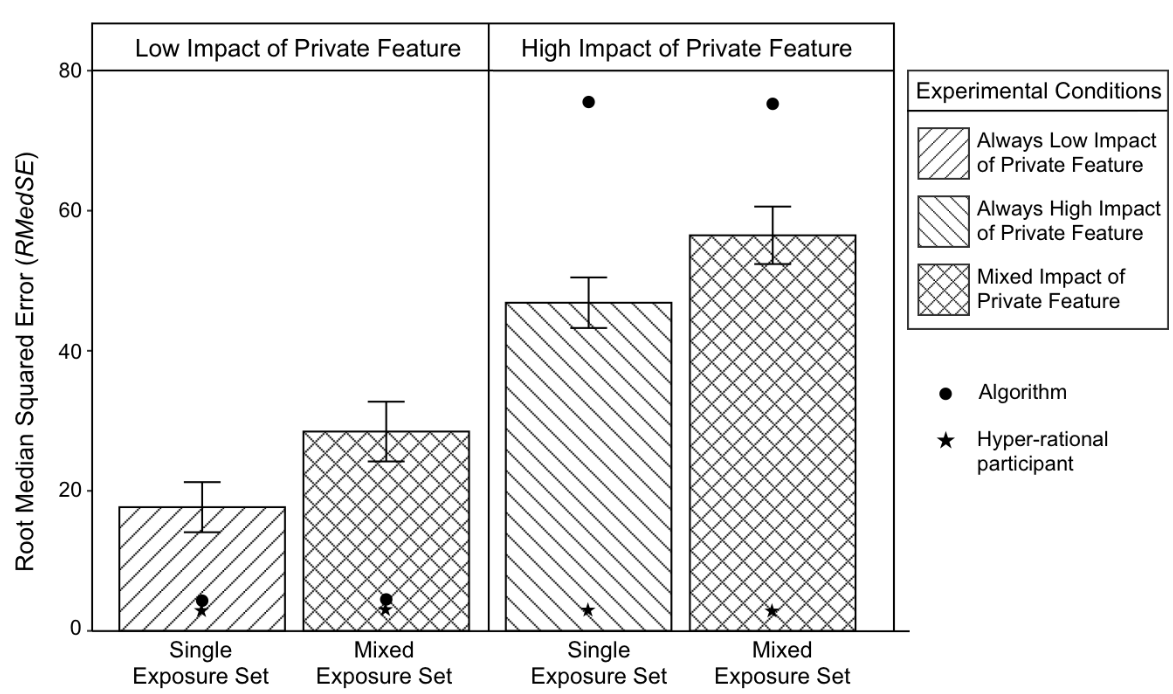


Figure 3 Root median squared error results are averaged (mean) by exposure set, separately for low and high impact of private feature; standard error bars are shown. Additionally, the mean *RMedSE* is reported for the algorithm and the hypothetical, hyper-rational participant.

Our primary interest is studying the prediction error of participants in the *Mixed Impact of Private Feature* 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 weighters for both subsets of products with low and high impact of private feature. To evaluate Hypothesis 2, 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

$$\frac{\sum_{j\in\mathcal{C}_M} RMedSE_j^L}{|\mathcal{C}_M|} \geq \frac{\sum_{j\in\mathcal{C}_L} RMedSE_j^L}{|\mathcal{C}_L|}. \tag{17}$$

As shown in the left two bars in Figure 3, participants in the *Mixed Impact of Private Feature* condition had a significantly larger mean *RMedSE*^{L} compared to participants in the *Always Low Impact of Private Feature* condition ($t(229.08) = 1.940, p = 0.0268$).

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

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

As shown in the right two bars in Figure 3, participants in the *Mixed Impact of Private Feature* condition had a significantly larger *RMedSE*^{H} compared to participants in the *Always High Impact of Private Feature* 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 weighters. This illustrates that a bias towards naïve advice weighting has a negative impact on prediction accuracy across the board.

4.2.3. Summary of Additional Analyses We reported results from all pre-registered main hypotheses and analyses above. We report several supplementary analyses in Appendix B. These include (1) repeating analyses at the task-level to show similar results, (2) conducting mediation analysis to show that the differences in *MedWOA* mediate the differences in *RMedSE*, (3) describing the performance of participants' demand predictions without the algorithm (in Step 3) to show how it compares to the algorithm, (4) 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 (5) reporting statistics on the time it took participants to complete tasks.

4.3. Discussion

Study 1 provides evidence supporting a bias towards *naïve advice weighting*: people take an overly-constant weighted average of the algorithm's predictions with their initial predictions. This bias leads to predictable patterns of over- and under-adherence to the algorithm and degrades performance when people face a mixed exposure set.

One can reasonably interpret the observed difference between the mixed exposure set and the single exposure sets as the empirical difference between naïve and sophisticated advice weighting. Of course, recall from §3.2.3 that the naïve advice weighting strategy is suboptimal not only because the weight on advice is overly constant, but also the optimal final prediction ($\mathbb{E}[Y_i]$) may not even be in the advice-weighting region (i.e., it may be outside the convex combination of the human's initial prediction and the algorithm's prediction). Indeed, in Study 1, participants' hypothetical optimal final predictions fall within their advice-weighting regions for under 55% of products; in contrast, participants' actual final predictions $\hat{y}_i^{final}$ fall within their advice-weighting regions for over 82% of products (across all treatment conditions).

What can system designers do to mitigate naïve advice weighting behavior? How can we help people better discriminate when they should adhere more or less to the algorithm? In §5, we design an intervention aimed at gaining insight into these questions.

5. Study 2: Designing Transparency to Mitigate Naïve Advice Weighting

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 (though it would be enough for a hyper-rational person to do so). How can system designers help people with bounded cognitive ability with this issue? Of course, by its very nature, system designers do not know peoples' private information. However, as detailed in Hypothesis 3, 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 (as well as its directional impact relative to the algorithm) and when they do not. Thus, we hypothesize that feature transparency will lead to improved performance accuracy for both subsets of products with low and high impact of private features, as detailed in Hypothesis 4.

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.

5.1. Design

The participant experience, data generation, and dependent variables are identical to the *Mixed Impact of Private Feature* 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 hypothetical, hyper-rational participant each have identical performance across all conditions.

1. *No Transparency.* This condition is identical to the *Mixed Impact of Private Feature* 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 subjects understood the transparency description. We also remind subjects of the transparency description when predicting demand with the algorithm for each of the 20 products in Step 5; screenshots are included in Appendix E. 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 successfully passed the comprehension check criteria by answering at least three of five questions correctly on their first try[5]. By randomly 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 (SD = 17.55), and the mean bonus payment was $1.54 (SD = 1.67).

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

[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.

$MedWOA$ 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.,

$$\frac{\sum_{j\in\mathcal{C}_{FT}} MedWOA_j^L}{|\mathcal{C}_{FT}|} - \frac{\sum_{j\in\mathcal{C}_{FT}} MedWOA_j^H}{|\mathcal{C}_{FT}|} \geq \frac{\sum_{j\in\mathcal{C}_{NT}} MedWOA_j^L}{|\mathcal{C}_{NT}|} - \frac{\sum_{j\in\mathcal{C}_{NT}} MedWOA_j^H}{|\mathcal{C}_{NT}|}. \tag{19}$$

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. Results from all regression analyses in this subsection are detailed in Table 2.

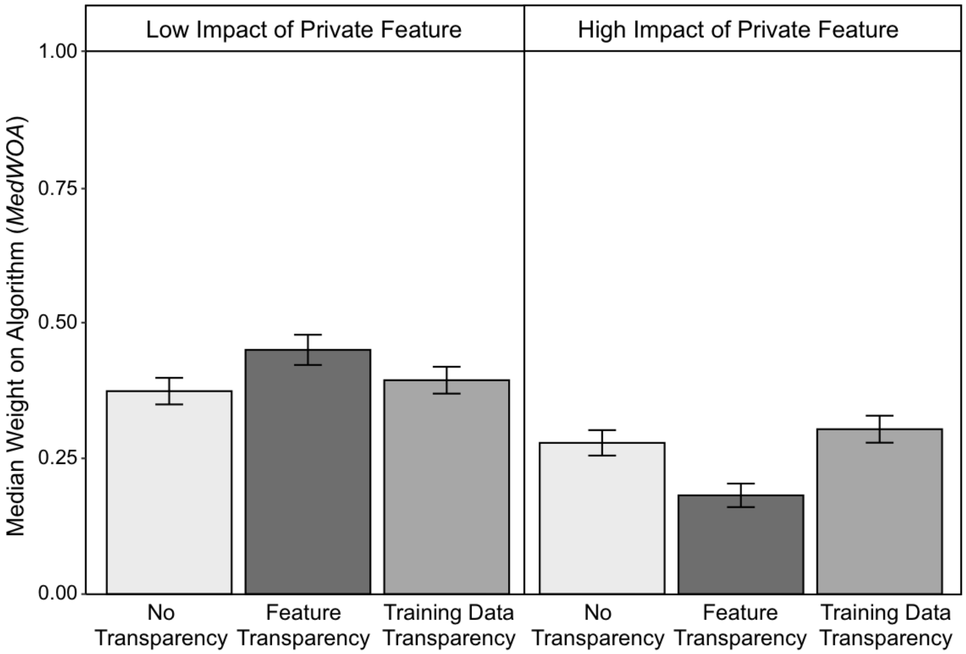


Figure 4 Median weight on algorithm results are averaged (mean) by transparency type, separately for low and high impact of private feature; standard error bars are shown.

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 ($\beta = -0.005, p = 0.871$).

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 $\lvert v_i \rvert$	0.0953***	0.0902***	0.0953***
	(0.0205)	(0.0231)	(0.0205)
Feature Transparency	-0.0967***	-0.1219***	
	(0.0319)	(0.0330)	
Low $\lvert v_i \rvert$ × Feature Transparency	0.1727***	0.1778***	
	(0.0352)	(0.0368)	
Training Data Transparency			0.0252
			(0.0342)
Low $\lvert v_i \rvert$ × 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*

Our results confirm Hypothesis 3 by showing that feature transparency mitigates naïve advice weighting behavior by helping humans recognize themselves 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 (*RMedSE*). As one would expect, within each condition, participants 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, since the algorithm provides considerably less value when the impact of private feature is high.

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. To evaluate Hypothesis 4, we first consider only

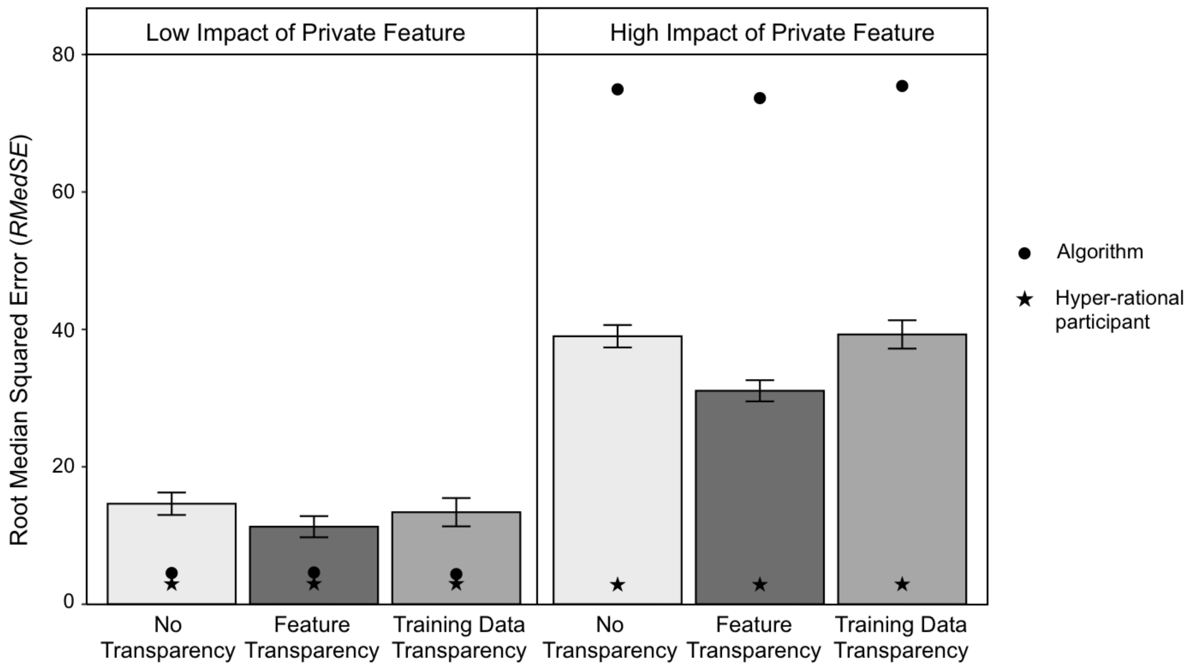


Figure 5 Root median squared error results are averaged (mean) by transparency type, separately for low and high impact of private feature; standard error bars are shown. Additionally, the mean *RMedSE* is reported for the algorithm and the hypothetical, hyper-rational participant.

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; specifically, we test whether

$$\frac{\sum_{j\in\mathcal{C}_{FT}} RMedSE_j^L}{|\mathcal{C}_{FT}|} \leq \frac{\sum_{j\in\mathcal{C}_{NT}} RMedSE_j^L}{|\mathcal{C}_{NT}|}. \tag{20}$$

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

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

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 significantly 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.

5.2.3. Summary of Additional Analyses We reported the results from all pre-registered main hypotheses and analyses above. We report on additional secondary and ex post analyses in Appendix C, summarized below.

Advice-Weighting Region Analysis Similar to Study 1, the optimal final prediction fell outside the advice-weighting region over 52% of the time across all conditions. However, amongst the decisions where the optimal final prediction fell outside the advice-weighting region, participants in *feature transparency* correctly went outside the advice-weighting region more than 2 times as frequently. See Appendix C.1 for details.

Text Analysis of Free-Response Question We included an open-ended question at the end of the study asking subjects 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”. This analysis suggests that *Feature Transparency* increases the prevalence of a strategy focusing on adjusting the algorithm’s predictions using private information. As one *Feature Transparency* participant explained: “The algorithm seemed to do a good job handling the data it had access to, so I relied it on to put me in the ballpark, then adjusted to weight [Feature] B's influence on demand." In contrast, participants without feature transparency were relatively more likely to follow a naïve advice weighting strategy. As, one *No Transparency* participant explained: "When presented with the algorithm, I made an average guess between my guess and the algorithm's number." See Appendix C.2 for details.

Supplementary DV: Absolute Percent Adjustment Error (APAE) We can directly examine participants' adjustment errors by defining a new *absolute percent adjustment error* 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*. See Appendix C.3 for details.

Time There are no significant differences between treatment conditions in the average time for initial predictions nor final predictions. See Appendix C.4 for details.

5.3. Discussion

How can system designers help humans identify and account for private information even if they don't know what the private information is? Study 2 shows that providing *feature transparency* – training subjects on what information the algorithm does use – helps to mitigate the *naïve advice weighting* behavior more effectively than other types of algorithm transparency (e.g., in this study, training data transparency) that uniformly increase subjects' adherence to the algorithm both when they should and should not do so. Our results suggest feature transparency helps mitigate both problems caused by NAW discussed in §3.2.3: participants more differentially adhere to the algorithm based on their private information, and they are more likely to make adjustments to the algorithm in the correct direction even when the optimal prediction falls outside the advice-weighting region.

6. Conclusion

This paper proposes and provides laboratory evidence that people's algorithm overrides are biased towards *naïve advice weighting*, taking a constant weighted average between what the algorithm recommends and what their own prediction would have been 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. However, providing people with *feature transparency* 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 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 as the final decision authority (e.g., for legal or ethical reasons), our results suggest that we should still let humans have override authority when they have access to substantial *unknown unknown* private information. That is, the algorithm doesn't even know which features it doesn't know. (If the algorithm knew it was missing Feature B, it could directly ask the human for Feature B's value). 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. 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*.

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*.

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.

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.

Dawes RM, Faust D, Meehl PE (1989) Clinical versus actuarial judgment. *Science*, 243(4899):1668–1674.

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.

Dietvorst BJ, Simmons JP, Massey C (2018) Overcoming algorithm aversion: People will use imperfect algorithms if they can (even slightly) modify them. *Management Science*, 64(3):1155–1170.

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.

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.

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, 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).

Kawaguchi K (2021) When will workers follow an algorithm? A field experiment with a retail business. *Management Science*, 67(3):1670–1695.

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*.

Kleinberg J, Lakkaraju H, Leskovec J, Ludwig J, Mullainathan S (2017) Human decisions and machine predictions. *The Quarterly Journal of Economics*.

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.

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*.

Shaked M, Shanthikumar JG (2007) *Stochastic orders* (Springer).

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

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 (2022) A heuristic for combining correlated experts when there is little data, working paper.

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

Sun J, Zhang DJ, Hu H, Van Mieghem JA (2022) Predicting human discretion to adjust algorithmic prescription: A large-scale field experiment in warehouse operations. *Management Science*, 68(2):846–865.

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

van Donselaar KH, Gaur V, van Woensel T, Broekmeulen RACM, Fransoo JC (2010) Ordering behavior in retail stores and implications for automated replenishment. *Management Science*, 56(5):766–784.

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

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 A: Proofs

Proof of Proposition 1: We can rewrite $NAW(\hat{\boldsymbol{y}}^{init})$ as

$$NAW(\hat{\boldsymbol{y}}^{init}): \min_{\lambda_k\in[0,1]\forall k\in\mathcal{S};\lambda_k=\lambda_{k'}\forall k,k'\in\mathcal{S}} \sum_{k\in\mathcal{S}} \big(\mathbb{E}[Y_k]-(\lambda_k\hat{y}_k^{alg}+(1-\lambda_k)\hat{y}_k^{init})\big)^2.$$

Note that this is identical to $SAW(\hat{\boldsymbol{y}}^{init})$ except it includes the additional constraint that $\lambda_k=\lambda_{k'}\forall k,k'\in\mathcal{S}$. Since $NAW(\hat{\boldsymbol{y}}^{init})$ and $SAW(\hat{\boldsymbol{y}}^{init})$ consist of identical decision variables, minimize the same objective, and the feasible region of $NAW(\hat{\boldsymbol{y}}^{init})$ is a subset of the feasible region of $SAW(\hat{\boldsymbol{y}}^{init})$, any feasible solution of $NAW(\hat{\boldsymbol{y}}^{init})$ must be a feasible solution of $SAW(\hat{\boldsymbol{y}}^{init})$. Thus, the optimal solution of $NAW(\hat{\boldsymbol{y}}^{init})$ is a feasible solution of $SAW(\hat{\boldsymbol{y}}^{init})$ with the same objective value, which gives us an upper bound on the value of $OPT^{SAW}(\hat{\boldsymbol{y}}^{init})$. □

Proof of Lemma 1: We first prove part (a). For any realization $\hat{\boldsymbol{y}}^{init}$, we can solve

$$\begin{aligned}
SAW(\hat{\boldsymbol{y}}^{init}) :&= \min_{\lambda_k\in[0,1]\forall k\in\mathcal{S}} \sum_{k\in\mathcal{S}} \big(\mathbb{E}[Y_k]-(\lambda_k\hat{y}_k^{alg}+(1-\lambda_k)\hat{y}_k^{init})\big)^2\\
&= \min_{\lambda_k\in[0,1]\forall k\in\mathcal{S}} \sum_{k\in\mathcal{S}} \big(\mathbb{E}[Y_k]-\lambda_k\hat{y}_k^{alg}-(1-\lambda_k)(\hat{y}_k^{init})\big)^2\\
&= \min_{\lambda_k\in[0,1]\forall k\in\mathcal{S}} \sum_{k\in\mathcal{S}} \big(\lambda_k\mathbb{E}[Y_k]+(1-\lambda_k)\mathbb{E}[Y_k]-\lambda_k\hat{y}_k^{alg}-(1-\lambda_k)(\hat{y}_k^{init})\big)^2\\
&= \min_{\lambda_k\in[0,1]\forall k\in\mathcal{S}} \sum_{k\in\mathcal{S}} \big(\lambda_k(\mathbb{E}[Y_k]-\hat{y}_k^{alg})+(1-\lambda_k)(\mathbb{E}[Y_k]-\hat{y}_k^{init})\big)^2\\
&= \sum_{k\in\mathcal{S}} \min_{\lambda_k\in[0,1]} \big(\lambda_k(\mathbb{E}[Y_k]-\hat{y}_k^{alg})+(1-\lambda_k)(\mathbb{E}[Y_k]-\hat{y}_k^{init})\big)^2. &(22)\\
&= \sum_{k\in\mathcal{S}} \min_{\lambda_k\in[0,1]} \big(\lambda_k(\mathbb{E}[Y_k]-\hat{y}_k^{alg})+(1-\lambda_k)z_k\big)^2 &(23)\\
&= \sum_{k\in\mathcal{S}_{L+}} \min_{\lambda_k\in[0,1]} \big(\lambda_k v_L+(1-\lambda_k)z_k\big)^2 + \sum_{k\in\mathcal{S}_{L-}} \min_{\lambda_k\in[0,1]} \big(-\lambda_k v_L+(1-\lambda_k)z_k\big)^2 &(24)\\
&\quad+ \sum_{k\in\mathcal{S}_{H+}} \min_{\lambda_k\in[0,1]} \big(\lambda_k v_H+(1-\lambda_k)z_k\big)^2 + \sum_{k\in\mathcal{S}_{H-}} \min_{\lambda_k\in[0,1]} \big(-\lambda_k v_H+(1-\lambda_k)z_k\big)^2.
\end{aligned}$$

The equality in (22) exploits the separability of $SAW(\hat{\boldsymbol{y}}^{init})$ by instance k. The equality in (23) comes from the realization of the random variables $Z_k=\mathbb{E}[Y_k]-\hat{Y}_k^{init}$ for realization $\hat{\boldsymbol{y}}^{init}$; note that we omit the explicit dependence of z_k on $\hat{\boldsymbol{y}}^{init}$ for ease of exposition. Finally, the equality in (24) follows from the values of v_k in each partition of $\mathcal{S}$.

Consider instance $k\in\mathcal{S}_{L+}$. To find $\lambda_k^{SAW}(\hat{\boldsymbol{y}}^{init})$, we simply need to solve

$$\min_{\lambda_k\in[0,1]} f(\lambda_k) = \min_{\lambda_k\in[0,1]} \big(\lambda_k v_L+(1-\lambda_k)z_k\big)^2. \tag{25}$$

Taking the derivative of $f(\lambda_k)$, we have

$$f'(\lambda_k)=2(\lambda_k v_L+(1-\lambda_k)z_k)(v_L-z_k).$$

Setting $f'(\lambda_k)=0$ and solving for λ_k gives us

$$\lambda_k=\frac{-z_k}{v_L-z_k}. \tag{26}$$

Here and elsewhere in our proofs, we choose to ignore special cases that lead to undefined terms for brevity. To verify (26) is an unconstrained minimum of $f(\lambda_k)$, we verify the convexity of $f(\lambda_k)$ by showing that the second derivative of $f(\lambda_k)$ is indeed positive:

$$f''(\lambda_k)=2(v_L-z_k)^2>0.$$

Note that if $\frac{-z_k}{v_L - z_k} \in [0,1]$, then this unconstrained solution is optimal for our constrained problem (25). Otherwise, consider the following two cases:

(i) $\frac{-z_k}{v_L - z_k} > 1$:

Since $f(\lambda_k)$ is convex, we know that it is decreasing for all values of $\lambda_k < \frac{-z_k}{v_L - z_k}$ including in $[0,1]$; thus, the minimum in (25) is achieved at $\lambda_k = 1$.

(ii) $\frac{-z_k}{v_L - z_k} < 0$:

Since $f(\lambda_k)$ is convex, we know that it is increasing for all values of $\lambda_k > \frac{-z_k}{v_L - z_k}$ including in $[0,1]$; thus, the minimum in (25) is achieved at $\lambda_k = 0$.

Thus, we can write the optimal solution to our constrained problem (25) as

$$\lambda_k^{SAW}(\hat{\boldsymbol{y}}^{init}) = \min\big(\max(\frac{-z_k}{v_L - z_k}, 0), 1\big),$$

for all $k \in \mathcal{S}_{L+}$. Since the optimal solution can be determined for any realization of random variables $\hat{Y}_k^{init}$ (and thus Z_k) for all $k \in \mathcal{S}$, we can characterize the optimal solution as a function of these random variables:

$$\Lambda_k^{SAW} = \min\big(\max(\frac{-Z_k}{v_L - Z_k}, 0), 1\big),$$

for all $k \in \mathcal{S}_{L+}$. Note that we have

$$\Lambda_k^{SAW} = \min\big(\max(\frac{-Z_k}{v_L - Z_k}, 0), 1\big) =_d \min\big(\max(\frac{-Z}{v_L - Z}, 0), 1\big) := \Lambda_{L+}^{SAW}, \tag{27}$$

for all $k \in \mathcal{S}_{L+}$ since by assumption, we have $Z_k =_d Z\ \forall k \in \mathcal{S}$. The result follows since $\min\big(\max(\frac{-Z}{v_L - Z}, 0), 1\big)$ is independent of $k\ \forall k \in \mathcal{S}_{L+}$.

Following similar steps as above for each of the other partitions of $\mathcal{S}$, we have

$$\Lambda_k^{SAW} =_d \Lambda_{L-}^{SAW} := \min\big(\max(\frac{-Z}{-v_L - Z}, 0), 1\big), \ \ \forall k \in \mathcal{S}_{L-}; \tag{28}$$

$$\Lambda_k^{SAW} =_d \Lambda_{H+}^{SAW} := \min\big(\max(\frac{-Z}{v_H - Z}, 0), 1\big), \ \ \forall k \in \mathcal{S}_{H+}; \tag{29}$$

$$\Lambda_k^{SAW} =_d \Lambda_{H-}^{SAW} := \min\big(\max(\frac{-Z}{-v_H - Z}, 0), 1\big), \ \ \forall k \in \mathcal{S}_{H-}. \tag{30}$$

This concludes the proof of part (a).

For part (b), we first prove that $\Lambda_{L+}^{SAW} \succcurlyeq_S \Lambda_{H+}^{SAW}$. To do so, we aim to show that

$$\min\big(\max(\frac{-z}{v_L - z}, 0), 1\big) \geq \min\big(\max(\frac{-z}{v_H - z}, 0), 1\big) \tag{31}$$

for any realization z of Z, in which case the stochastic dominance result $\Lambda_{L+}^{SAW} \succcurlyeq_S \Lambda_{H+}^{SAW}$ follows.

We will show that (31) holds in each of three exhaustive cases characterizing the value of $\frac{-z}{v_L - z}$.

(i) $\frac{-z}{v_L - z} \geq 1$:

In this case, $\min\big(\max(\frac{-z}{v_L - z}, 0), 1\big) = 1$, which is an upper bound for $\min\big(\max(\frac{-z}{v_H - z}, 0), 1\big)$, giving us the result.

(ii) $\frac{-z}{v_L - z} \leq 0$:

In this case, $\min\big(\max(\frac{-z}{v_L-z}, 0), 1\big) = 0$, so we must show $\min\big(\max(\frac{-z}{v_H-z}, 0), 1\big) = 0$, or equivalently, we must show that $\frac{-z}{v_H-z} \leq 0$. Consider the following two subcases which could make $\frac{-z}{v_L-z} \leq 0$:

- The numerator is positive and the denominator is negative, i.e., $z \leq 0$ and $v_L < z$. This gives us $v_L < z \leq 0$, which is impossible since, by definition, $v_L \geq 0$.
- The numerator is negative and the denominator is positive, i.e., $z \geq 0$ and $v_L > z$. This gives us $0 \leq z < v_L < v_H$; therefore $\frac{-z}{v_H-z} \leq 0$ since its numerator is negative and denominator is positive.

(iii) $0 < \frac{-z}{v_L-z} < 1$:

Consider the following two subcases which could make $\frac{-z}{v_L-z} > 0$:

- The numerator and denominator are both negative, i.e., $z > 0$ and $v_L < z$. Since by definition we have $v_L \geq 0$, this gives us $0 \leq v_L < z$. Note that $0 > v_L - z \geq 0 - z = -z$ and thus $\frac{-z}{v_L-z} \geq 1$, which violates the requirements of this subcase.
- The numerator and denominator are both positive, i.e., $v_L \geq 0 > z$. Consider the function $f(x) = \frac{-z}{x-z}$. If we can show that $f(x)$ is decreasing, then we could conclude $f(v_L) = \frac{-z}{v_L-z} \geq \frac{-z}{v_H-z} = f(v_H)$ since $v_H > v_L$. We indeed have $f'(x) = \frac{z}{(x-z)^2} < 0$, since $z < 0$ in this subcase.

We have shown that (31) holds for any realization z; thus, $\Lambda^{SAW}_{L+} \succcurlyeq_S \Lambda^{SAW}_{H+}$ follows.

Similarly, we next prove that $\Lambda^{SAW}_{L-} \succcurlyeq_S \Lambda^{SAW}_{H-}$. To do so, we aim to show that

$$\min\big(\max(\frac{-z}{-v_L - z}, 0), 1\big) \geq \min\big(\max(\frac{-z}{-v_H - z}, 0), 1\big) \tag{32}$$

for any realization z of Z, in which case the stochastic dominance result $\Lambda^{SAW}_{L-} \succcurlyeq_S \Lambda^{SAW}_{H-}$ follows.

We will show that (32) holds in each of three exhaustive cases characterizing the value of $\frac{-z}{-v_L-z}$.

(i) $\frac{-z}{-v_L-z} \geq 1$:

In this case, $\min\big(\max(\frac{-z}{-v_L-z}, 0), 1\big) = 1$, which is an upper bound for $\min\big(\max(\frac{-z}{-v_H-z}, 0), 1\big)$, giving us the result.

(ii) $\frac{-z}{-v_L-z} \leq 0$:

In this case, $\min\big(\max(\frac{-z}{-v_L-z}, 0), 1\big) = 0$, so we must show $\min\big(\max(\frac{-z}{-v_H-z}, 0), 1\big) = 0$, or equivalently, we must show that $\frac{-z}{-v_H-z} \leq 0$. Consider the following two subcases which could make $\frac{-z}{-v_L-z} \leq 0$:

- The numerator is positive and the denominator is negative, i.e., $-v_L < z \leq 0$. Since $v_H > v_L$, we have $-v_H - z < -v_L - z < 0$; therefore $\frac{-z}{-v_H-z} \leq 0$ since its numerator is positive and denominator is negative.
- The numerator is negative and the denominator is positive, i.e., $z \geq 0$ and $-v_L > z$. This gives us $-v_L > z \geq 0$, which is impossible since, by definition, $-v_L \leq 0$.

(iii) $0 < \frac{-z}{-v_L-z} < 1$:

Consider the following two subcases which could make $\frac{-z}{-v_L-z} > 0$:

- The numerator and denominator are both negative, i.e., $-v_L \leq 0 < z$. Consider the function $f(x) = \frac{-z}{x-z}$. If we can show that $f(x)$ is increasing, then we could conclude $f(-v_L) = \frac{-z}{-v_L-z} \geq \frac{-z}{-v_H-z} = f(-v_H)$ since $-v_H < -v_L$. We indeed have $f'(x) = \frac{z}{(x-z)^2} > 0$, since $z > 0$ in this subcase.

- The numerator and denominator are both positive, i.e., $z < -v_L \leq 0$. Note that this implies $0 < -v_L - z \leq -z$ and thus $\frac{-z}{-v_L - z} \geq 1$, which violates the requirements of this subcase.

We have shown that (32) holds for any realization z; thus, $\Lambda_{L^-}^{SAW} \not\succeq_S \Lambda_{H^-}^{SAW}$ follows.□

Proof of Proposition 2: The proof of the stochastic inequality in (a) follows directly from Lemma 1(b) and the closure properties of first order stochastic dominance (see Theorem 1.A.3 in Shaked and Shanthikumar 2007). To show the limiting behaviors for (b), first consider the characterization of $\Lambda_k^{SAW}, \forall k \in \{\mathcal{S}_{L^-}, \mathcal{S}_{H^-}\}$ from the proof of Lemma 1, where the impact of private features for each instance k is $-v_k$ such that $v_k \geq 0$:

$$\Lambda_k^{SAW} = \min\big(\max(\frac{-Z_k}{-v_k - Z_k}, 0), 1\big), \ \ \forall k \in \{\mathcal{S}_{L^-}, \mathcal{S}_{H^-}\}$$

This can be rewritten as:

$$\forall k \in \{\mathcal{S}_{L^-}, \mathcal{S}_{H^-}\}, \ \ \Lambda_k^{SAW} = \begin{cases} \frac{Z_k}{v_k + Z_k} & Z_k > 0 \\ 0 & -v_k \leq Z_k \leq 0 \\ 1 & Z_k < -v_k \end{cases}$$

Then, $\forall k \in \{\mathcal{S}_{L^-}, \mathcal{S}_{H^-}\}$:

$$\mathbb{E}[\Lambda_k^{SAW}] = \int_{-\infty}^{-v_k} 1 * g(z_k)\, dz_k + \int_{-v_k}^{0} 0 * g(z_k)\, dz_k + \int_{0}^{+\infty} \frac{z_k}{v_k + z_k} * g(z_k)\, dz_k$$

where $g(\cdot)$ is the probability density function of Z_k. This can be re-written as:

$$\mathbb{E}[\Lambda_k^{SAW}] = \int_{-\infty}^{+\infty} [\mathbb{1}_{\{z_k < -v_k\}} + \mathbb{1}_{\{z_k \geq 0\}}(\frac{z_k}{v_k + z_k})] * g(z_k)\, dz_k$$

Then the $\lim_{v_k \to 0} \mathbb{E}[\Lambda_k^{SAW}]$ is:

$$\begin{aligned} \lim_{v_k \to 0} \mathbb{E}[\Lambda_k^{SAW}] &= \int_{-\infty}^{+\infty} [\mathbb{1}_{\{z_k < 0\}} + \mathbb{1}_{\{z_k \geq 0\}}] * g(z_k)\, dz_k \\ &= \int_{-\infty}^{+\infty} g(z_k)\, dz_k \\ &= 1 \end{aligned}$$

Similarly, the $\lim_{v_k \to \infty} \mathbb{E}[\Lambda_k^{SAW}]$ is:

$$\begin{aligned} \lim_{v_k \to \infty} \mathbb{E}[\Lambda_k^{SAW}] &= \int_{-\infty}^{+\infty} [\mathbb{1}_{\{z_k < -\infty\}} + \mathbb{1}_{\{z_k \geq 0\}}(0)] * g(z_k)\, dz_k \\ &= \int_{-\infty}^{+\infty} 0 * g(z_k)\, dz_k \\ &= 0 \end{aligned}$$

We next characterize $\Lambda_k^{SAW}, \forall k \in \{\mathcal{S}_{L^+}, \mathcal{S}_{H^+}\}$, where the impact of private features for each instance k is $v_k \geq 0$:

$$\Lambda_k^{SAW} = \min\big(\max(\frac{-Z_k}{v_k - Z_k}, 0), 1\big), \ \ \forall k \in \{\mathcal{S}_{L^+}, \mathcal{S}_{H^+}\}$$

This can be rewritten as:

$$\forall k \in \{\mathcal{S}_{L^+}, \mathcal{S}_{H^+}\}, \ \ \Lambda_k^{SAW} = \begin{cases} 1 & Z_k > v_k \\ 0 & 0 \leq Z_k \leq v_k \\ \frac{Z_k}{Z_k - v_k} & Z_k < 0 \end{cases}$$

Then, $\forall k \in \{\mathcal{S}_{L^+}, \mathcal{S}_{H^+}\}$:

$$\mathbb{E}[\Lambda_k^{SAW}] = \int_{-\infty}^{0} \frac{z_k}{z_k - v_k} * g(z_k)\, dz_k + \int_{0}^{v_k} 0 * g(z_k)\, dz_k + \int_{v_k}^{+\infty} 1 * g(z_k)\, dz_k$$

This can be re-written as:

$$\mathbb{E}[\Lambda_k^{SAW}] = \int_{-\infty}^{+\infty} [\mathbb{1}_{\{z_k > v_k\}} + \mathbb{1}_{\{z_k \le 0\}}(\frac{z_k}{z_k - v_k})] * g(z_k)\, dz_k$$

Then the $\lim_{v_k \to 0} \mathbb{E}[\Lambda_k^{SAW}]$ is:

$$\begin{aligned}\lim_{v_k \to 0} \mathbb{E}[\Lambda_k^{SAW}] &= \int_{-\infty}^{+\infty} [\mathbb{1}_{\{z_k > 0\}} + \mathbb{1}_{\{z_k \le 0\}}] * g(z_k)\, dz_k \\ &= \int_{-\infty}^{+\infty} g(z_k)\, dz_k \\ &= 1\end{aligned}$$

Similarly, the $\lim_{v_k \to \infty} \mathbb{E}[\Lambda_k^{SAW}]$ is:

$$\begin{aligned}\lim_{v_k \to \infty} \mathbb{E}[\Lambda_k^{SAW}] &= \int_{-\infty}^{+\infty} [\mathbb{1}_{\{z_k > \infty\}} + \mathbb{1}_{\{z_k \le 0\}}(0)] * g(z_k)\, dz_k \\ &= \int_{-\infty}^{+\infty} 0 * g(z_k)\, dz_k \\ &= 0\end{aligned}$$

Now, $\lim_{v_k \to 0} \mathbb{E}[\Lambda_k^{SAW}] = 1$ for every $k \in \mathcal{S}$. Similarly, $\lim_{v_k \to \infty} \mathbb{E}[\Lambda_k^{SAW}] = 0$ for every $k \in \mathcal{S}$. Thus, following from the linearity of expected values, $\lim_{v_L \to 0} \mathbb{E}[\Lambda_L^{SAW}] = 1$ and $\lim_{v_H \to \infty} \mathbb{E}[\Lambda_H^{SAW}] = 0$. □

Proof of Proposition 3: The following is an outline of our proof:

Step 1: Find expressions for $\mathbb{E}[\Lambda_L^{SAW}], \mathbb{E}[\Lambda_H^{SAW}]$, and $\mathbb{E}[\Lambda^{NAW}]$ by enumerating the possible realizations of Z for each instance.

Step 2: Prove $\mathbb{E}[\Lambda_L^{SAW}] > \mathbb{E}[\Lambda^{NAW}]$.

Step 3: Prove $\mathbb{E}[\Lambda^{NAW}] > \mathbb{E}[\Lambda_H^{SAW}]$.

Step 1

We first find an expression for $\mathbb{E}[\Lambda_L^{SAW}]$. We can write

$$\begin{aligned}\mathbb{E}[\Lambda_L^{SAW}] &= \frac{1}{2}\Big(\mathbb{E}[\Lambda_{L^+}^{SAW}] + \mathbb{E}[\Lambda_{L^-}^{SAW}]\Big) \\ &= \frac{1}{2}\Big(\mathbb{E}[\min\big(\max(\frac{-Z}{v_L - Z}, 0), 1\big)] + \mathbb{E}[\min\big(\max(\frac{-Z}{-v_L - Z}, 0), 1\big)]\Big),\end{aligned}$$

where the second equality is from (27) and (28). We can calculate each expectation separately by enumerating the two possible values of Z: c and $-c$.

$$\mathbb{E}[\min\big(\max(\frac{-Z}{v_L - Z}, 0), 1\big)] = \frac{1}{2}\min\big(\max(\frac{-c}{v_L - c}, 0), 1\big) + \frac{1}{2}\min\big(\max(\frac{c}{v_L + c}, 0), 1\big) = \frac{1}{2}\big(1 + \frac{c}{v_L + c}\big).$$

$$\mathbb{E}[\min\big(\max(\frac{-Z}{-v_L - Z}, 0), 1\big)] = \frac{1}{2}\min\big(\max(\frac{-c}{-v_L - c}, 0), 1\big) + \frac{1}{2}\min\big(\max(\frac{c}{-v_L + c}, 0), 1\big) = \frac{1}{2}\big(\frac{-c}{-v_L - c} + 1\big).$$

Note that these two expectations are identical, and thus we have

$$\mathbb{E}[\Lambda_L^{SAW}] = \frac{1}{2}\big(1 + \frac{c}{v_L + c}\big) = \frac{v_L + 2c}{2(v_L + c)}. \tag{33}$$

We next follow similar steps to find an expression for $\mathbb{E}[\Lambda_H^{SAW}]$. We can write

$$\begin{aligned}\mathbb{E}[\Lambda_H^{SAW}] &= \frac{1}{2}\Big(\mathbb{E}[\Lambda_{H+}^{SAW}] + \mathbb{E}[\Lambda_{H-}^{SAW}]\Big)\\ &= \frac{1}{2}\Big(\mathbb{E}[\min\big(\max(\frac{-Z}{v_H - Z}, 0), 1\big)] + \mathbb{E}[\min\big(\max(\frac{-Z}{-v_H - Z}, 0), 1\big)]\Big),\end{aligned}$$

where the second equality is from (29) and (30). We can calculate each expectation separately by enumerating the two possible values of Z: c and $-c$.

$$\mathbb{E}[\min\big(\max(\frac{-Z}{v_H - Z}, 0), 1\big)] = \frac{1}{2}\min\big(\max(\frac{-c}{v_H - c}, 0), 1\big) + \frac{1}{2}\min\big(\max(\frac{c}{v_H + c}, 0), 1\big) = \frac{1}{2}\big(0 + \frac{c}{v_H + c}\big).$$
$$\mathbb{E}[\min\big(\max(\frac{-Z}{-v_H - Z}, 0), 1\big)] = \frac{1}{2}\min\big(\max(\frac{-c}{-v_H - c}, 0), 1\big) + \frac{1}{2}\min\big(\max(\frac{c}{-v_H + c}, 0), 1\big) = \frac{1}{2}\big(\frac{-c}{-v_H - c} + 0\big).$$

Note that these two expectations are identical, and thus we have

$$\mathbb{E}[\Lambda_H^{SAW}] = \frac{1}{2}\big(\frac{c}{v_H + c}\big) = \frac{c}{2(v_H + c)}. \tag{34}$$

We next aim to find an expression for $\mathbb{E}[\Lambda^{NAW}]$. To do so, we must first find the optimal solution $\lambda^{NAW}(\hat{\boldsymbol{y}}^{init})$ for any realization $\hat{\boldsymbol{y}}^{init}$ (or equivalently, any vector of realizations $Z_k = z_k \ \forall k$, where we will use k to denote the set for which the instance belongs). We can solve

$$\min_{\lambda \in [0,1]} f(\lambda) = \big(\lambda v_L + (1-\lambda) z_{L+}\big)^2 + \big(-\lambda v_L + (1-\lambda) z_{L-}\big)^2 + \big(\lambda v_H + (1-\lambda) z_{H+}\big)^2 + \big(-\lambda v_H + (1-\lambda) z_{H-}\big)^2.$$

Note that this follows from (24) with the extra constraint that $\lambda_k = \lambda \ \forall k$, and recognizing that we have a cardinality of one for each partition in the special case that we consider. Taking the derivative of $f(\lambda)$, we have

$$\begin{aligned}f'(\lambda) = {} & 2(v_L - z_{L+})z_{L+} + 2\lambda(v_L - z_{L+})^2 + 2(-v_L - z_{L-})z_{L-} + 2\lambda(-v_L - z_{L-})^2\\ & + 2(v_H - z_{H+})z_{H+} + 2\lambda(v_H - z_{H+})^2 + 2(-v_H - z_{H-})z_{H-} + 2\lambda(-v_H - z_{H-})^2.\end{aligned}$$

Setting $f'(\lambda) = 0$ and solving for λ gives us

$$\lambda = \frac{-z_{L+}(v_L - z_{L+}) - z_{L-}(-v_L - z_{L-}) - z_{H+}(v_H - z_{H+}) - z_{H-}(-v_H - z_{H-})}{(v_L - z_{L+})^2 + (-v_L - z_{L-})^2 + (v_H - z_{H+})^2 + (-v_H - z_{H-})^2}.$$

To verify this is an unconstrained minimum of $f(\lambda)$, we verify the convexity of $f(\lambda)$:

$$f''(\lambda) = 2(v_L - z_{L+})^2 + 2(-v_L - z_{L-})^2 + 2(v_H - z_{H+})^2 + 2(-v_H - z_{H-})^2 \geq 0.$$

Following the same convexity argument as in the proof of Lemma 1, we can write the optimal solution to our constrained problem as

$$\lambda^{NAW}(\hat{\boldsymbol{y}}^{init}) = \min\big(\max(\frac{-z_{L+}(v_L - z_{L+}) - z_{L-}(-v_L - z_{L-}) - z_{H+}(v_H - z_{H+}) - z_{H-}(-v_H - z_{H-})}{(v_L - z_{L+})^2 + (-v_L - z_{L-})^2 + (v_H - z_{H+})^2 + (-v_H - z_{H-})^2}, 0), 1\big).$$

Since the optimal solution can be determined for any realization of random variables $\hat{Y}_k^{init}$ (and thus Z_k) for all $k \in \mathcal{S}$, we can characterize the optimal solution as a function of these random variables:

$$\Lambda^{NAW} = \min\big(\max(\frac{-Z_{L^+}(v_L - Z_{L^+}) - Z_{L^-}(-v_L - Z_{L^-}) - Z_{H^+}(v_H - Z_{H^+}) - Z_{H^-}(-v_H - Z_{H^-})}{(v_L - Z_{L^+})^2 + (-v_L - Z_{L^-})^2 + (v_H - Z_{H^+})^2 + (-v_H - Z_{H^-})^2}, 0), 1\big). \tag{35}$$

Now that we have an expression for Λ^{NAW}, we next calculate $\mathbb{E}[\Lambda^{NAW}]$ for our special case. Since $Z_{L^+}, Z_{L^-}, Z_{H^+}, Z_{H^-} =_d Z$ which can take two possible values, c and $-c$, we can enumerate $2^4 = 16$ possible sets of realizations of Z_k $\forall k$. In what follows, we will calculate $\lambda^{NAW}(\hat{\boldsymbol{y}}^{init})$, λ^{NAW} for short, for each of the 16 possible sets of realizations; then we can weight each one with probability $\frac{1}{16}$ and sum to get $\mathbb{E}[\Lambda^{NAW}]$.

(i) $z_{L^+} = -c, z_{L^-} = -c, z_{H^+} = -c, z_{H^-} = -c$:

The unconstrained solution is

$$\lambda = \frac{c(v_L + c) + c(-v_L + c) + c(v_H + c) + c(-v_H + c)}{(v_L + c)^2 + (-v_L + c)^2 + (v_H + c)^2 + (-v_H + c)^2} = \frac{2c^2}{v_L^2 + v_H^2 + 2c^2}.$$

Since $\lambda \in (0, 1)$, we have

$$\lambda^{NAW} = \frac{2c^2}{v_L^2 + v_H^2 + 2c^2}. \tag{36}$$

(ii) $z_{L^+} = c, z_{L^-} = c, z_{H^+} = c, z_{H^-} = c$:

The unconstrained solution is

$$\lambda = \frac{-c(v_L - c) - c(-v_L - c) - c(v_H - c) - c(-v_H - c)}{(v_L - c)^2 + (-v_L - c)^2 + (v_H - c)^2 + (-v_H - c)^2}.$$

Note that this is equivalent to the expression for λ in case (i), and thus λ^{NAW} is defined in (36).

(iii) $z_{L^+} = -c, z_{L^-} = -c, z_{H^+} = c, z_{H^-} = c$:

The unconstrained solution is

$$\lambda = \frac{c(v_L + c) + c(-v_L + c) - c(v_H - c) - c(-v_H - c)}{(v_L + c)^2 + (-v_L + c)^2 + (v_H - c)^2 + (-v_H - c)^2}.$$

Note that this is equivalent to the expression for λ in case (i), and thus λ^{NAW} is defined in (36).

(iv) $z_{L^+} = c, z_{L^-} = c, z_{H^+} = -c, z_{H^-} = -c$:

The unconstrained solution is

$$\lambda = \frac{-c(v_L - c) - c(-v_L - c) + c(v_H + c) + c(-v_H + c)}{(v_L - c)^2 + (-v_L - c)^2 + (v_H + c)^2 + (-v_H + c)^2}.$$

Note that this is equivalent to the expression for λ in case (i), and thus λ^{NAW} is defined in (36).

(v) $z_{L^+} = -c, z_{L^-} = c, z_{H^+} = -c, z_{H^-} = c$:

The unconstrained solution is

$$\lambda = \frac{c(v_L + c) - c(-v_L - c) + c(v_H + c) - c(-v_H - c)}{(v_L + c)^2 + (-v_L - c)^2 + (v_H + c)^2 + (-v_H - c)^2} = \frac{c(v_L + c) + c(v_H + c)}{(v_L + c)^2 + (v_H + c)^2}.$$

Since $\lambda \in (0, 1)$, we have

$$\lambda^{NAW} = \frac{c(v_L + c) + c(v_H + c)}{(v_L + c)^2 + (v_H + c)^2}. \tag{37}$$

(vi) $z_{L^+} = -c, z_{L^-} = c, z_{H^+} = c, z_{H^-} = -c$:

The unconstrained solution is

$$\lambda = \frac{c(v_L + c) - c(-v_L - c) - c(v_H - c) + c(-v_H + c)}{(v_L + c)^2 + (-v_L - c)^2 + (v_H - c)^2 + (-v_H + c)^2} = \frac{c(v_L + c) + c(-v_H + c)}{(v_L + c)^2 + (v_H - c)^2}.$$

Since we have $v_L \le c \le v_H \le 2v_L$, $\lambda \in (0,1)$ for all values of c, and we have

$$\lambda^{NAW} = \frac{c(v_L + c) + c(-v_H + c)}{(v_L + c)^2 + (v_H - c)^2}. \tag{38}$$

(vii) $z_{L^+} = c, z_{L^-} = -c, z_{H^+} = -c, z_{H^-} = c$:

The unconstrained solution is

$$\lambda = \frac{-c(v_L - c) + c(-v_L + c) + c(v_H + c) - c(-v_H - c)}{(v_L - c)^2 + (-v_L + c)^2 + (v_H + c)^2 + (-v_H - c)^2} = \frac{c(-v_L + c) + c(v_H + c)}{(v_L - c)^2 + (v_H + c)^2}.$$

Since we have $v_L \le c \le v_H \le 2v_L$, $\lambda \in (0,1)$ for all values of c, and we have

$$\lambda^{NAW} = \frac{c(-v_L + c) + c(v_H + c)}{(v_L - c)^2 + (v_H + c)^2}. \tag{39}$$

(viii) $z_{L^+} = c, z_{L^-} = -c, z_{H^+} = c, z_{H^-} = -c$:

The unconstrained solution is

$$\lambda = \frac{-c(v_L - c) + c(-v_L + c) - c(v_H - c) + c(-v_H + c)}{(v_L - c)^2 + (-v_L + c)^2 + (v_H - c)^2 + (-v_H + c)^2} = \frac{c(-v_L + c) + c(-v_H + c)}{(v_L - c)^2 + (v_H - c)^2}.$$

Note that $\lambda \le 0$ when $c \le \frac{v_L + v_H}{2}$, and $\lambda \ge 1$ when $c \ge \frac{v_L^2 + v_H^2}{v_L + v_H}$. Thus we have three possible values of λ^{NAW}, depending on the value of c:

$$\lambda^{NAW} = \begin{cases} 0; & c \le \frac{v_L + v_H}{2} \\ \frac{c(-v_L + c) + c(-v_H + c)}{(v_L - c)^2 + (v_H - c)^2}; & \frac{v_L + v_H}{2} < c < \frac{v_L^2 + v_H^2}{v_L + v_H} \\ 1; & c \ge \frac{v_L^2 + v_H^2}{v_L + v_H} \end{cases}. \tag{40}$$

(ix) $z_{L^+} = -c, z_{L^-} = c, z_{H^+} = c, z_{H^-} = c$:

The unconstrained solution is

$$\lambda = \frac{c(v_L + c) - c(-v_L - c) - c(v_H - c) - c(-v_H - c)}{(v_L + c)^2 + (-v_L - c)^2 + (v_H - c)^2 + (-v_H - c)^2} = \frac{4c^2 + 2cv_L}{2(v_L + c)^2 + (v_H - c)^2 + (v_H + c)^2}.$$

Since $\lambda \in (0,1)$, we have

$$\lambda^{NAW} = \frac{4c^2 + 2cv_L}{2(v_L + c)^2 + (v_H - c)^2 + (v_H + c)^2}. \tag{41}$$

(x) $z_{L^+} = -c, z_{L^-} = c, z_{H^+} = -c, z_{H^-} = -c$:

The unconstrained solution is

$$\lambda = \frac{c(v_L + c) - c(-v_L - c) + c(v_H + c) + c(-v_H + c)}{(v_L + c)^2 + (-v_L - c)^2 + (v_H + c)^2 + (-v_H + c)^2}.$$

Note that this is equivalent to the expression for λ in case (ix), and thus λ^{NAW} is defined in (41).

(xi) $z_{L+} = c, z_{L-} = -c, z_{H+} = c, z_{H-} = c$:

The unconstrained solution is

$$\lambda = \frac{-c(v_L - c) + c(-v_L + c) - c(v_H - c) - c(-v_H - c)}{(v_L - c)^2 + (-v_L + c)^2 + (v_H - c)^2 + (-v_H - c)^2} = \frac{4c^2 - 2cv_L}{2(v_L - c)^2 + (v_H - c)^2 + (v_H + c)^2}.$$

Since we have $v_L \leq c \leq v_H$, $\lambda \in (0,1)$ for all values of c, and we have

$$\lambda^{NAW} = \frac{4c^2 - 2cv_L}{2(v_L - c)^2 + (v_H - c)^2 + (v_H + c)^2}. \tag{42}$$

(xii) $z_{L+} = c, z_{L-} = -c, z_{H+} = -c, z_{H-} = -c$:

The unconstrained solution is

$$\lambda = \frac{-c(v_L - c) + c(-v_L + c) + c(v_H + c) + c(-v_H + c)}{(v_L - c)^2 + (-v_L + c)^2 + (v_H + c)^2 + (-v_H + c)^2}.$$

Note that this is equivalent to the expression for λ in case (xi), and thus λ^{NAW} is defined in (42).

(xiii) $z_{L+} = c, z_{L-} = c, z_{H+} = -c, z_{H-} = c$:

The unconstrained solution is

$$\lambda = \frac{-c(v_L - c) - c(-v_L - c) + c(v_H + c) - c(-v_H - c)}{(v_L - c)^2 + (-v_L - c)^2 + (v_H + c)^2 + (-v_H - c)^2} = \frac{4c^2 + 2cv_H}{(v_L - c)^2 + (v_L + c)^2 + 2(v_H + c)^2}.$$

Since $\lambda \in (0,1)$, we have

$$\lambda^{NAW} = \frac{4c^2 + 2cv_H}{(v_L - c)^2 + (v_L + c)^2 + 2(v_H + c)^2}. \tag{43}$$

(xiv) $z_{L+} = -c, z_{L-} = -c, z_{H+} = -c, z_{H-} = c$:

The unconstrained solution is

$$\lambda = \frac{c(v_L + c) + c(-v_L + c) + c(v_H + c) - c(-v_H - c)}{(v_L + c)^2 + (-v_L + c)^2 + (v_H + c)^2 + (-v_H - c)^2}.$$

Note that this is equivalent to the expression for λ in case (xiii), and thus λ^{NAW} is defined in (43).

(xv) $z_{L+} = c, z_{L-} = c, z_{H+} = c, z_{H-} = -c$:

The unconstrained solution is

$$\lambda = \frac{-c(v_L - c) - c(-v_L - c) - c(v_H - c) + c(-v_H + c)}{(v_L - c)^2 + (-v_L - c)^2 + (v_H - c)^2 + (-v_H + c)^2} = \frac{4c^2 - 2cv_H}{(v_L - c)^2 + (v_L + c)^2 + 2(v_H - c)^2}.$$

Since we have $v_L \leq c \leq v_H \leq 2v_L$, $\lambda \in (0,1)$ for all values of c, and we have

$$\lambda^{NAW} = \frac{4c^2 - 2cv_H}{(v_L - c)^2 + (v_L + c)^2 + 2(v_H - c)^2}. \tag{44}$$

(xvi) $z_{L+} = -c, z_{L-} = -c, z_{H+} = c, z_{H-} = -c$:

The unconstrained solution is

$$\lambda = \frac{c(v_L + c) + c(-v_L + c) - c(v_H - c) + c(-v_H + c)}{(v_L + c)^2 + (-v_L + c)^2 + (v_H - c)^2 + (-v_H + c)^2}.$$

Note that this is equivalent to the expression for λ in case (xv), and thus λ^{NAW} is defined in (44).

Note that for each case other than case (viii), the expression for λ^{NAW} is identical for all values of c such that $v_L \leq c \leq v_H \leq 2v_L$. To simplify the notation for the following expression of $\mathbb{E}[\Lambda^{NAW}]$, we denote $\lambda^{NAW}_{(viii)}(c)$ as the expression for λ^{NAW} in case (viii) which depends on the value of c. We can write

$$\mathbb{E}[\Lambda^{NAW}] = \frac{1}{4}\Big(\frac{2c^2}{v_L^2+v_H^2+2c^2}\Big) \tag{45}$$

$$+\frac{1}{16}\Big(\frac{c(v_L+c)+c(v_H+c)}{(v_L+c)^2+(v_H+c)^2}\Big)+\frac{1}{16}\Big(\frac{c(v_L+c)+c(-v_H+c)}{(v_L+c)^2+(v_H-c)^2}\Big)+\frac{1}{16}\Big(\frac{c(-v_L+c)+c(v_H+c)}{(v_L-c)^2+(v_H+c)^2}\Big)+\frac{1}{16}\lambda^{NAW}_{(viii)}(c) \tag{46}$$

$$+\frac{1}{8}\Big(\frac{4c^2+2cv_L}{2(v_L+c)^2+(v_H-c)^2+(v_H+c)^2}\Big)+\frac{1}{8}\Big(\frac{4c^2-2cv_L}{2(v_L-c)^2+(v_H-c)^2+(v_H+c)^2}\Big) \tag{47}$$

$$+\frac{1}{8}\Big(\frac{4c^2+2cv_H}{(v_L-c)^2+(v_L+c)^2+2(v_H+c)^2}\Big)+\frac{1}{8}\Big(\frac{4c^2-2cv_H}{(v_L-c)^2+(v_L+c)^2+2(v_H-c)^2}\Big). \tag{48}$$

The term in (45) comes from cases (i)-(iv); the terms in (46) come from cases (v)-(viii); the terms in (47) come from cases (ix)-(xii); and the terms in (48) come from cases (xiii)-(xvi).

Step 2

To prove $\mathbb{E}[\Lambda_L^{SAW}] > \mathbb{E}[\Lambda^{NAW}]$, we will first show that $\mathbb{E}[\Lambda_L^{SAW}] \geq \frac{3}{4}$. Then we will show that $\mathbb{E}[\Lambda^{NAW}] \leq \frac{11}{15}$. This will conclude this step of the proof since $\mathbb{E}[\Lambda_L^{SAW}] \geq \frac{3}{4} > \frac{11}{15} \geq \mathbb{E}[\Lambda^{NAW}]$.

By taking the derivative of (34) with respect to c, we can show that $\mathbb{E}[\Lambda_L^{SAW}]$ is increasing in c:

$$\frac{\partial(\mathbb{E}[\Lambda_L^{SAW}])}{\partial c} = \frac{2v_L}{4(v_L+c)^2} \geq 0.$$

Thus, $\mathbb{E}[\Lambda_L^{SAW}]$ is lower bounded when $c = v_L$:

$$\mathbb{E}[\Lambda_L^{SAW}] \geq \frac{v_L+2v_L}{2(v_L+v_L)} = \frac{3}{4}. \tag{49}$$

To bound $\mathbb{E}[\Lambda^{NAW}] \leq \frac{11}{15}$, we will upper bound (45) by $\frac{1}{4}\Big(\frac{2}{3}\Big)$, upper bound (46) by $\frac{1}{4}\Big(\frac{4}{5}\Big)$, upper bound (47) by $\frac{1}{4}\Big(\frac{2}{3}\Big)$, and upper bound (48) by $\frac{1}{4}\Big(\frac{4}{5}\Big)$. In doing so, we will have

$$\mathbb{E}[\Lambda^{NAW}] \leq \frac{1}{4}\Big(\frac{2}{3}\Big)+\frac{1}{4}\Big(\frac{4}{5}\Big)+\frac{1}{4}\Big(\frac{2}{3}\Big)+\frac{1}{4}\Big(\frac{4}{5}\Big) = \frac{11}{15}. \tag{50}$$

The proof technique is the same for each of these four upper bounds: first, we show that each term is increasing in c in order to find an upper bound when $c = v_H$, and then we show that the resulting bound is decreasing in v_L in order to find an upper bound when $v_L = 0$. Details are provided below.

To upper bound (45) by $\frac{1}{4}\Big(\frac{2}{3}\Big)$, we can take the derivative with respect to c to show that (45) is increasing in c:

$$\frac{\partial\Big(\frac{2c^2}{v_L^2+v_H^2+2c^2}\Big)}{\partial c} = \frac{4cv_L^2+4cv_H^2}{(v_L^2+v_H^2+2c^2)^2} > 0.$$

Thus, (45) is upper bounded when $c = v_H$:

$$\frac{2c^2}{v_L^2+v_H^2+2c^2} \leq \frac{2v_H^2}{v_L^2+3v_H^2}.$$

This is decreasing in v_L, so it can further be upper bounded when $v_L = 0$. Thus we have

$$\frac{1}{4}\Big(\frac{2c^2}{v_L^2+v_H^2+2c^2}\Big) \le \frac{1}{4}\Big(\frac{2v_H^2}{3v_H^2}\Big) = \frac{1}{6}. \tag{51}$$

To upper bound (46) by $\frac{1}{4}\Big(\frac{4}{5}\Big)$, we first note that each λ is upper bounded by 1, so it suffices to show the following bound for any value of c:

$$\frac{1}{16}\Big(\frac{c(v_L+c)+c(v_H+c)}{(v_L+c)^2+(v_H+c)^2}+1+\frac{c(-v_L+c)+c(v_H+c)}{(v_L-c)^2+(v_H+c)^2}+1\Big) \le \frac{1}{4}\Big(\frac{4}{5}\Big) = \frac{1}{5}. \tag{52}$$

Taking the derivative with respect to c of the first term, we have

$$\frac{\partial\Big(\frac{c(v_L+c)+c(v_H+c)}{(v_L+c)^2+(v_H+c)^2}\Big)}{\partial c} = \frac{v_L^3+2c^2v_L+v_Lv_H^2+4cv_L^2+v_H^3+v_L^2v_H+2c^2v_H+4cv_H^2}{((v_L+c)^2+(v_H+c)^2)^2} \ge 0$$

This gives us the following upper bound when $c = v_H$:

$$\frac{c(v_L+c)+c(v_H+c)}{(v_L+c)^2+(v_H+c)^2} \le \frac{v_Lv_H+3v_H^2}{v_L^2+2v_Lv_H+5v_H^2}.$$

We can further take the derivative of this bound with respect to v_L to show that it's decreasing in v_L:

$$\frac{\partial\Big(\frac{v_Lv_H+3v_H^2}{v_L^2+2v_Lv_H+5v_H^2}\Big)}{\partial v_L} = \frac{-v_L^2v_H-6v_Lv_H^2-v_H^3}{(v_L^2+2v_Lv_H+5v_H^2)^2} \le 0.$$

By using $v_L = 0$, we can get the bound

$$\frac{c(v_L+c)+c(v_H+c)}{(v_L+c)^2+(v_H+c)^2} \le \frac{3v_H^2}{5v_H^2} = \frac{3}{5}. \tag{53}$$

We can similarly bound the remaining term in (52) by first taking the derivative with respect to c:

$$\frac{\partial\Big(\frac{c(-v_L+c)+c(v_H+c)}{(v_L-c)^2+(v_H+c)^2}\Big)}{\partial c} = \frac{-v_L^3-2c^2v_L-v_Lv_H^2+4cv_L^2+4cv_H^2+v_L^2v_H+2c^2v_H+v_H^3}{((v_L-c)^2+(v_H+c)^2)^2} \ge 0.$$

This gives us the following upper bound when $c = v_H$:

$$\frac{c(-v_L+c)+c(v_H+c)}{(v_L-c)^2+(v_H+c)^2} \le \frac{3v_H^2-v_Lv_H}{v_L^2+5v_H^2-2v_Lv_H}.$$

We can further take the derivative of this bound with respect to v_L to show that it's decreasing in v_L:

$$\frac{\partial\Big(\frac{3v_H^2-v_Lv_H}{v_L^2+5v_H^2-2v_Lv_H}\Big)}{\partial v_L} = \frac{v_L^2v_H+v_H^3-6v_Lv_H^2}{(v_L^2+5v_H^2-2v_Lv_H)^2} \le 0.$$

By using $v_L = 0$, we can get the bound

$$\frac{c(-v_L+c)+c(v_H+c)}{(v_L-c)^2+(v_H+c)^2} \le \frac{3v_H^2}{5v_H^2} = \frac{3}{5}. \tag{54}$$

Using (53) and (54) to bound (52), we have the following upper bound for (46):

$$\frac{1}{16}\Big(\frac{c(v_L+c)+c(v_H+c)}{(v_L+c)^2+(v_H+c)^2}+1+\frac{c(-v_L+c)+c(v_H+c)}{(v_L-c)^2+(v_H+c)^2}+1\Big) \le \frac{1}{16}\Big(\frac{3}{5}+1+\frac{3}{5}+1\Big) = \frac{1}{5}. \tag{55}$$

To upper bound (47) by $\frac{1}{4}\Big(\frac{2}{3}\Big)$, we take the derivative of each term with respect to c and show that each term is increasing in c:

$$\frac{\partial\Big(\frac{4c^2+2cv_L}{2(v_L+c)^2+(v_H-c)^2+(v_H+c)^2}\Big)}{\partial c} = \frac{16cv_L^2+16cv_H^2+4v_L^3+4v_Lv_H^2+8c^2v_L}{(2(v_L+c)^2+(v_H-c)^2+(v_H+c)^2)^2} \ge 0;$$

$$\frac{\partial\Big(\frac{4c^2-2cv_L}{2(v_L-c)^2+(v_H-c)^2+(v_H+c)^2}\Big)}{\partial c}=\frac{16cv_L^2+16cv_H^2-4v_L^3-4v_Lv_H^2-8c^2v_L}{(2(v_L-c)^2+(v_H-c)^2+(v_H+c)^2)^2}\geq 0.$$

This gives us the following upper bound for (47) when $c=v_H$:

$$\begin{aligned}&\frac{1}{8}\Big(\frac{4c^2+2cv_L}{2(v_L+c)^2+(v_H-c)^2+(v_H+c)^2}+\frac{4c^2-2cv_L}{2(v_L-c)^2+(v_H-c)^2+(v_H+c)^2}\Big)\\&\leq\frac{1}{8}\Big(\frac{2v_H^2+v_Lv_H}{v_L^2+3v_H^2+2v_Lv_H}+\frac{2v_H^2-v_Lv_H}{v_L^2+3v_H^2-2v_Lv_H}\Big)=\frac{1}{8}\Big(\frac{12v_H^4}{v_L^4+9v_H^4+2v_L^2v_H^2}\Big)\end{aligned}$$

This is decreasing in v_L, so we can further upper bound (47) when $v_L=0$:

$$\frac{1}{8}\Big(\frac{4c^2+2cv_L}{2(v_L+c)^2+(v_H-c)^2+(v_H+c)^2}+\frac{4c^2-2cv_L}{2(v_L-c)^2+(v_H-c)^2+(v_H+c)^2}\Big)\leq\frac{1}{8}\Big(\frac{12v_H^4}{9v_H^4}\Big)=\frac{1}{6}.\tag{56}$$

To upper bound (48) by $\frac{1}{4}\Big(\frac{4}{5}\Big)$, we take the derivative of each term with respect to c and show that each term is increasing in c:

$$\frac{\partial\Big(\frac{4c^2+2cv_H}{(v_L-c)^2+(v_L+c)^2+2(v_H+c)^2}\Big)}{\partial c}=\frac{16cv_H^2+16cv_L^2+4v_H^3+4v_Hv_L^2+8c^2v_H}{((v_L-c)^2+(v_L+c)^2+2(v_H+c)^2)^2}\geq 0;$$

$$\frac{\partial\Big(\frac{4c^2-2cv_H}{(v_L-c)^2+(v_L+c)^2+2(v_H-c)^2}\Big)}{\partial c}=\frac{16cv_H^2+16cv_L^2-4v_H^3-4v_Hv_L^2-8c^2v_H}{((v_L-c)^2+(v_L+c)^2+2(v_H-c)^2)^2}\geq 0.$$

This gives us the following upper bound for (48) when $c=v_H$:

$$\begin{aligned}&\frac{1}{8}\Big(\frac{4c^2+2cv_H}{(v_L-c)^2+(v_L+c)^2+2(v_H+c)^2}+\frac{4c^2-2cv_H}{(v_L-c)^2+(v_L+c)^2+2(v_H-c)^2}\Big)\\&\leq\frac{1}{8}\Big(\frac{3v_H^2}{v_L^2+5v_H^2}+\frac{v_H^2}{v_L^2+v_H^2}\Big)=\frac{1}{8}\Big(\frac{8v_H^4+4v_L^2v_H^2}{5v_H^4+v_L^4+6v_L^2v_H^2}\Big).\end{aligned}$$

We can take the derivative of this bound with respect to v_L to show that it's decreasing in v_L:

$$\frac{\partial\Big(\frac{8v_H^4+4v_L^2v_H^2}{5v_H^4+v_L^4+6v_L^2v_H^2}\Big)}{\partial v_L}=\frac{-56v_Lv_H^6-8v_L^5v_H^2-32v_L^3v_H^4}{(5v_H^4+v_L^4+6v_L^2v_H^2)^2}\leq 0.$$

By using $v_L=0$, we can further bound (48) by

$$\frac{1}{8}\Big(\frac{4c^2+2cv_H}{(v_L-c)^2+(v_L+c)^2+2(v_H+c)^2}+\frac{4c^2-2cv_H}{(v_L-c)^2+(v_L+c)^2+2(v_H-c)^2}\Big)\leq\frac{1}{8}\Big(\frac{8v_H^4}{5v_H^4}\Big)=\frac{1}{5}.\tag{57}$$

Finally, using the bounds derived in (51), (55), (56), and (57), we can write

$$\mathbb{E}[\Lambda^{NAW}]\leq\frac{1}{6}+\frac{1}{5}+\frac{1}{6}+\frac{1}{5}=\frac{11}{15}.\tag{58}$$

From (49) and (58), we can conclude Step 2 of our proof since we have shown $\mathbb{E}[\Lambda_L^{SAW}]\geq\frac{3}{4}>\frac{11}{15}\geq\mathbb{E}[\Lambda^{NAW}]$.

Step 3

To prove $\mathbb{E}[\Lambda^{NAW}]>\mathbb{E}[\Lambda_H^{SAW}]$, we use (34), (45)-(48), and the fact that $\lambda_{(viii)}^{NAW}(c)\geq 0$ to write

$$\mathbb{E}[\Lambda^{NAW}]-\mathbb{E}[\Lambda_H^{SAW}]\geq\frac{1}{4}\Big(\frac{2c^2}{v_L^2+v_H^2+2c^2}\Big)\tag{59}$$

$$+\frac{1}{16}\Big(\frac{c(v_L+c)+c(v_H+c)}{(v_L+c)^2+(v_H+c)^2}+\frac{c(v_L+c)+c(-v_H+c)}{(v_L+c)^2+(v_H-c)^2}+\frac{c(-v_L+c)+c(v_H+c)}{(v_L-c)^2+(v_H+c)^2}\Big) \tag{60}$$

$$+\frac{1}{8}\Big(\frac{4c^2+2cv_L}{2(v_L+c)^2+(v_H-c)^2+(v_H+c)^2}+\frac{4c^2-2cv_L}{2(v_L-c)^2+(v_H-c)^2+(v_H+c)^2}\Big) \tag{61}$$

$$-\frac{3}{4}\Big(\frac{c}{2c+2v_H}\Big) \tag{62}$$

$$+\frac{1}{8}\Big(\frac{4c^2+2cv_H}{(v_L-c)^2+(v_L+c)^2+2(v_H+c)^2}+\frac{4c^2-2cv_H}{(v_L-c)^2+(v_L+c)^2+2(v_H-c)^2}-\frac{2c}{2c+2v_H}\Big), \tag{63}$$

which we aim to show is non-negative. To do so, we will lower bound the right-hand side of (59) by $\frac{1}{4}\Big(\frac{2}{7}\Big)$, lower bound (60) by $\frac{1}{16}\Big(\frac{179}{195}\Big)$, lower bound (61) by $\frac{1}{8}\Big(\frac{8}{15}\Big)$, lower bound (62) by $-\frac{3}{4}\Big(\frac{1}{4}\Big)$, and lower bound (63) by $\frac{1}{8}\Big(\frac{1}{33}\Big)$. This would give us

$$\mathbb{E}[\Lambda^{NAW}]-\mathbb{E}[\Lambda_H^{SAW}]\geq\frac{1}{4}\Big(\frac{2}{7}\Big)+\frac{1}{16}\Big(\frac{179}{195}\Big)+\frac{1}{8}\Big(\frac{8}{15}\Big)-\frac{3}{4}\Big(\frac{1}{4}\Big)+\frac{1}{8}\Big(\frac{1}{33}\Big)=\frac{353}{30030}>0, \tag{64}$$

concluding our proof.

To find a lower bound for the right-hand side of (59), we already know from Step 2 that it is increasing in c, thus we can use $c=v_L$ as our lower bound:

$$\frac{1}{4}\Big(\frac{2c^2}{v_L^2+v_H^2+2c^2}\Big)\geq\frac{1}{4}\Big(\frac{2v_L^2}{3v_L^2+v_H^2}\Big).$$

This bound is decreasing in v_H, so we can use $v_H=2v_L$ to further bound

$$\frac{1}{4}\Big(\frac{2c^2}{v_L^2+v_H^2+2c^2}\Big)\geq\frac{1}{4}\Big(\frac{2v_L^2}{3v_L^2+(2v_L)^2}\Big)=\frac{1}{4}\Big(\frac{2}{7}\Big). \tag{65}$$

To find a lower bound for (60), we already know from Step 2 that each term is increasing in c, thus we can use $c=v_L$ as our lower bound:

$$\frac{1}{16}\Big(\frac{c(v_L+c)+c(v_H+c)}{(v_L+c)^2+(v_H+c)^2}+\frac{c(v_L+c)+c(-v_H+c)}{(v_L+c)^2+(v_H-c)^2}+\frac{c(-v_L+c)+c(v_H+c)}{(v_L-c)^2+(v_H+c)^2}\Big)$$
$$\geq\frac{1}{16}\Big(\frac{3v_L^2+v_Lv_H}{5v_L^2+v_H^2+2v_Lv_H}+\frac{3v_L^2-v_Lv_H}{5v_L^2+v_H^2-2v_Lv_H}+\frac{v_L}{v_H+v_L}\Big).$$

We next show that each of these terms is decreasing in v_H so that we can further bound using $v_H=2v_L$. The third term is trivially decreasing in v_H. Taking the derivative of the first two terms with respect to v_H, we have

$$\frac{\partial\Big(\frac{3v_L^2+v_Lv_H}{5v_L^2+v_H^2+2v_Lv_H}\Big)}{\partial v_H}=\frac{-6v_L^2v_H-v_L^3-v_Lv_H^2}{(5v_L^2+v_H^2+2v_Lv_H)^2}\leq 0;$$

$$\frac{\partial\Big(\frac{3v_L^2-v_Lv_H}{5v_L^2+v_H^2-2v_Lv_H}\Big)}{\partial v_H}=\frac{-6v_L^2v_H+v_L^3+v_Lv_H^2}{(5v_L^2+v_H^2-2v_Lv_H)^2}\leq 0.$$

Thus we have

$$\frac{1}{16}\Big(\frac{c(v_L+c)+c(v_H+c)}{(v_L+c)^2+(v_H+c)^2}+\frac{c(v_L+c)+c(-v_H+c)}{(v_L+c)^2+(v_H-c)^2}+\frac{c(-v_L+c)+c(v_H+c)}{(v_L-c)^2+(v_H+c)^2}\Big)$$
$$\geq\frac{1}{16}\Big(\frac{5v_L^2}{13v_L^2}+\frac{v_L}{5v_L^2}+\frac{v_L}{3v_L}\Big)=\frac{1}{16}\Big(\frac{179}{195}\Big). \tag{66}$$

To find a lower bound for (61), we already know from Step 2 that each term is increasing in c, thus we can use $c = v_L$ as our lower bound:

$$\begin{aligned}
&\frac{1}{8}\Big(\frac{4c^2+2cv_L}{2(v_L+c)^2+(v_H-c)^2+(v_H+c)^2}+\frac{4c^2-2cv_L}{2(v_L-c)^2+(v_H-c)^2+(v_H+c)^2}\Big)\\
&\geq \frac{1}{8}\Big(\frac{6v_L^2}{8v_L^2+(v_H-v_L)^2+(v_H+v_L)^2}+\frac{2v_L^2}{(v_H-v_L)^2+(v_H+v_L)^2}\Big).
\end{aligned}$$

Since both terms of the bound are decreasing in v_H, we can use $v_H = 2v_L$ to further bound

$$\frac{1}{8}\Big(\frac{4c^2+2cv_L}{2(v_L+c)^2+(v_H-c)^2+(v_H+c)^2}+\frac{4c^2-2cv_L}{2(v_L-c)^2+(v_H-c)^2+(v_H+c)^2}\Big) \geq \frac{1}{8}\Big(\frac{6v_L^2}{18v_L^2}+\frac{2v_L^2}{10v_L^2}\Big) \geq \frac{1}{8}\Big(\frac{8}{15}\Big) \tag{67}$$

To find a lower bound for (62), we can take the derivative with respect to c to show that it is decreasing in c:

$$\frac{\partial\Big(\frac{-c}{2c+2v_H}\Big)}{\partial c} = \frac{-2v_H}{(2c+2v_H)^2} \leq 0.$$

Thus we can use $c = v_H$ as our lower bound:

$$\frac{3}{4}\Big(\frac{-c}{2c+2v_H}\Big) \geq \frac{3}{4}\Big(\frac{-v_H}{4v_H}\Big) \geq -\frac{3}{4}\Big(\frac{1}{4}\Big). \tag{68}$$

To find a lower bound for (63), note that we already know from Step 2 that the first term is increasing in c. We next aim to show that, together, the last two terms are also increasing in c (we must do this since the last term in isolation is not increasing in c). Combining the last two terms, we have

$$\begin{aligned}
\frac{4c^2-2cv_H}{(v_L-c)^2+(v_L+c)^2+2(v_H-c)^2}-\frac{c}{c+v_H} &= \frac{(4c^2-2cv_H)(c+v_H)-c(2v_L^2+2v_H^2-4cv_H+4c^2)}{(2v_L^2+2v_H^2-4cv_H+4c^2)(c+v_H)}\\
&= \frac{3c^2v_H-2cv_H^2-cv_L^2}{v_H^3+2c^3+v_L^2v_H+cv_L^2-cv_H^2}.
\end{aligned}$$

Taking the derivative with respect to c gives us

$$\begin{aligned}
&\frac{\partial\Big(\frac{3c^2v_H-2cv_H^2-cv_L^2}{v_H^3+2c^3+v_L^2v_H+cv_L^2-cv_H^2}\Big)}{\partial c}\\
&= \frac{3c^2v_L^2v_H+8c^3v_H^2+4c^3v_L^2+6cv_H^4+6cv_L^2v_H^2-6c^4v_H-3c^2v_H^3-2v_H^5-3v_L^2v_H^3-v_L^4v_H}{(v_H^3+2c^3+v_L^2v_H+cv_L^2-cv_H^2)^2} \geq 0.
\end{aligned}$$

Thus, we can use $c = v_L$ as a lower bound:

$$\begin{aligned}
&\frac{1}{8}\Big(\frac{4c^2+2cv_H}{(v_L-c)^2+(v_L+c)^2+2(v_H+c)^2}+\frac{4c^2-2cv_H}{(v_L-c)^2+(v_L+c)^2+2(v_H-c)^2}-\frac{2c}{2c+2v_H}\Big)\\
&= \frac{1}{8}\Big(\frac{4c^2+2cv_H}{(v_L-c)^2+(v_L+c)^2+2(v_H+c)^2}+\frac{3c^2v_H-2cv_H^2-cv_L^2}{v_H^3+2c^3+v_L^2v_H+cv_L^2-cv_H^2}\Big)\\
&\geq \frac{1}{8}\Big(\frac{4v_L^2+2v_Lv_H}{6v_L^2+2v_H^2+4v_Lv_H}+\frac{3v_L^2v_H-2v_Lv_H^2-v_L^3}{v_H^3+3v_L^3+v_L^2v_H-v_Lv_H^2}\Big).
\end{aligned}$$

We next show that each of these terms is decreasing in v_H so that we can further bound using $v_H = 2v_L$:

$$\frac{\partial\Big(\frac{4v_L^2+2v_Lv_H}{6v_L^2+2v_H^2+4v_Lv_H}\Big)}{\partial v_H} = \frac{-4v_L^3-4v_Lv_H^2-16v_L^2v_H}{(6v_L^2+2v_H^2+4v_Lv_H)^2} \leq 0;$$

$$\frac{\partial\left(\frac{3v_L^2 v_H - 2v_L v_H^2 - v_L^3}{v_H^3 + 3v_L^3 + v_L^2 v_H - v_L v_H^2}\right)}{\partial v_H} = \frac{10v_L^5 - 5v_L^2 v_H^3 - 14v_L^4 v_H + 4v_L^3 v_H^2 + 2v_L v_H^4}{(v_H^3 + 3v_L^3 + v_L^2 v_H - v_L v_H^2)^2} \leq 0.$$

Thus we can bound (63) with

$$\begin{aligned}&\frac{1}{8}\Big(\frac{4c^2 + 2cv_H}{(v_L - c)^2 + (v_L + c)^2 + 2(v_H + c)^2} + \frac{4c^2 - 2cv_H}{(v_L - c)^2 + (v_L + c)^2 + 2(v_H - c)^2} - \frac{2c}{2c + 2v_H}\Big)\\ \geq{}& \frac{1}{8}\Big(\frac{4v_L^2 + 2v_L(2v_L)}{6v_L^2 + 2(2v_L)^2 + 4v_L(2v_L)} + \frac{3v_L^2(2v_L) - 2v_L(2v_L)^2 - v_L^3}{(2v_L)^3 + 3v_L^3 + v_L^2(2v_L) - v_L(2v_L)^2}\Big) = \frac{1}{8}\Big(\frac{8v_L^2}{22v_L^2} - \frac{3v_L^3}{9v_L^3}\Big) = \frac{1}{8}\Big(\frac{1}{33}\Big) > 0.\end{aligned} \tag{69}$$

With the bounds from (65)-(69), our proof follows immediately from (64). □

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 16), we use a regression model. A t-test is not appropriate for this analysis given that participants in the *Mixed Impact* treatment condition generate two $MedWOA$ observations each while participants in the *Always Low Impact* and *Always High Impact* 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* treatment condition and across participants in the two *Always Low Impact* and *Always High Impact* 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):

$$MedWOA_j^k = \beta_0 + \beta_1 \text{Low Impact}_j^k + \beta_2 \text{Single Exposure Set}_j + \beta_3 \text{Low Impact}_j^k \times \text{Single Exposure Set}_j \tag{70}$$

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 16 then reduces to:

$$(\beta_0 + \beta_1) - (\beta_0) \leq (\beta_0 + \beta_1 + \beta_2 + \beta_3) - (\beta_0 + \beta_2) \tag{71}$$

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

Table 3 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 $\|v_i\|$)	0.0223
	(0.0220)
Single Exposure Set	-0.2143***
	(0.0454)
Low $\|v_i\|$ × 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' windorized 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.

Table 4 Participants' Task-Level Weight on Algorithm Results

Dependent Variable:	WOA (Winsorized)				
Model:	Single Expo Set	Mixed Expo Set	Low $\lvert v_i \rvert$	High $\lvert v_i \rvert$	All Data
Variables					
Low $\lvert v_i \rvert$	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 $\lvert v_i \rvert$ × 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 5 Participants' Task-Level Prediction Error Results

Dependent Variable:	Final Absolute Error ($\lvert \hat{y}_{ij}^{final} - y_i \rvert$)			
Model:	Single Expo Set	Mixed Expo Set	Low $\lvert v_i \rvert$	High $\lvert v_i \rvert$
Variables				
Low $\lvert v_i \rvert$	-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 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 6 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 7 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.4. 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 8 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.5. Experiment 1 Initial Predictions ($\hat{y}_{ij}^{init}$) Without the Algorithm in Step 3 vs. Step 5

One might be concerned that subjects 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 9 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.6. 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 10 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 $\|v_i\|$ & Single Expo Set	M=11.556 (SD=9.771)	M=7.948 (SD=14.151)
High $\|v_i\|$ & Single Expo Set	M=15.766 (SD=18.117)	M=7.142 (SD=8.059)
Low $\|v_i\|$ & Mixed Expo Set	M=11.054 (SD=9.393)	M=6.937 (SD=5.712)
High $\|v_i\|$ & 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 11 Unpaired t-test comparisons of Step 5 prediction time

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

Participants' optimal final predictions fell within their advice-weighting regions at proportions similar to in Study 1. Furthermore, the proportion of instances where the optimal final prediction 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 optimal final prediction 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 optimal final prediction 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 information 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 12 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 13 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: Model:	$RMedSE$ (1)	SD(WOA) (2)	$MedWOA^L - MedWOA^H$ (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

$$APAE_i = \left| \frac{(\hat{y}_i^{final} - \hat{y}_i^{alg}) - (\mathbb{E}[Y_i] - \hat{y}_i^{alg})}{\mathbb{E}[Y_i] - \hat{y}_i^{alg}} \right| \tag{72}$$

where $\mathbb{E}[Y_i]$ is the hyper-rational 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 impact of private feature (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 14 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 15 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*. See Appendix C.4 for details. This may be 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 16 **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)

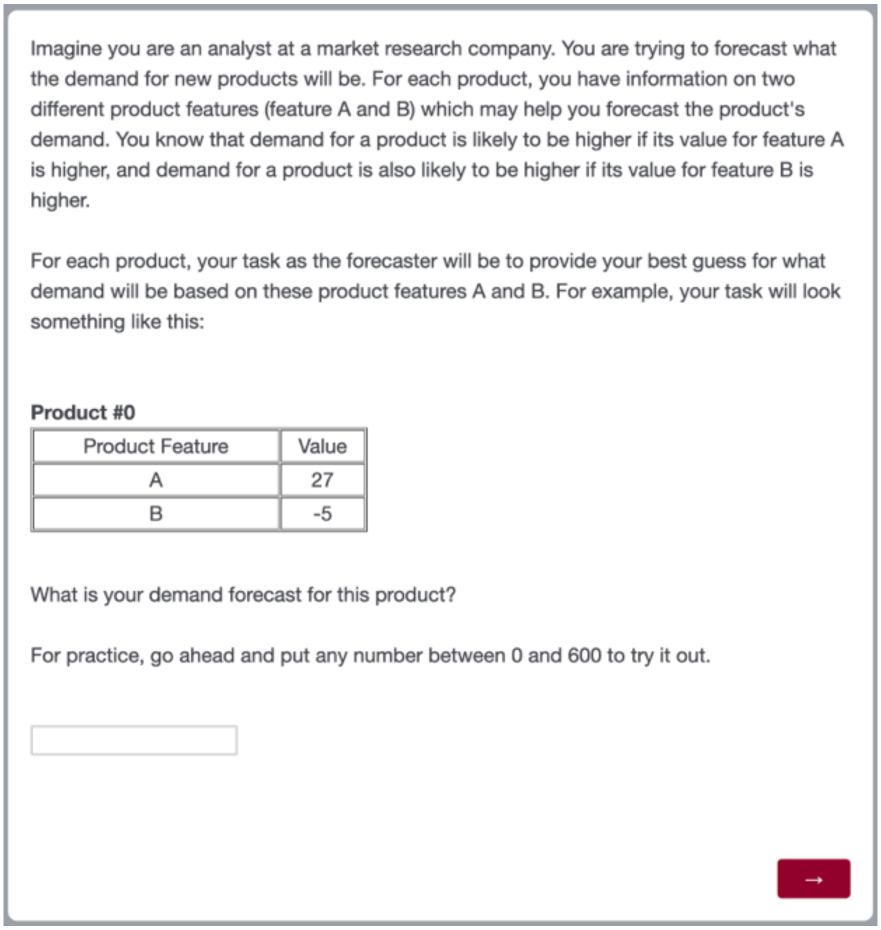
Timings are averaged per participant and impact of private feature and means are taken across conditions

Table 17 **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 1 Participant Experience

D.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.

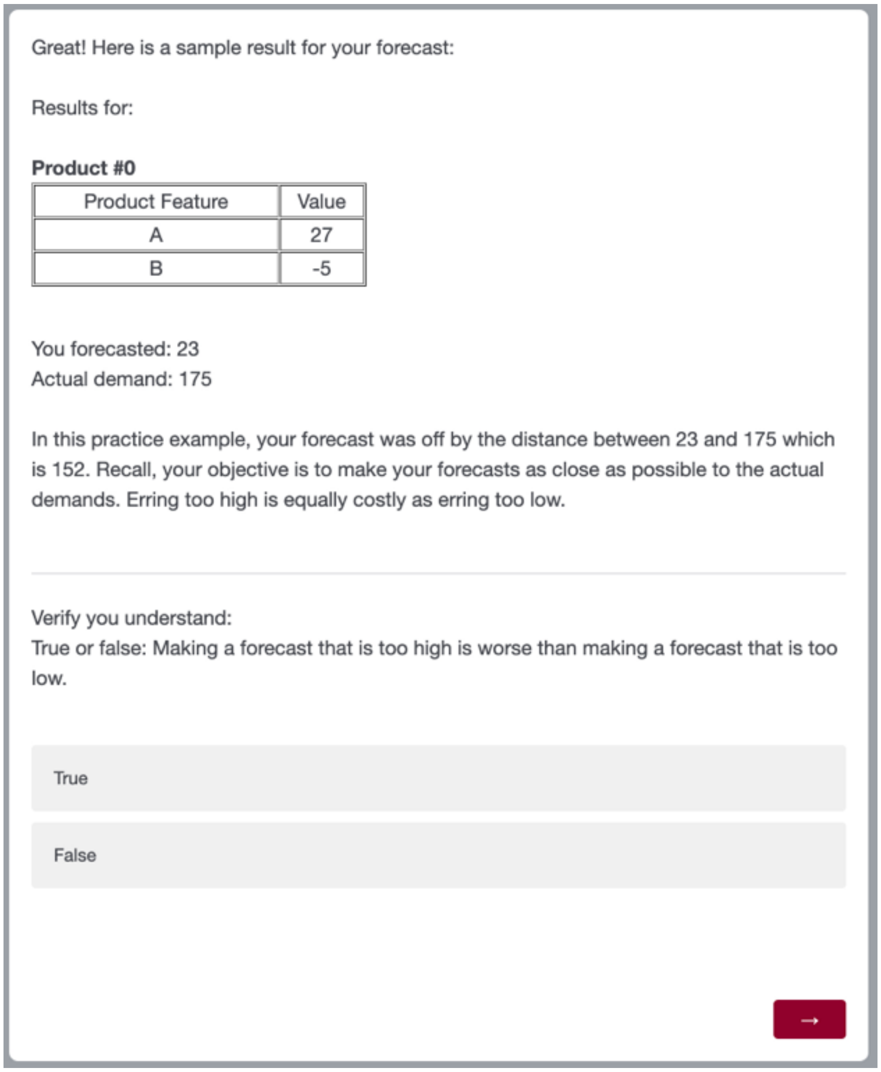
For each product, your task as the forecaster will be to provide your best guess for what demand will be based on these product features A and B. For example, your task will look something like this:

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:

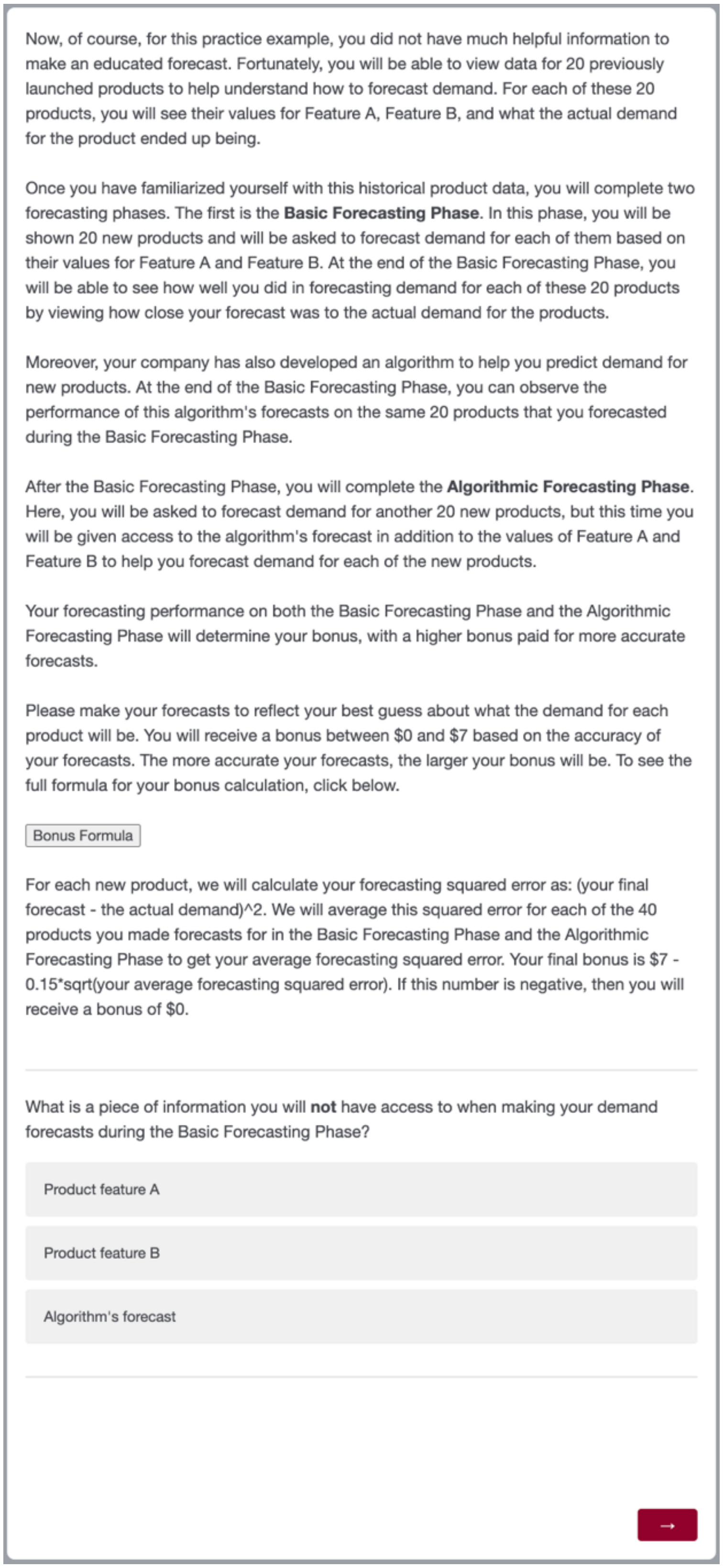
Product #0

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.

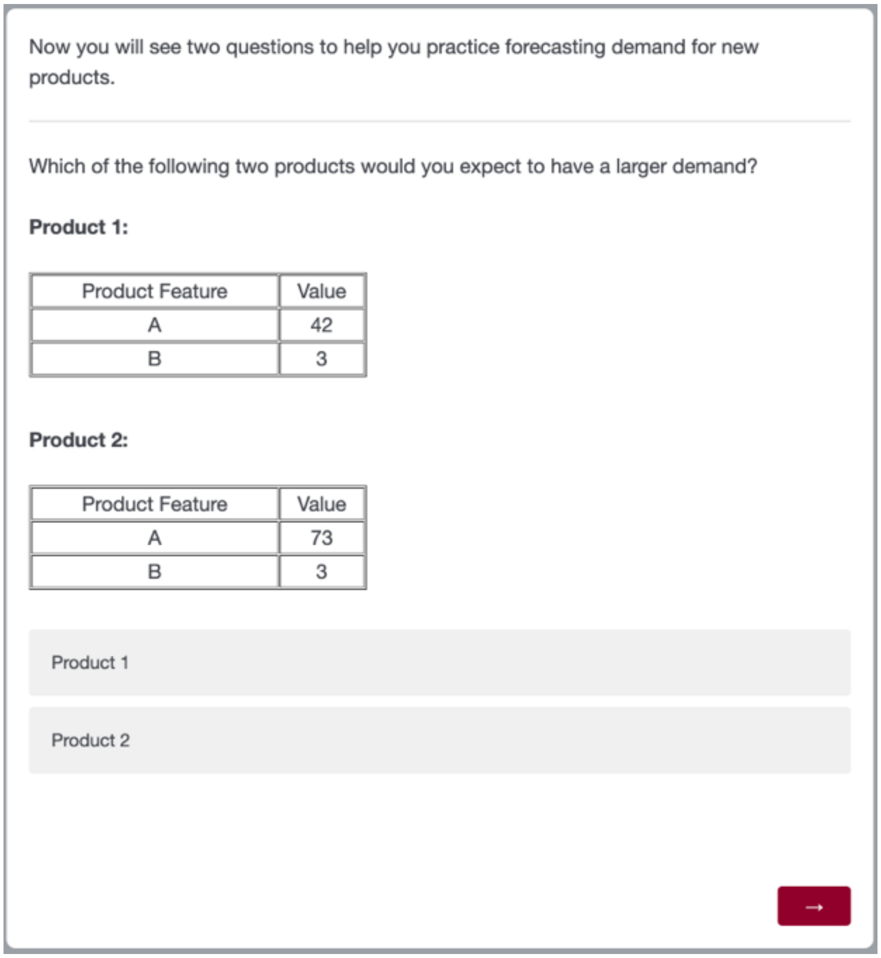
For each new product, we will calculate your forecasting squared error as: (your final forecast - the actual demand)^2. We will average this squared error for each of the 40 products you made forecasts for in the Basic Forecasting Phase and the Algorithmic Forecasting Phase to get your average forecasting 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.

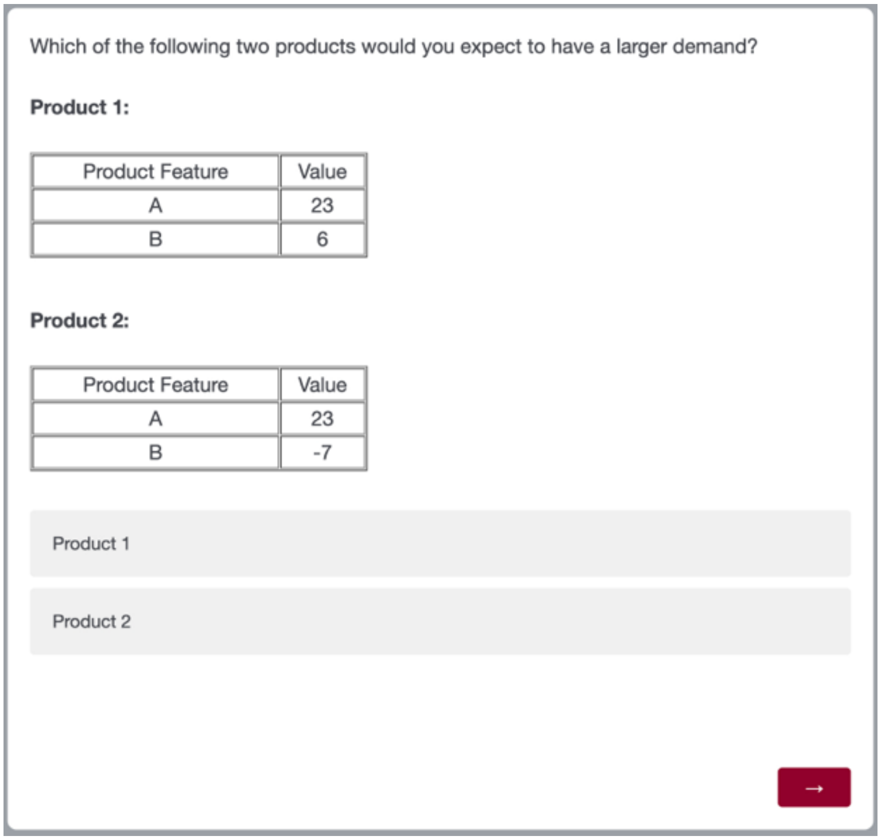
Which of the following two products would you expect to have a larger demand?

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

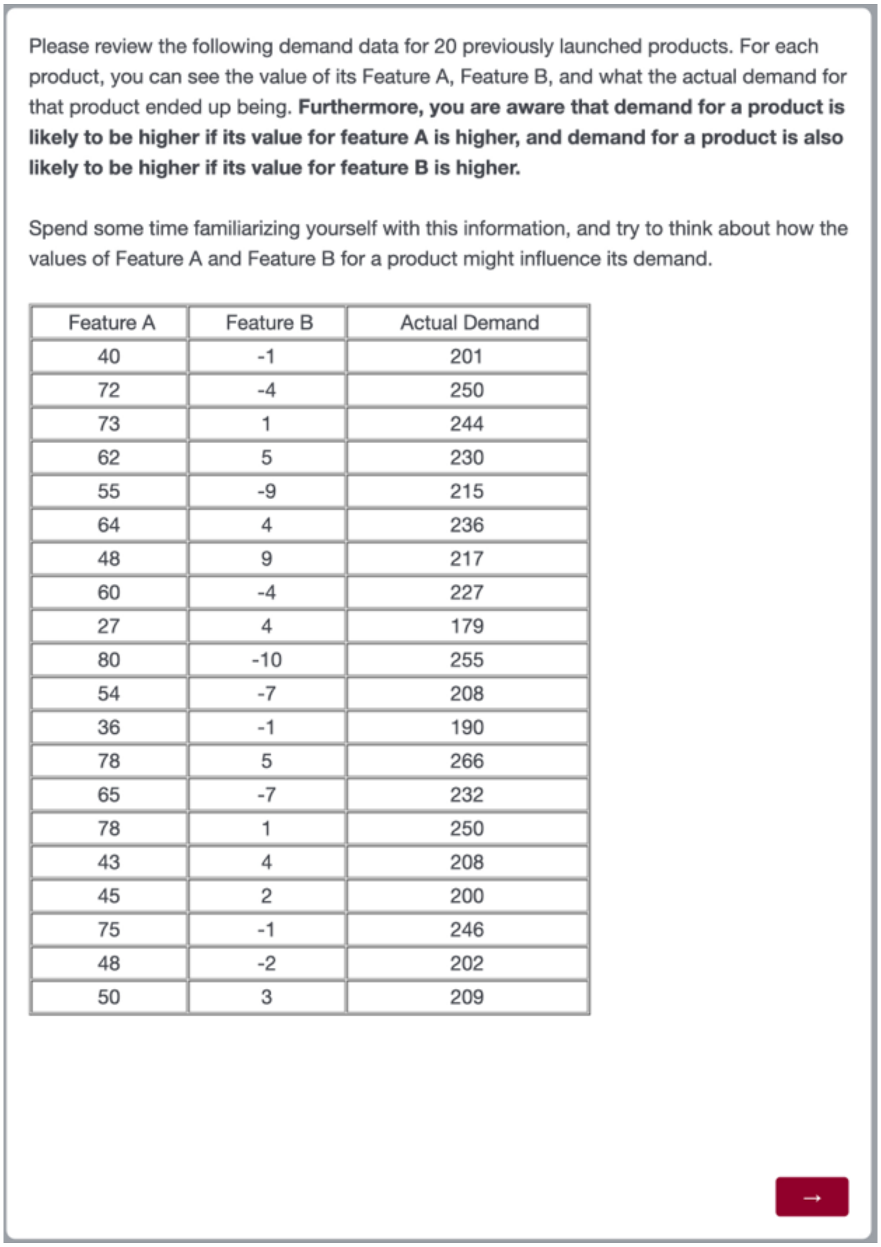
Product 2:

Product Feature	Value
A	23
B	-7

Product 1

Product 2

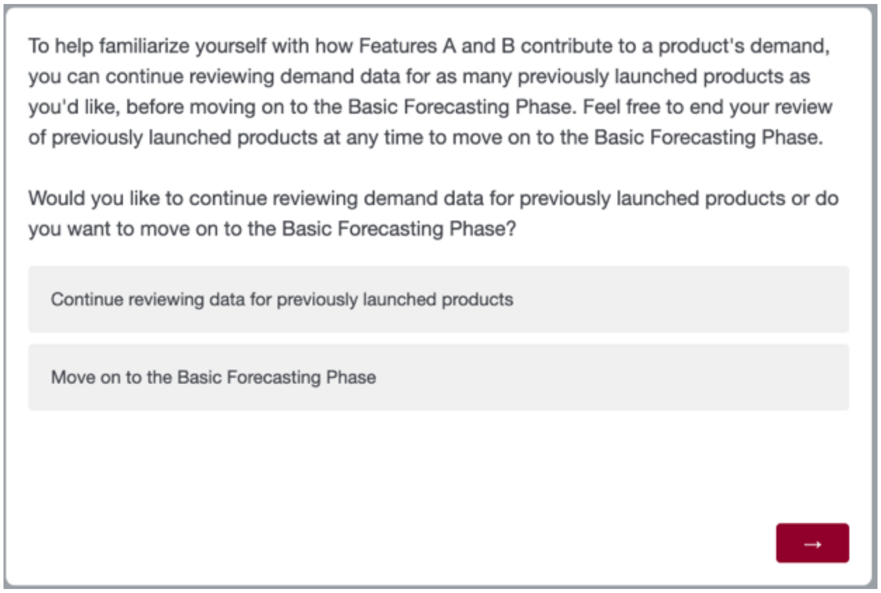
D.2. Step 2: Historical Data Review



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

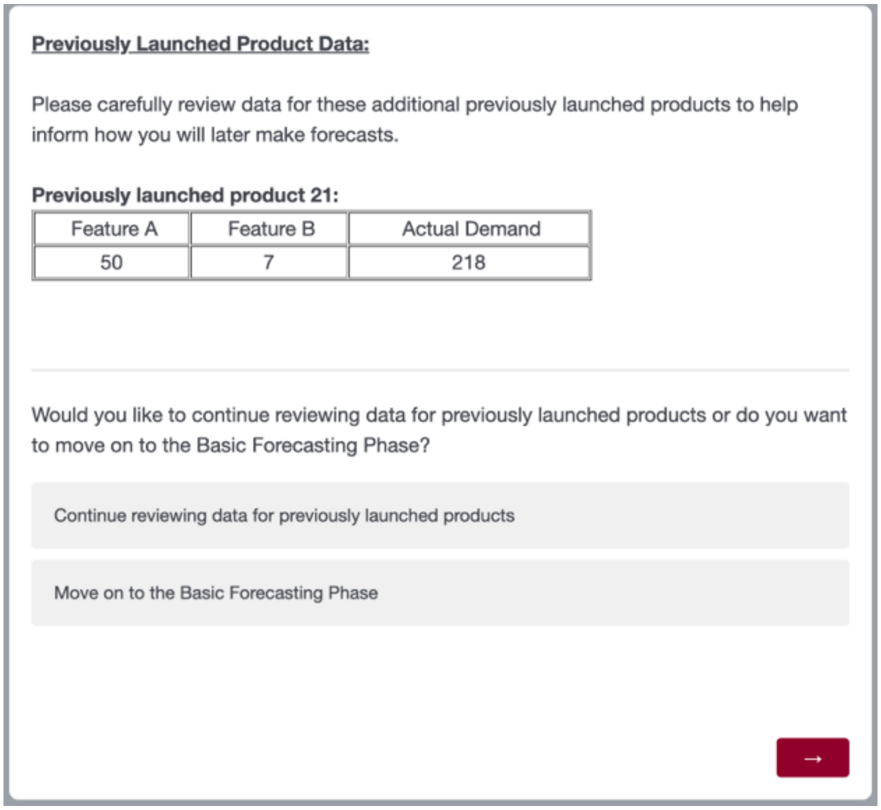


To help familiarize yourself with how Features A and B contribute to a product's demand, you can continue reviewing demand data for as many previously launched products as you'd like, before moving on to the Basic Forecasting Phase. Feel free to end your review of previously launched products at any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

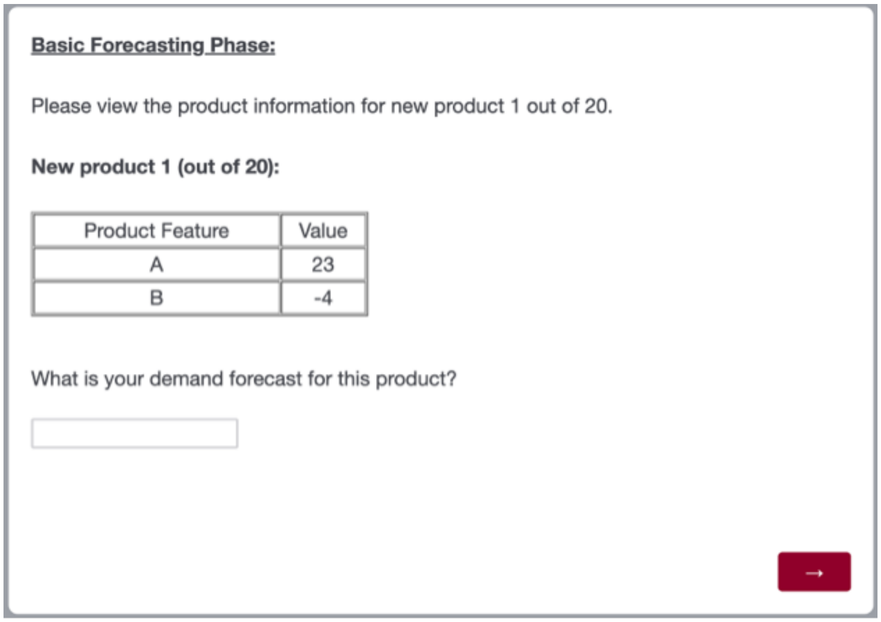
Feature A	Feature B	Actual Demand
50	7	218

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

D.3. Step 3: Demand Predictions without Algorithm



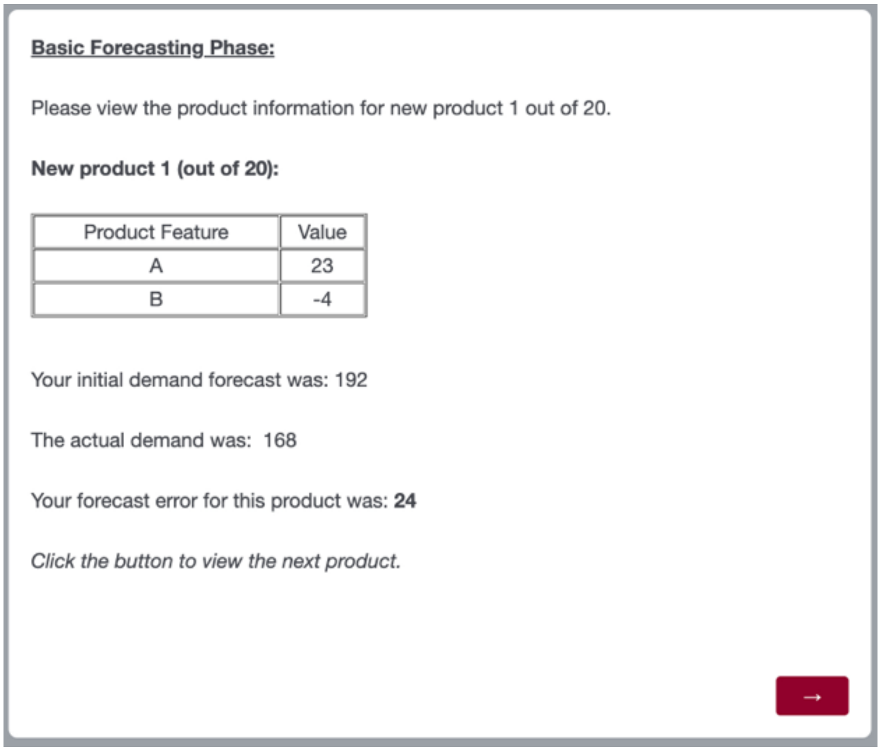
Basic Forecasting Phase:

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

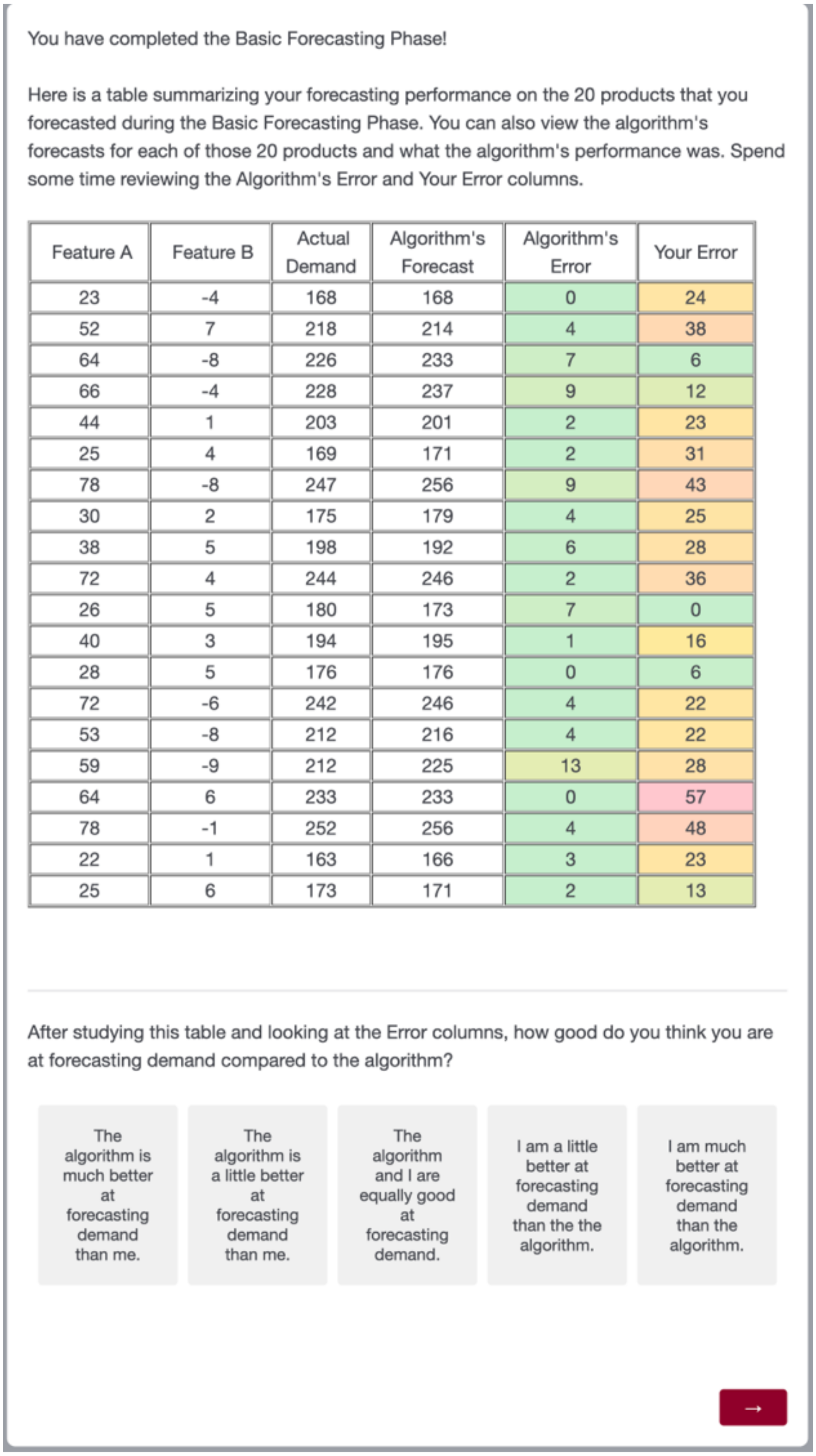
Your initial demand forecast was: 192

The actual demand was: 168

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

Click the button to view the next product.

D.4. Step 4: Algorithm Introduction



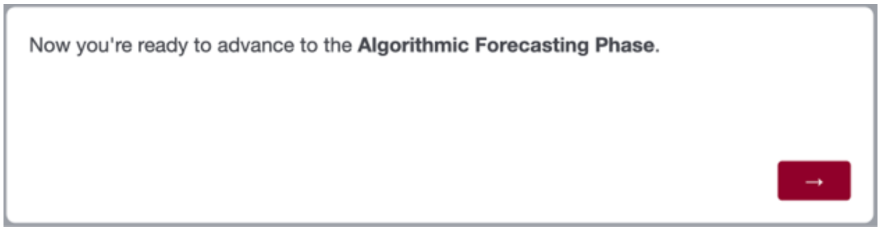
You have completed the Basic Forecasting Phase!

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.

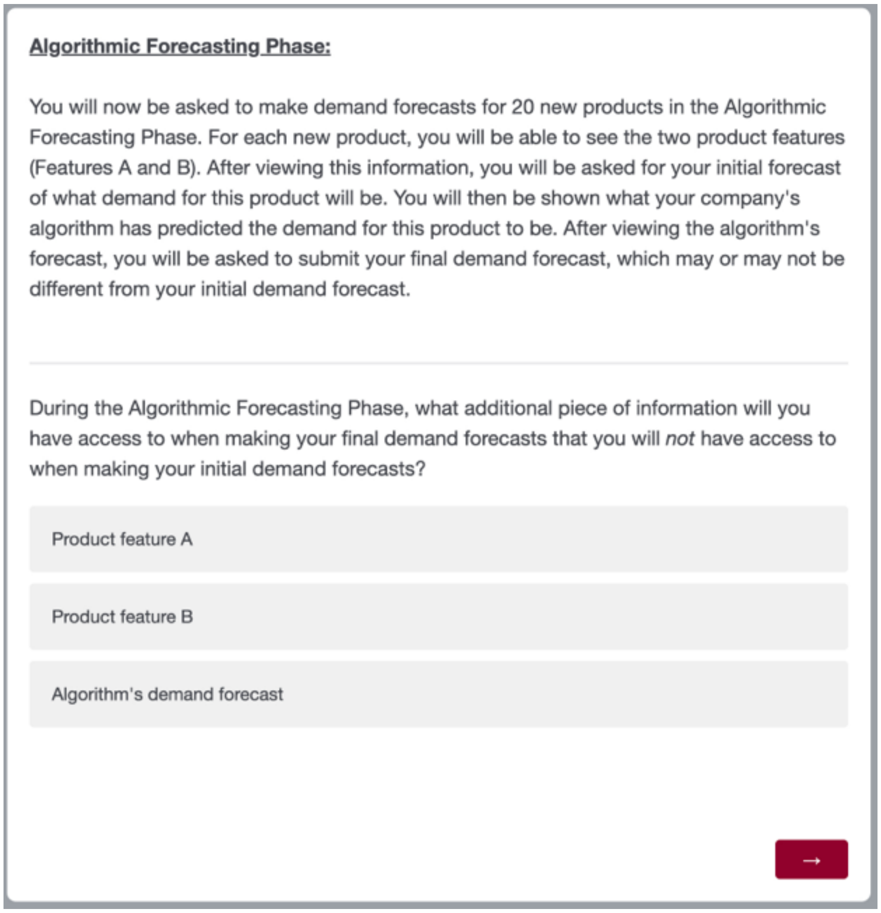
Feature A	Feature B	Actual Demand	Algorithm's Forecast	Algorithm's Error	Your Error
23	-4	168	168	0	24
52	7	218	214	4	38
64	-8	226	233	7	6
66	-4	228	237	9	12
44	1	203	201	2	23
25	4	169	171	2	31
78	-8	247	256	9	43
30	2	175	179	4	25
38	5	198	192	6	28
72	4	244	246	2	36
26	5	180	173	7	0
40	3	194	195	1	16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



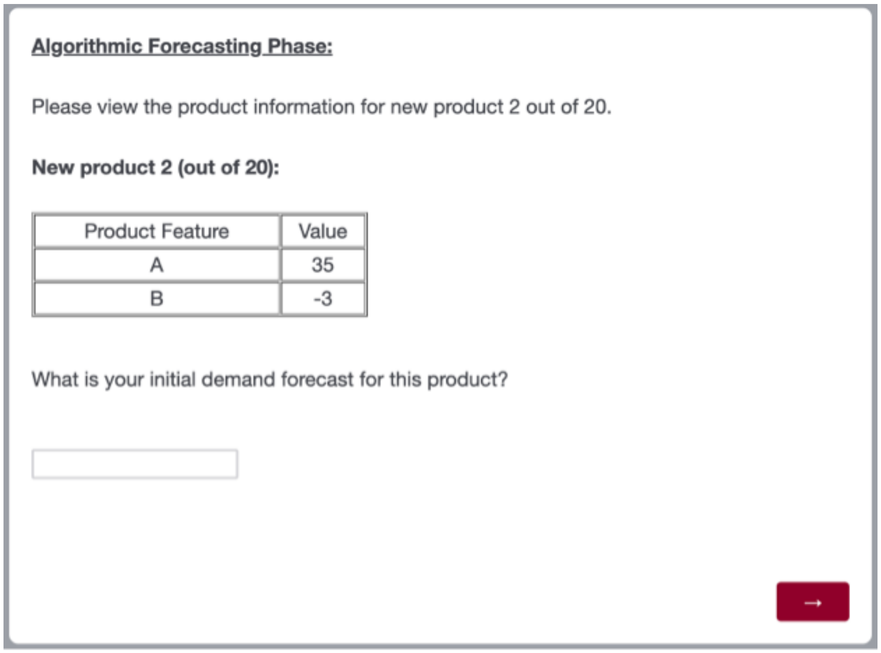
Algorithmic Forecasting Phase:

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

D.5. Step 5: Demand Predictions with Algorithm



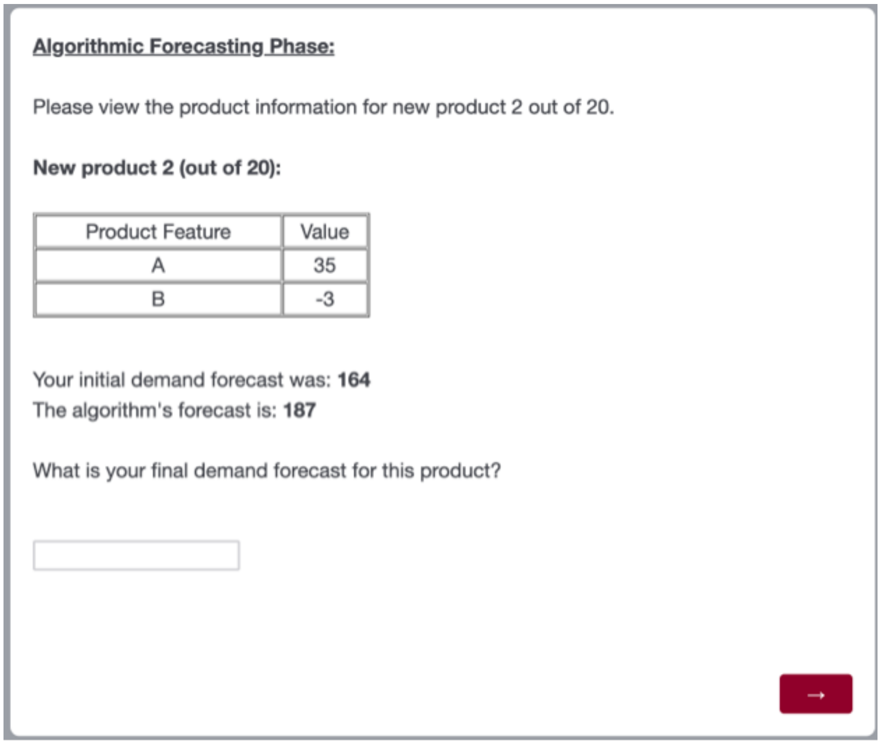
Algorithmic Forecasting Phase:

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:

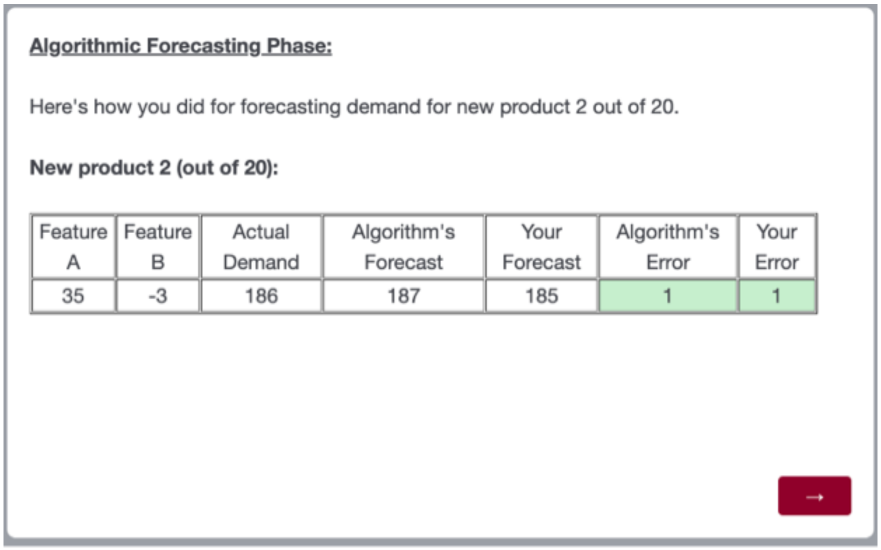
Please view the product information for new product 2 out of 20.

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:

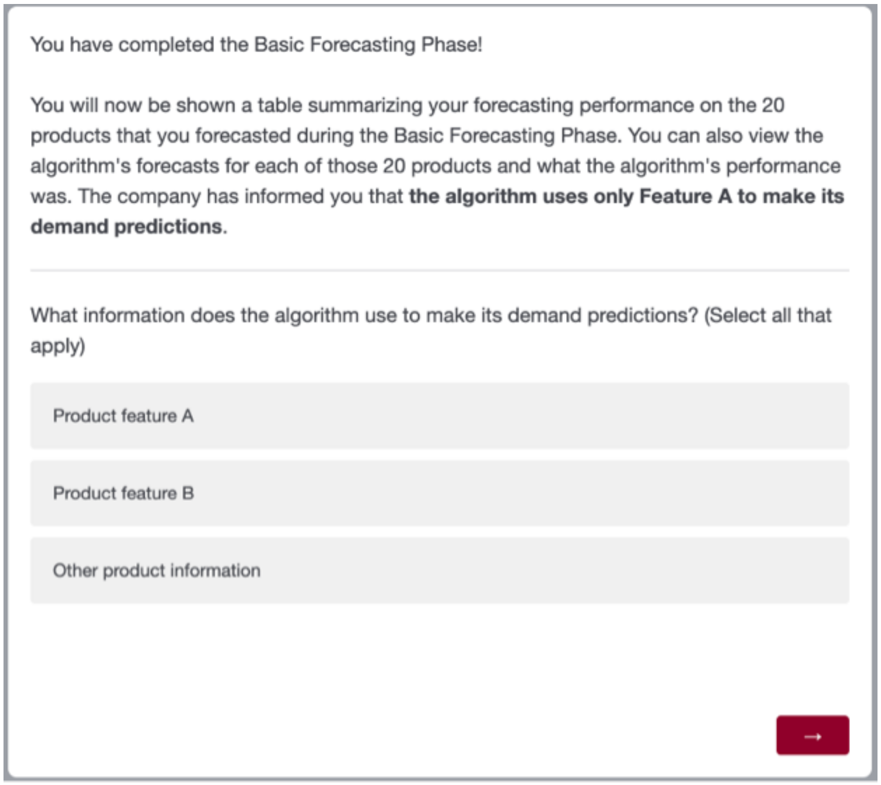
Here's how you did for forecasting demand for new product 2 out of 20.

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 E: Experiment 2 Participant Experience Changes

E.1. Feature Transparency



You have completed the Basic Forecasting Phase!

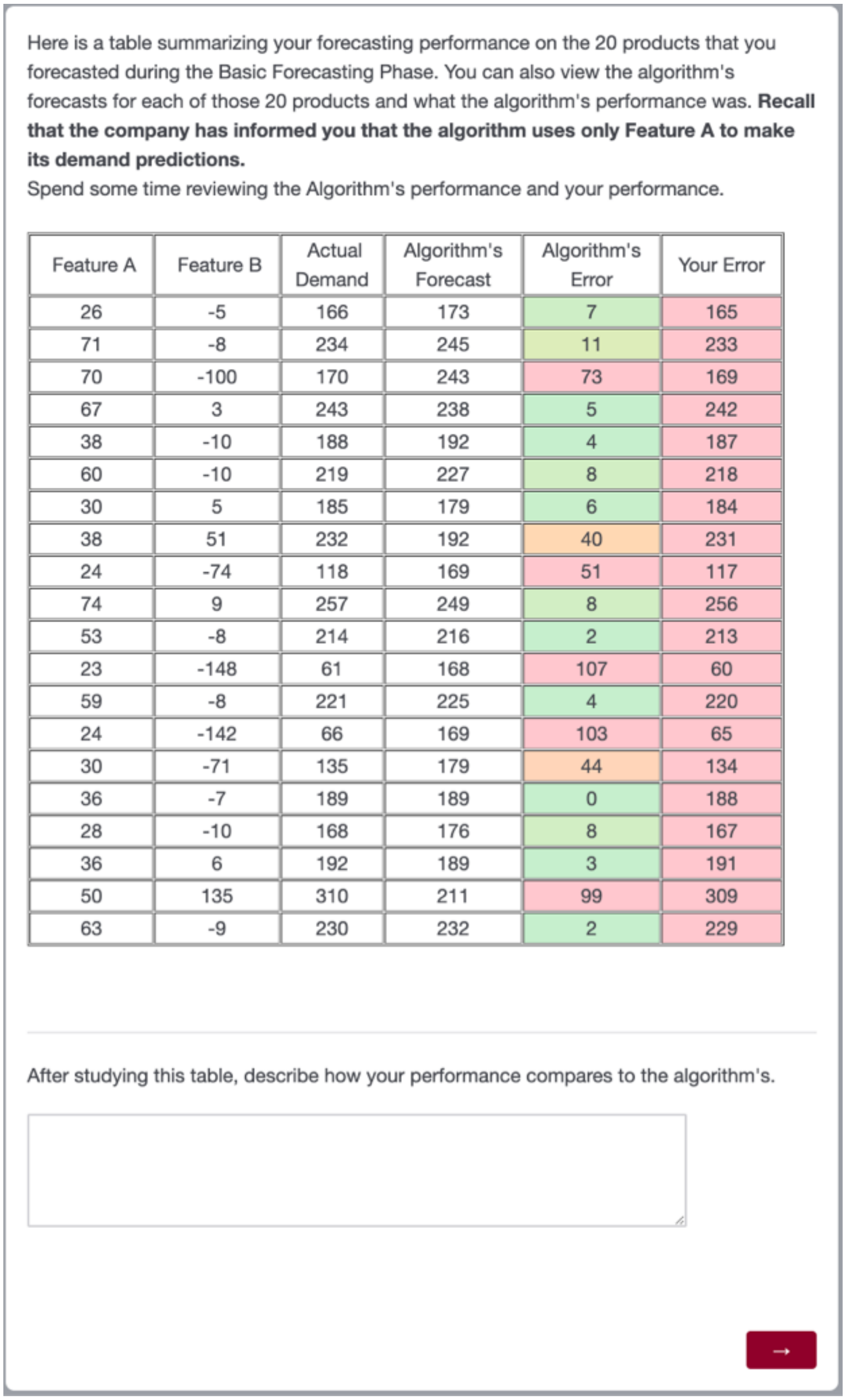
You will now be shown a table summarizing your forecasting performance on the 20 products that you forecasted during the Basic Forecasting Phase. You can also view the algorithm's forecasts for each of those 20 products and what the algorithm's performance was. The company has informed you that **the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

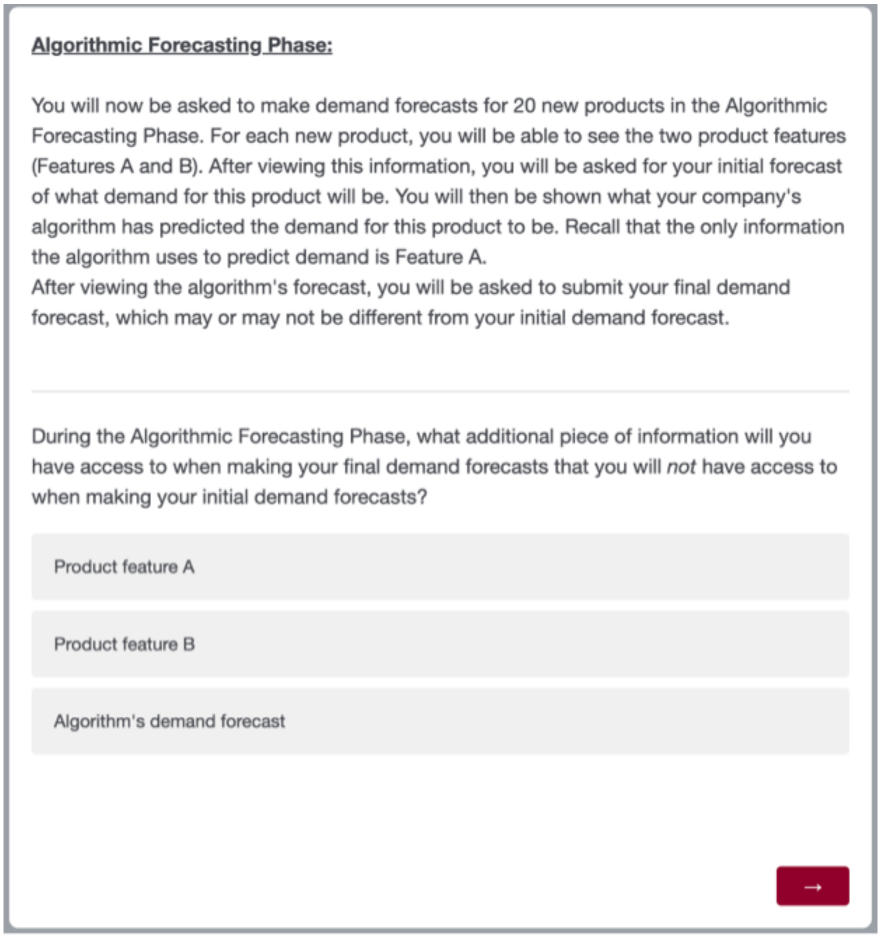


Here is a table summarizing your forecasting performance on the 20 products that you forecasted during the Basic Forecasting Phase. You can also view the algorithm's forecasts for each of those 20 products and what the algorithm's performance was. **Recall that the company has informed you that the algorithm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

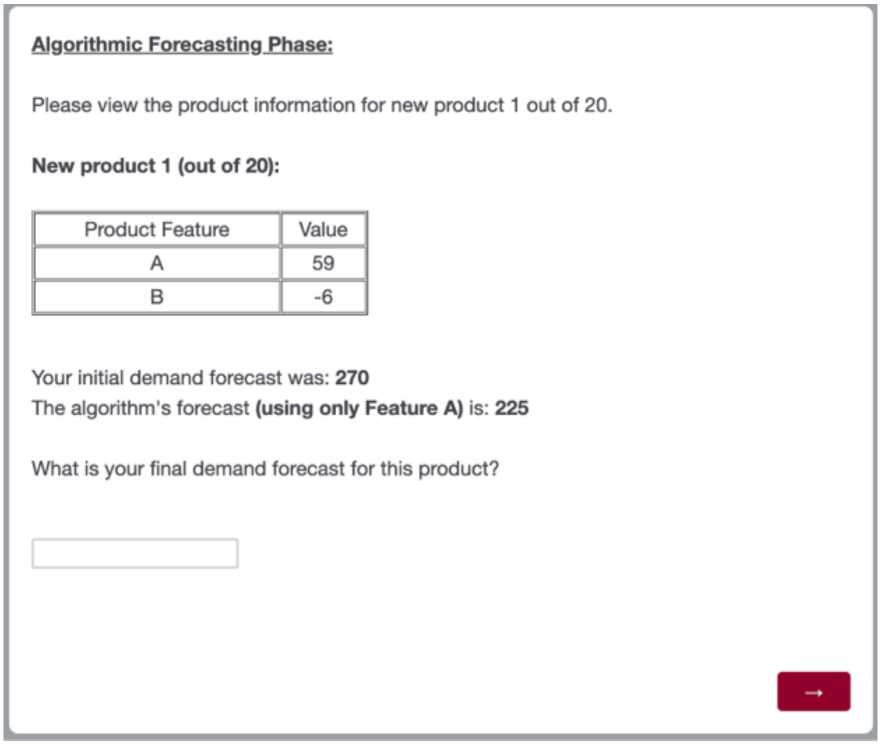


Algorithmic Forecasting Phase:

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.

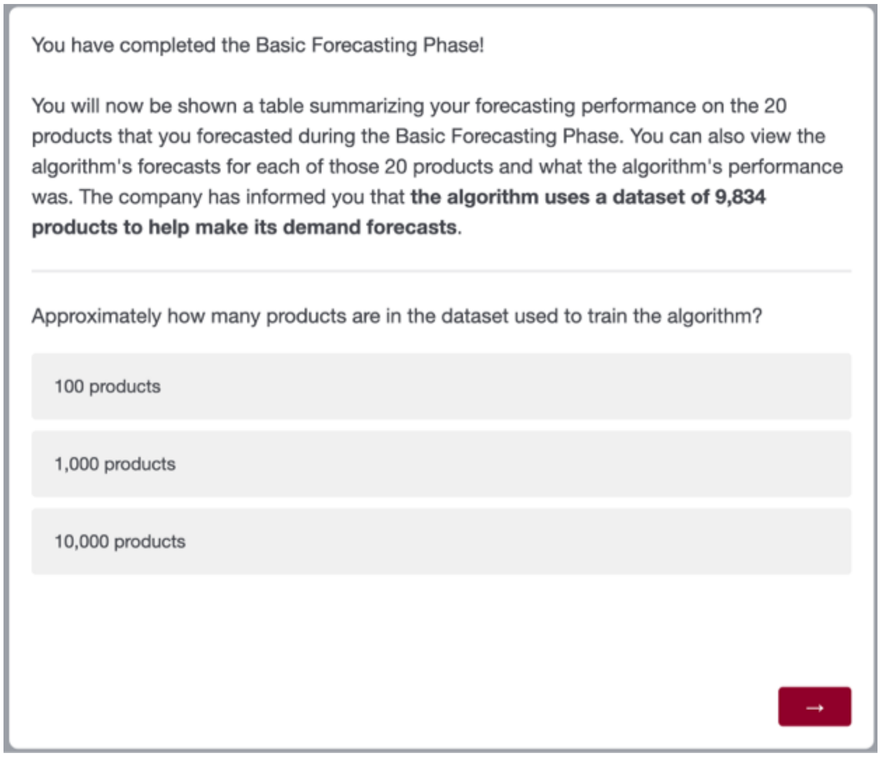
New product 1 (out of 20):

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?

E.2. Training Data Transparency

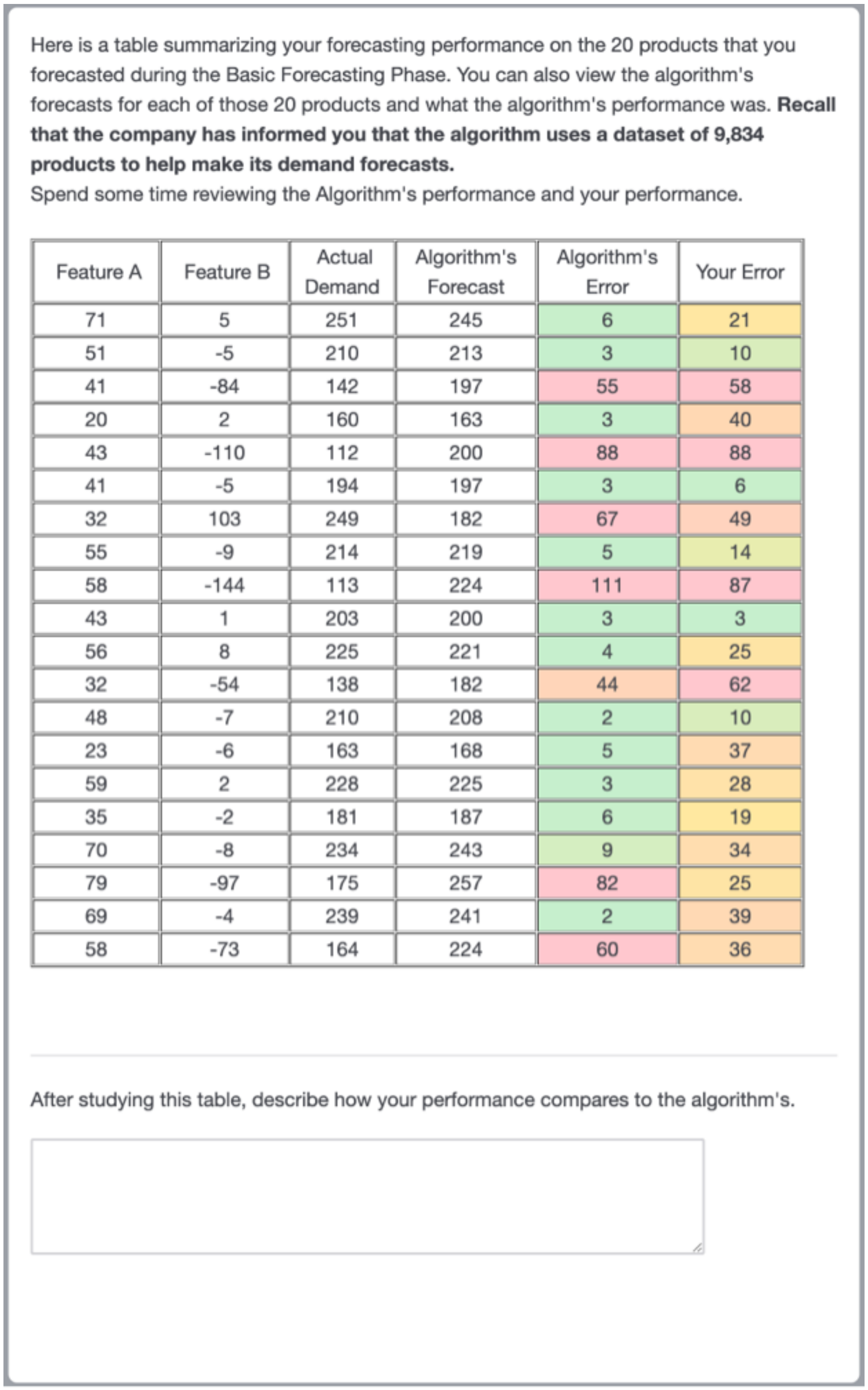


You have completed the Basic Forecasting Phase!

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?

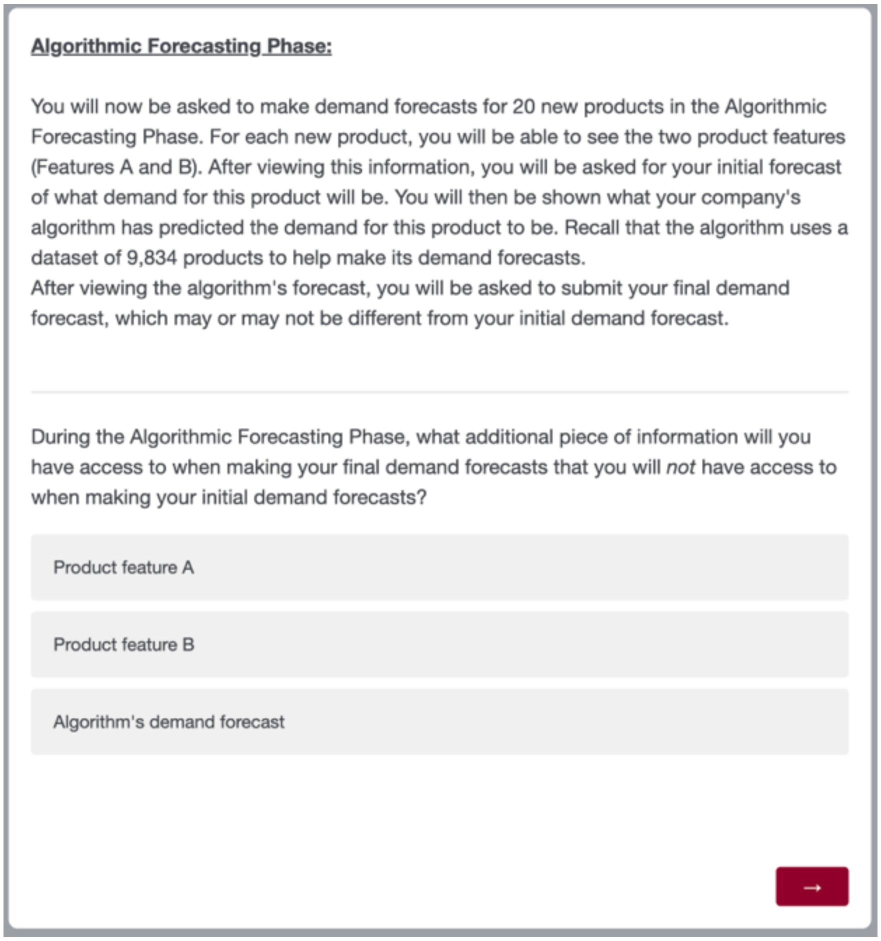
- 100 products
- 1,000 products
- 10,000 products



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:

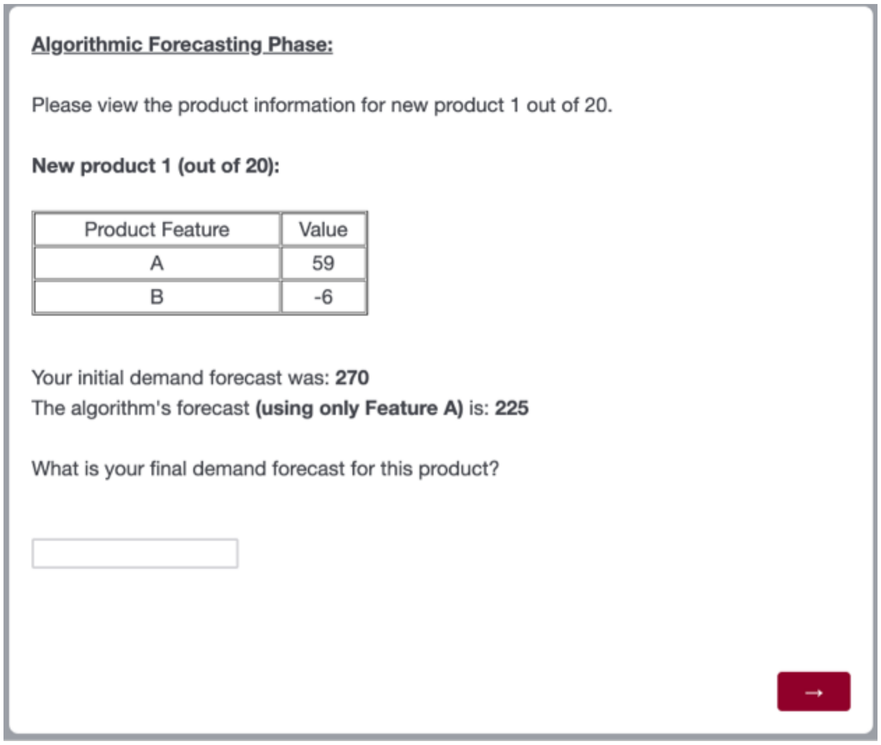
You will now be asked to make demand forecasts for 20 new products in the Algorithmic Forecasting Phase. For each new product, you will be able to see the two product features (Features A and B). After viewing this information, you will be asked for your initial forecast of what demand for this product 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	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?