A RDU analysis

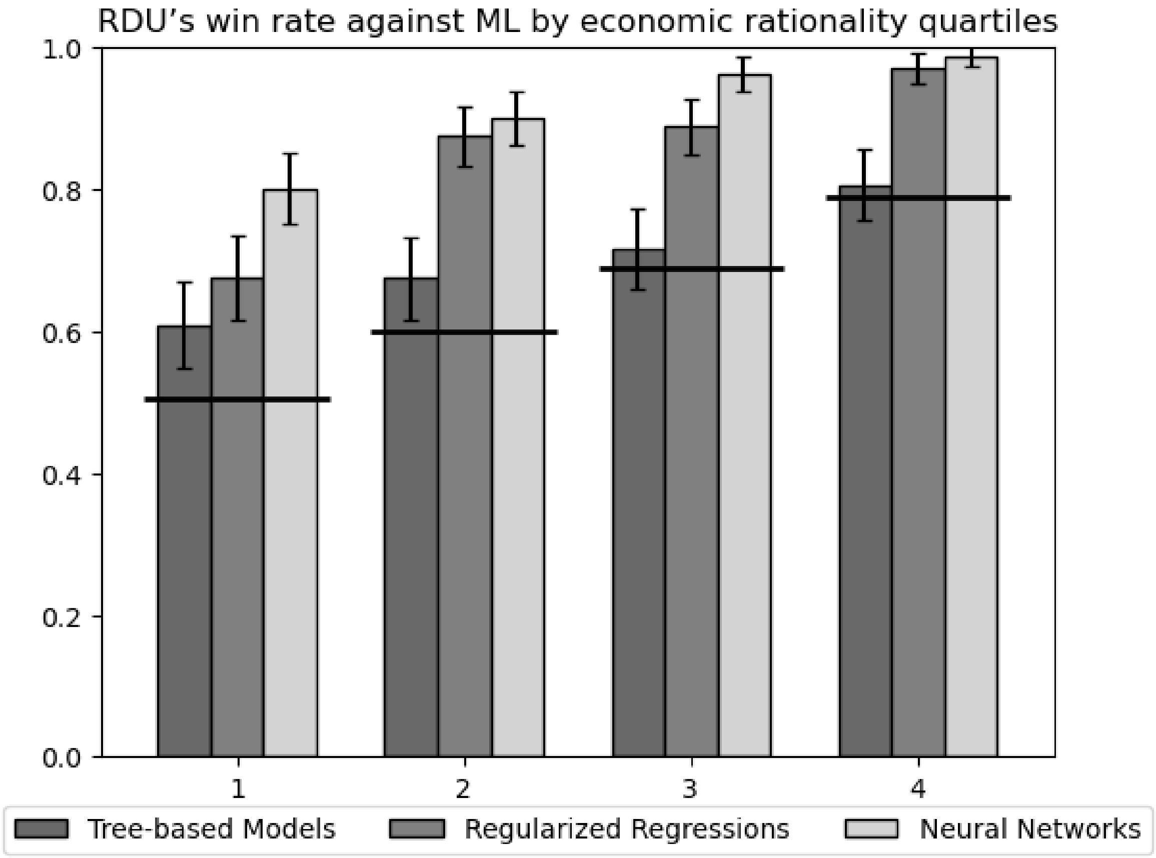
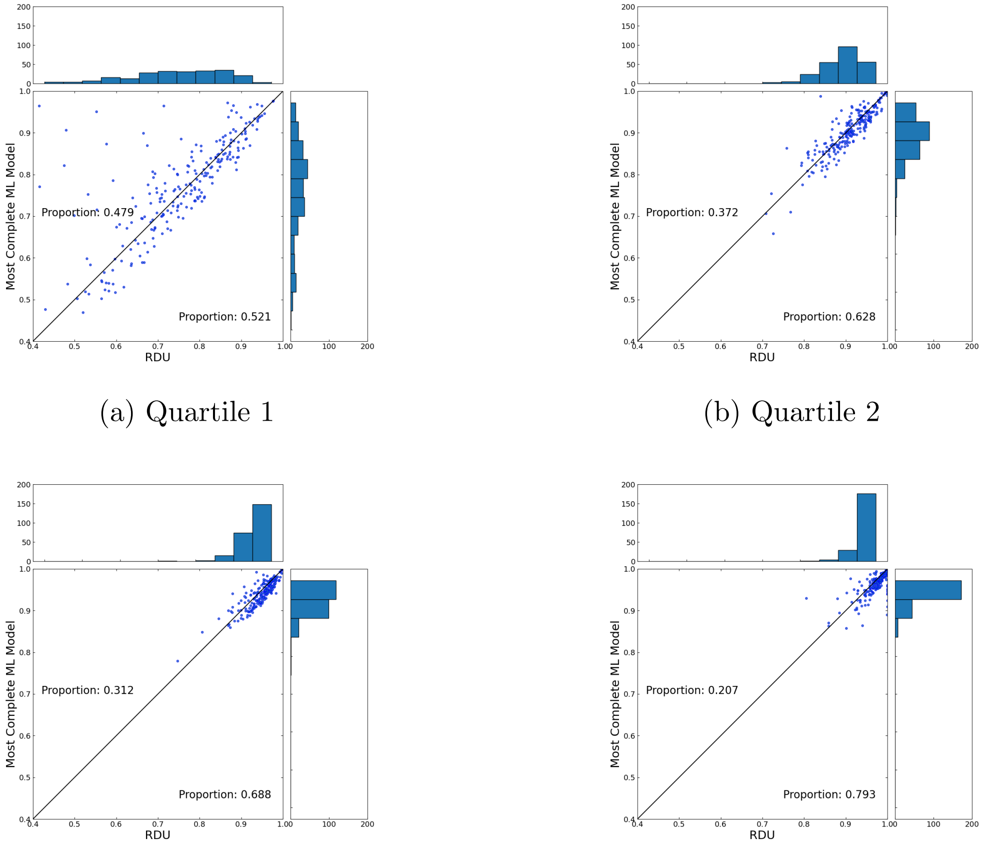


Figure A.1: The fraction of subjects for whom RDU is more complete than the most complete regularized regressions, tree-based, and neural networks model, as well as more complete than the best ML model overall (the horizontal lines), quartiles of consistency scores with GARP and FOSD (Nishimura et al. (2017) and Polisson et al. (2020)). This score measures the amount by which each budget constraint must be relaxed in order to remove all violations of GARP and FOSD and it is bounded between 0 and 1. The closer it is to 1, the smaller the perturbation of budget lines required to remove all violations and thus the closer the data are to satisfying GARP and FOSD. The quartiles are [0, 0.831), [0.831, 0.950), [0.950, 0.988) and [0.988, 1).

			RDU's win rate against ML by rationality quartiles				Absolute completeness difference between RDU and ML by rationality quartiles				
Panel A: RDU and ML model classes	Average completeness	RDU's win rate against ML	1st	2nd	3rd	4th	1st	2nd	3rd	4th	Restrictiveness
RDU	89.2%	-	-	-	-	-	-	-	-	-	16.5%
	[88.3%, 89.9%]										
Regularized Regressions	79.5%	85.1%	67.5%	87.4%	88.8%	97.0%	3.1%	7.5%	9.9%	18.0%	20.6%
	[77.8%, 80.6%]										
Tree-based Models	89.1%	70.1%	60.8%	67.4%	71.7%	80.6%	-2.0%	0.6%	0.7%	0.8%	9.4%
	[88.4%, 89.9%]										
Neural Networks	71.6%	92.6%	79.6%	92.9%	98.8%	99.2%	8.7%	14.4%	16.8%	30.7%	14.3%
	[68.7%, 73.7%]										
Panel B: Regularized Regressions											
Lasso	75.9%	89.6%	77.9%	90.8%	91.7%	98.3%	6.4%	11.5%	14.0%	21.3%	20.6%
	[74.2%, 76.9%]										
OLS	70.2%	87.1%	70.8%	90.0%	90.0%	97.9%	10.6%	10.4%	15.9%	39.0%	20.6%
	[57.5%, 74.7%]										
Ridge	70.6%	87.0%	70.8%	89.5%	90.0%	97.9%	10.5%	10.2%	15.6%	38.1%	20.6%
	[58.2%, 75.0%]										
Panel C: Tree-based Models											
Mean	86.6%	85.9%	77.5%	88.3%	86.3%	91.6%	2.4%	3.5%	2.4%	2.0%	12.3%
	[85.7%, 87.4%]										
Linear	82.9%	86.5%	81.3%	85.8%	87.5%	91.6%	11.8%	5.8%	3.7%	3.6%	5.4%
	[81.6%, 84.0%]										
SVR	85.7%	88.5%	80.4%	90.4%	87.9%	95.4%	3.5%	3.9%	2.9%	3.5%	10.7%
	[84.8%, 86.6%]										
RF	88.0%	79.9%	70.0%	78.7%	80.8%	90.3%	-0.1%	1.5%	1.4%	1.7%	11.9%
	[87.2%, 88.8%]										

Table A.1: The completeness and restrictiveness of RDU and ML models



(a) Quartile 1

(b) Quartile 2

(c) Quartile 3

(d) Quartile 4

Figure A.2: Scatterplot of completeness of RDU and the most complete machine learning model by rationality quartile.

B The experiment

Choi et al. (2007b) developed an experimental graphical interface that allows subjects to make numerous choices over a wide range of budget sets, and this yields a rich dataset that is well-suited to analysis at the level of the individual subject. With the interface, subjects see on a computer screen a geometrical representation of a standard consumer decision problem – the selection of a bundle of commodities from a standard budget set – and choose allocations through a simple "point-and-click." The experiment consisted of 50 independent decision problems.

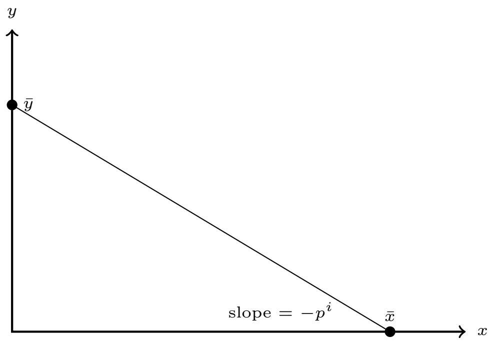


Figure B.3: Example of a Budget Line $\mathcal{B}^i$ with Two States and Two Assets

Figure B.3 showcases an example problem. In each decision problem, a subject was asked to allocate tokens (the experimental currency) between two accounts, labeled x and y. The x account corresponds to the x-axis and the y account corresponds to the y-axis in a two-dimensional graph. Each choice involved choosing a point on a budget line of possible token allocations. Each decision problem started by having the computer select a budget line randomly from the set of lines that intersect at least one axis at or above the 50 token level and intersect both axes at or below the 100 token level.[18]

The payoff at each decision round was determined by the number of tokens in the x account and the number of tokens in the y account. At the end of the round, the computer randomly selected one of the accounts, x or y, determined at random and equally likely. Each subject only received the number of tokens allocated to the account that was chosen. At the end of the experiment, the computer selected one decision round for each participant, revealed the chosen account for that round, and the subject was paid the amount he had earned in that round. Our dataset is comprised of nearly a thousand subjects from several studies including the

[18] The budget lines selected for each subject in their decision problems were independent of each other and of the budget lines selected for other subjects in their decision problems. To choose an allocation, subjects used the mouse to move the pointer on the computer screen to the desired allocation. Choices were restricted to allocations on the budget constraint, so that subjects could not violate budget balancedness.

(symmetric) data collected by Choi et al. (2007a) and data from identical experiments with different subject pools collected by Zame et al. (2020) and Cappelen et al. (2021), as well as new data from identical experiments.[19] In all of these experiments, the individual-level data consist of 50 decision problems.[20] See Choi et al. (2007b) and Choi et al. (2007a) for an extended description of the experimental interface.[21]

[19] Choi et al. (2007a) studied a symmetric treatment, in which the two accounts were equally likely and two asymmetric treatments in which one of the accounts was always selected with probability 1/3 and the other account was selected with probability 2/3.

[20] We do not include the data of Choi et al. (2014) which consist of 25, rather than 50, decision problems. The datasets of Choi et al. (2007a) and Choi et al. (2014) have been analyzed in many papers, including Halevy et al. (2018), Polisson et al. (2020), and De Clippel and Rozen (2021), among others.

[21] The experimental platform is applicable to many other types of individual choice problems. Ahn et al. (2014) extended the earlier experimental work of Choi et al. (2007a) in settings with risk (known probabilities) to settings with ambiguity (unknown probabilities). Fisman et al. (2007), Fisman et al. (2015a), Fisman et al. (2015b), Fisman et al. (2017) and Li et al. (2017), Li et al. (2022) employ a similar experimental methodology to study social preferences across a number of different samples, including a nationally representative sample.

C Revealed preference tests

The most basic question to ask about choice data is whether it is consistent with individual utility maximization. We thus want to relate the out-of-sample prediction accuracy of the economic model, as well as of the ML models, to the consistency of individual behaviors with utility maximization. If budget sets are linear (as in our experiments), classical revealed preference theory (Afriat, 1967; Varian, 1982, 1983) provides a direct test: choices in a finite collection of budget sets are consistent with maximizing a well-behaved (that is, piecewise linear, continuous, increasing, and concave) utility function if and only if they satisfy GARP. Because GARP provides an exact test of utility maximization – either the data satisfy GARP or they do not – but individual choices frequently involve at least some errors, we assess how nearly individual choice behavior complies with GARP by using Afriat (1972) critical cost efficiency index (CCEI), which measures the fraction by which each budget constraint must be shifted in order to remove all violations of GARP. By definition, the CCEI is between 0 and 1: indices closer to 1 mean the data are closer to perfect consistency with GARP and hence to perfect consistency with utility maximization.

But not any consistent preference ordering is admissible. Clearly, choices can be consistent with GARP yet fail to be reconciled with any utility function that is normatively appealing given the decision problem at hand. Given the two states are equally likely, allocating fewer tokens to the cheaper security ($x_s > x_{s'}$ when $p_s < p_{s'}$) is a violation of monotonicity with respect to FOSD. Violations of FOSD are errors – the failure to recognize that some allocations yield payoff distributions with unambiguously lower returns.[22] To test whether choice behavior satisfies GARP and FOSD (for a given subject), we combine the actual data from the experiment and the mirror-image data and compute the CCEI for this combined data set. By definition, the CCEI score for the combined data set consisting of 100 observations can be no bigger than the CCEI score for the actual data. Relying on Nishimura et al. (2017), Polisson et al. (2020) show that when states are equiprobable (as in our experiment), the CCEI score for the combined data set is a measure of consistency with GARP and FOSD for each subject because a well-behaved utility function is monotone with respect to FOSD if and only if it is symmetric.[23]

[22] Almost all decision-theoretic models that have been proposed as alternatives to EUT of which we are aware obey monotonicity with respect to FOSD, including RDU (Quiggin 1982, 1993), Weighted Expected Utility (Dekel 1986; Chew 1989), and CPT (Tversky and Kahneman (1992)). As noted by Quiggin (1990), Wakker and Tversky (1993) and Starmer (2000) prominent non-EUT models, including Prospect Theory (Kahneman and Tversky (1979)), were amended to avoid violations of FOSD.

[23] Clearly, any decision to allocate fewer tokens to the cheaper security (positions along the shorter side of the budget line relative to the 45-degree line) will necessarily generate a simple violation of the Weak Axiom of Revealed Preference (WARP) involving its mirror-image decision.

D Machine learning models

Regularized regressions Regularized regression, in its simplest form, assumes a linear relationship between outcomes and covariates, whose coefficient is estimated using ordinary least squares with a penalty term. Roughly, the penalty term lets the model "learn" which variables are important, and which to ignore. While including a penalty biases the coefficients, doing so also reduces the chance of overfitting, or "chasing noise." We consider two popular models of regularized regression that add the norm of the coefficient vector as the penalty. The two differ in which norm is implemented as the penalty. First, we consider Lasso (Tibshirani (1996)), which penalizes using the L_1 norm. Formally, estimating relative demand using Lasso generates a mapping $\hat{f}_{Lasso}$:

$$\hat{f}_{Lasso}(\mathcal{B}) = \hat{\beta}^T \mathcal{B},$$

where $\hat{\beta}$ solves

$$\hat{\beta} = \text{argmin}_{\beta} \sum_{i=1}^{50} (\mathbf{x}^i - \beta^T \mathcal{B}^i)^2 + \lambda \mid\mid \beta \mid\mid_1$$

Second, we consider ridge regression (Hoerl and Kennard (1970)), which penalizes using the L_2 norm. Formally, estimating relative demand using Ridge generates a mapping $\hat{f}_{Ridge}$:

$$\hat{f}_{Ridge}(\mathcal{B}) = \hat{\beta}^T \mathcal{B},$$

$$\hat{\beta} = \text{argmin}_{\beta} \sum_{i=1}^{50} (\mathbf{x}^i - \beta^T \mathcal{B}^i)^2 + \lambda (\mid\mid \beta \mid\mid_2)^{1/2}$$

The parameter λ affects the degree to which the size of β affects the objective function. If $\lambda = 0$, then the optimization is OLS. We use leave-one-out cross-validation to determine the parameter $\lambda \in [0, 0.2, 0.4, 0.6, 0.8, 1]$. The budget set $\mathcal{B}^i$ is encoded as an intercept $1/p_1$ and the price ratio p_2/p_1. The parameter vector θ for regularized regressions models is a linear coefficient vector.

Tree-based Let t denote one of the possible variables associated with a budget set. Unlike the linear relationship assumed in regularized regression, tree-based models divide the set of budget sets $\boldsymbol{B}$ into subsets (based on the prices and the endowment) and estimate a model on each of the subsets. This division is done recursively. That is, given some index of observations Z corresponding to data $\{(\mathcal{B}^i, \mathbf{x}^i)\}_{i \in Z}$, the the algorithm considers all further binary partitions that can be represented as separating data based on a variable x being above or below a given threshold k: $\{(\mathcal{B}^i, \mathbf{x}^i)\}_{i \in Z \text{ and } t^i \leq k}$ and $\{(\mathcal{B}^i, \mathbf{x}^i)\}_{i \in Z \text{ and } t^i > k}$. Of these partitions, the selected partition is the (t, k) pair that minimizes error when applying optimal models to each partition.

$$(t^*, k^*) \in \text{argmin}_{(t,k)} \left\{ \sum_{i:i\in Z, t^i \leq k} \ell\left[f_{\theta}^{\leq}(\mathcal{B}^i), \mathbf{x}^i\right] + \sum_{i:i\in Z, t^i > k} \ell\left[f_{\theta}^{>}(\mathcal{B}^i), \mathbf{x}^i\right] \right\},$$

where $f_{\theta}^{\leq} = \text{argmin}_{f\in\mathcal{F}_\Theta} \sum_{i:i\in Z, t^i\leq k} \ell(f(\mathcal{B}^i, \mathbf{x}^i))$ and $f_{\theta}^{>} = \text{argmin}_{f\in\mathcal{F}_\Theta} \sum_{i:i\in Z, t^i> k} \ell(f(\mathcal{B}^i, \mathbf{x}^i))$.

The process is then reapplied for the two subsets of the resulting partition, and so on. This partitioning process generates both the (locally) best partition of budget sets and the (locally) best model estimate for the partition. In aggregate, the algorithm returns a piecewise demand function. To predict the relative demand of some budget set $\mathcal{B}^i$, first the subset containing $\mathcal{B}^i$ determines which model to use. Then, evaluating that model determines the demand.

This partitioning process, if allowed to continue without restraint, would end with each data point in its own partition, with perfect within-sample prediction. To prevent such overfitting, we limit the decision trees in two simple ways. First, we set a minimum number of observations per partition. This prevents the algorithm from splitting a partition if doing so would result in an insufficiently large sample size. Second, we limit the "depth", or number of partitions away from $\boldsymbol{B}$, of a tree. These limits are determined endogenously for each subject by performing 3-fold cross validation. In this procedure, data is randomly split into three equally sized subsamples. We choose the maximum depth to search over 2, 4, 6, and 8; we choose the minimum observations per partition to search over 2, 4, 6, 8, and 10.

The standard decision tree model, denoted Mean, takes the sample mean token share x of each subset. We use Mean as well as three extensions. The first extension, known more broadly as model trees (Quinlan et al. (1992)), changes the estimated model from a sample mean to a linear regression (Linear). Mean is nested in Linear. The second extension, support vector regression trees, instead uses a support vector regression of each subset. Support vector regression attempts to find the flattest demand mapping possible such that the token share predictions are accurate up to some $\epsilon \geq 0$ (see Smola and Schölkopf (2004)). The last tree-based model, the random forest model (RF) averages the decision rules of multiple standard decision trees. Each tree is given a bootstrapped data set, and is generally seen as an improvement over singular decision trees (Breiman (2001)). In addition to limits on depth and minimum sample size, RF regulates the number of trees, which we choose to be 10, 50, and 100 trees. Because each tree not trained on the original data set, there is no nesting and thus no restrictiveness or completeness guarantees between RF and the other tree-based models. Additionally, since trees are inherently nonparametric, they cannot be easily described by a parameter vector θ.

Neural networks Neural networks, specifically a multilayer perceptron, transform budget sets into relative demand predictions by nonlinear regression, whose functional form assumes a series of nested transformations. In our setup, the transformation takes two parts. First, a budget set $\mathcal{B}$ undergoes an affine transformation $W^{(0)}\mathcal{B}+b^{(0)}$, where $W^{(0)}$ and $b^{(0)}$ are a matrix and vector of size $n_0 \times 2$ and $n_0 \times 1$, respectively. The dimension n_0 is prespecified by the analyst. Second, the affine transformation is again transformed by a function $\sigma^{(0)} : \mathbb{R}^{n_0} \to \mathbb{R}^{n_0}$ to obtain a new vector $\mathcal{B}^{(1)} = \sigma^{(0)}(W^{(0)}\mathcal{B}+b^{(0)})$. The function σ is also prespecified by the analyst. The resulting vector, $\mathcal{B}^{(1)}$, is referred to as a "hidden layer". It is then used as the input to generate another hidden layer, $\mathcal{B}^{(2)} = \sigma^{(1)}(W^{(1)}\mathcal{B} + b^{(1)})$, using a new affine transformation defined by $\underset{n_1\times n_0}{W^{(1)}}$ and $\underset{n_1\times 1}{b^{(1)}}$ as well as transformation by $\sigma^{(1)}$. This process continues for the number of hidden layers prespecified by the analyst. The final affine transformation results in a scalar value that can be interpreted as the estimated relative demand.

For a multilayer perceptron, the parameter values $W^{(i)}$ and $b^{(i)}$ are estimated, while the analyst has the freedom to choose the number of layers, the dimensions of each layer, the $\sigma^{(i)}$ functions, and a number of parameters associated with the estimation of $W^{(i)}$ and $b^{(i)}$.

We use the layer count, layer dimension, and $\sigma^{(i)}$ values from Zhao et al. (2020). $\sigma^{(i)}$ are all chosen to be the same component-wise maximum function $\sigma(x) = \max(0, x)$. This function, the rectified linear unit ("ReLU") function, keeps all positive components of a vector, and sets all negative components to zero. We use 3-fold cross-validation to simultaneously determine the individual-best layer count and layer dimension. We search over all combinations of $\{1, 2, 3\}$ hidden layers, as well as all combinations of $\{15, 20, 25\}$ for the size of each layer, for a total of 39 "architectures" investigated.

We use the L-BFGS algorithm to estimate $W^{(i)}$ and $b^{(i)}$ (for a full treatment, see Bottou et al. (2018) and Sun et al. (2019)). This algorithm is readily available in software packages such as Python's `sklearn`. The estimation objective function to be minimized is mean squared error, which is the same objective function used to evaluate all models (through completeness and restrictiveness). For example, given a network of 2 hidden layers each with dimension 15, the objective function is:

$$\min_{\underset{15\times 2}{W^{(0)}},\underset{15\times 15}{W^{(1)}},\underset{1\times 15}{W^{(2)}},\underset{15\times 1}{b^{(0)}},\underset{15\times 1}{b^{(1)}},\underset{1\times 1}{b^{(2)}}} \sum_{x^i \in \mathcal{D}} \ell(f(W,b),\mathbf{x}^i) = \left[\mathbf{x}^i - W^{(2)}\sigma\left(W^{(1)}\sigma\left(W^{(0)}\mathcal{B}^i + b^{(0)}\right) + b^{(1)}\right) - b^{(2)}\right]^2$$

Let w denote a vectorized version of W and b. The estimation method is quasi-Newtonian, and involves iteratively updating parameters in the direction of the gradient of the loss function with respect to the parameters w, $\nabla \mathrm{L}(w)$. The general form is

$$w_{k+1} \leftarrow w_k - \alpha_k H_k \nabla \mathrm{L}(w),$$

where w_k denotes the k^{th} iteration of updating w, α_k is a step-size parameter value chosen to satisfy

$$\min_{\alpha} L\left(w_k - \alpha H_k \nabla \mathrm{L}(w)\right),$$

and H_k is an updating estimate of the inverse of the Hessian matrix $\nabla^2 \mathrm{L}(w)$. Let $s_k = w_{k+1} - w_k$ and $v_k = \nabla \mathrm{L}(w_{k+1}) - \nabla \mathrm{L}(w_k)$. Then,

$$H_{k+1} \leftarrow \left(I - \frac{v_k s_k^T}{s_k^T v_k}\right)^T H_k \left(I - \frac{v_k s_k^T}{s_k^T v_k}\right) + \frac{s_k s_k^T}{s_k^T v_k}$$

E Asymmetric probability

We investigate a more complex environment, using data with asymmetric probabilities of 1/3 and 2/3 but otherwise identical experiment and analysis, on 46 subjects who participated in the original Choi et al. (2007) study. In the asymmetric environment where $P(s_x) = 1/3$ and $P(s_y) = 2/3$, RDU corresponds to

$$\text{RDU}(x, y; \rho) = \begin{cases} w(1/3) \cdot u(x) + [1 - w(1/3)] \cdot u(y) & x < y \\ w(2/3) \cdot u(y) + [1 - w(2/3)] \cdot u(x) & x > y \end{cases}$$

To economize on notation, let $w_1 := w(1/3)$ and $w_2 := w(2/3)$. Optimal demand results in checking the maximum utility of two cases: where $x > y$ and where $x < y$. For both cases, x is solved for and $y = \frac{1-p_x \cdot x}{p_y}$.

For CRRA utility $u(x) = \frac{x^{1-\rho}}{1-\rho}$ (and $\log(x)$ when $\rho = 1$), assuming $x < y$, the optimal solution is

$$x^*_{lower} = \begin{cases} \frac{1}{p_x + p_y \left[\frac{w_1}{1-w_1} * \frac{p_y}{p_x}\right]^{-1/\rho}} & w_1 < \frac{p_x}{p_x+p_y} \\ \frac{1}{p_x+p_y} & w_1 \geq \frac{p_x}{p_x+p_y} \end{cases}$$

Assuming $x > y$, the optimal solution is

$$x^*_{upper} = \begin{cases} \frac{1}{p_x + p_y \left[\frac{1-w_2}{w_2} * \frac{p_y}{p_x}\right]^{-1/\rho}} & w_2 < \frac{p_y}{p_x+p_y} \\ \frac{1}{p_x+p_y} & w_2 \geq \frac{p_y}{p_x+p_y} \end{cases}$$

The optimal demand $x^*_{CRRA} = \arg\max_{x \in \{x^*_{lower}, x^*_{upper}\}} \text{RDU}(x, \frac{1-p_x \cdot x}{p_y}; \rho)$.

For CARA utility $u(x) = e^{-Ax}$, $x \geq 0$, assuming $x < y$, the optimal solution is

$$x^*_{lower} = \begin{cases} 0 & w_1 \leq \frac{p_x}{p_x+p_y \cdot \exp(A/p_y)} \\ \frac{1}{p_x+p_y} - \frac{\log\left(\frac{1-w_1}{w_1} \cdot \frac{p_x}{p_y}\right)}{A \cdot \left(1+\frac{p_x}{p_y}\right)} & w_1 \in [\frac{p_x}{p_x+p_y \exp(A/p_y)}, \frac{p_x}{p_x+p_y}] \\ \frac{1}{p_x+p_y} & w_1 \geq \frac{p_x}{p_x+p_y} \end{cases}$$

Assuming $x > y$, the optimal solution is

$$x^*_{upper} = \begin{cases} \frac{1}{p_x+p_y} & w_2 \geq \frac{p_y}{p_x+p_y} \\ \frac{1}{p_x+p_y} - \frac{\log\left(\frac{w_2}{1-w_2} \cdot \frac{p_x}{p_y}\right)}{A \cdot \left(1+\frac{p_x}{p_y}\right)} & w_2 \in [\frac{p_y}{p_x \cdot \exp(A/p_x)+p_y}, \frac{p_y`}{p_x+p_y}] \\ \frac{1}{p_x} & w_2 \leq \frac{p_y}{p_x \cdot \exp(A/p_x)+p_y} \end{cases}$$

The optimal demand $x^*_{CARA} = \arg\max_{x \in \{x^*_{lower}, x^*_{upper}\}} \text{RDU}(x, \frac{1-p_x \cdot x}{p_y}; \rho)$.

Replications of Tables 1, 2, and A.1, along with Figure 1, are reported below. Overall, the results are fairly robust: RDU is more complete than any individual model, although EUT has slightly reduced completeness. Additionally, RDU's win rate against machine learning models is increasing by rationality quartile, although the fewer subjects gives noisier results.

			RDU's win rate against ML by rationality quartiles				Absolute completeness difference between RDU and ML by rationality quartiles				
Panel A: RDU and ML model classes	Average completeness	RDU's win rate against ML	1st	2nd	3rd	4th	1st	2nd	3rd	4th	Restrictiveness
RDU	90.1% [87.8%, 91.8%]	-	-	-	-	-	-	-	-	-	13.5%
Regularized Regressions	84.2% [80.5%, 86.9%]	84.8%	72.7%	75.0%	91.7%	100.0%	5.4%	2.6%	4.6%	11.5%	22.1%
Tree-based Models	90.2% [88.3%, 91.9%]	54.3%	27.3%	75.0%	50.0%	63.6%	-2.0%	0.9%	0.2%	0.1%	8.1%
Neural Networks	81.8% [78.0%, 84.6%]	91.3%	81.8%	91.7%	91.7%	100.0%	11.2%	3.8%	9.6%	8.6%	15.9%
Panel B: Regularized Regressions											
Lasso	75.8% [72.4%, 78.9%]	95.7%	81.8%	100.0%	100.0%	100.0%	11.5%	12.1%	13.7%	20.1%	22.1%
OLS	81.8% [71.6%, 85.8%]	87.0%	81.8%	75.0%	91.7%	100.0%	13.3%	2.7%	4.7%	13.3%	22.1%
Ridge	82.2% [72.7%, 86.0%]	87.0%	81.8%	75.0%	91.7%	100.0%	12.4%	2.6%	4.6%	12.8%	22.1%
Panel C: Tree-based Models											
Mean	89.4% [87.3%, 91.2%]	71.7%	45.5%	83.3%	75.0%	81.8%	-0.4%	1.4%	0.7%	0.8%	10.1%
Linear	88.0% [85.7%, 90.0%]	80.4%	54.5%	91.7%	91.7%	81.8%	1.2%	3.3%	2.1%	1.6%	10.7%
SVR	86.2% [83.3%, 88.4%]	84.8%	90.9%	83.3%	75.0%	90.9%	5.7%	3.9%	2.1%	4.3%	7.0%
RF	86.3% [84.0%, 88.3%]	84.8%	72.7%	100.0%	83.3%	81.8%	2.7%	5.6%	3.2%	3.5%	9.0%

Table E.3: The completeness and restrictiveness of RDU and ML models

Table E.3: The completeness and restrictiveness of EUT and RDU

			EUT's win rate against RDU by rationality quartiles				Absolute completeness difference between EUT and RDU by rationality quartiles				
Panel A: EUT and RDU	Average completeness	EUT win rate against RDU	1st	2nd	3rd	4th	1st	2nd	3rd	4th	Restrictiveness
EUT	83.5% [79.6%, 86.4%]	50.0%	50.0%	50.0%	50.0%	50.0%	0.0%	0.0%	0.0%	0.0%	30.2%
RDU	90.1% [87.8%, 91.8%]	17.4%	27.3%	16.7%	8.3%	18.2%	-6.8%	-4.8%	-6.6%	-8.4%	13.5%
Panel B: CRRA Only											
EUT CRRA	82.3% [78.5%, 85.3%]	50.0%	50.0%	50.0%	50.0%	50.0%	0.0%	0.0%	0.0%	0.0%	31.2%
RDU CRRA	89.6% [87.1%, 91.4%]	19.6%	27.3%	16.7%	16.7%	18.2%	-6.4%	-6.1%	-6.9%	-9.9%	13.2%
Panel C: CARA Only											
EUT CARA	82.6% [78.2%, 85.6%]	50.0%	50.0%	50.0%	50.0%	50.0%	0.0%	0.0%	0.0%	0.0%	29.6%
RDU CARA	89.2% [86.9%, 90.9%]	15.2%	27.3%	16.7%	8.3%	9.1%	-7.5%	-4.5%	-6.4%	-8.2%	13.8%

Table E.3: The completeness and restrictiveness of EUT and ML models

	Average	EUT's win rate	EUT's win rate against ML by rationality quartiles				Absolute completeness difference between EUT and ML by rationality quartiles				
Panel A: EUT and ML model classes	Completeness	against model	1st	2nd	3rd	4th	1st	2nd	3rd	4th	Restrictiveness
EUT	83.5% [79.6%, 86.4%]	-	-	-	-	-	-	-	-	-	30.2%
Regularized Regressions	84.2% [80.5%, 86.9%]	50.0%	72.7%	33.3%	25.0%	72.7%	-1.4%	-2.2%	-2.0%	3.0%	22.1%
Tree-based Models	90.2% [88.3%, 91.9%]	19.6%	18.2%	16.7%	25.0%	18.2%	-8.7%	-3.9%	-6.5%	-8.3%	8.1%
Neural Networks	81.8% [78.0%, 84.6%]	54.3%	63.6%	33.3%	75.0%	45.5%	4.5%	-0.9%	2.9%	0.1%	15.9%
Panel B: Regularized regressions											
Lasso	75.8% [72.4%, 78.9%]	87.0%	90.9%	91.7%	75.0%	90.9%	4.7%	7.3%	7.1%	11.7%	22.1%
OLS	81.8% [71.6%, 85.8%]	52.2%	72.7%	33.3%	25.0%	81.8%	6.5%	-2.1%	-1.9%	4.8%	22.1%
Ridge	82.2% [72.7%, 86.0%]	52.2%	72.7%	33.3%	25.0%	81.8%	5.6%	-2.2%	-2.0%	4.3%	22.1%
Panel C: Tree-based models											
Mean	89.4% [87.3%, 91.2%]	23.9%	27.3%	25.0%	25.0%	18.2%	-7.2%	-3.3%	-6.0%	-7.6%	10.1%
Linear	88.0% [85.7%, 90.0%]	43.5%	45.5%	50.0%	33.3%	45.5%	-5.6%	-1.5%	-4.5%	-6.9%	10.7%
SVR	86.2% [83.3%, 88.4%]	47.8%	63.6%	41.7%	33.3%	54.5%	-1.1%	-0.9%	-4.6%	-4.2%	7.0%
RF	86.3% [84.0%, 88.3%]	52.2%	45.5%	58.3%	41.7%	63.6%	-4.0%	0.8%	-3.5%	-5.0%	9.0%

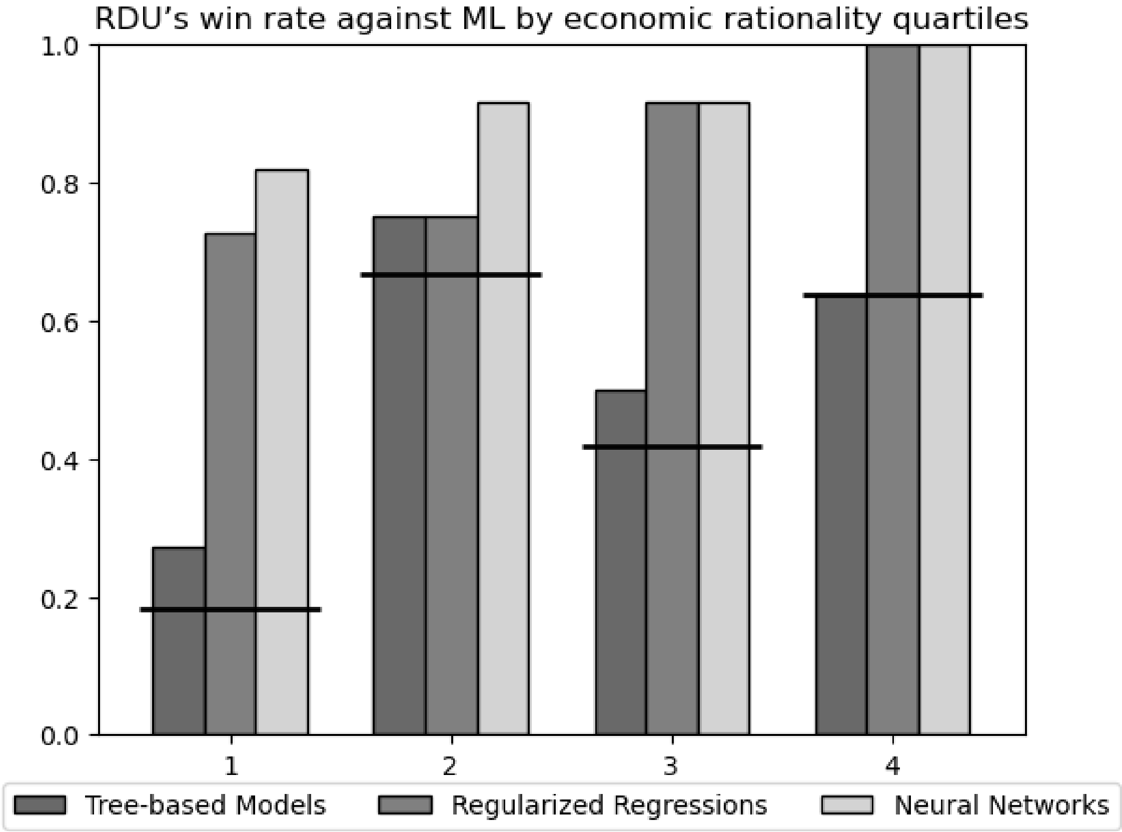


Figure E.4: The fraction of subjects for whom RDU is more complete than the most complete ML models, 2D Asymmetric.

F Simulated agents with logit noise

To observe the effects of increased data on model performance, we simulate agents with CRRA Bernoulli utility $u(x) = \frac{x^{1-\rho}}{1-\rho}$ with parameter $\rho = 0.5$. Each agent makes choices with noise, with the probability of a specific allocation $\mathbf{x}$ being chosen from a budget set with prices $\mathbf{p}$ according to a logistic distribution

$$P(\mathbf{x}) = \frac{\exp\left[\gamma \mathbb{E}u(\mathbf{x})\right]}{\int_{\mathbf{x'}|\mathbf{px'}=1} \exp\left[\gamma \mathbb{E}u(\mathbf{x'})\right]}$$

As γ approaches zero, the distribution approaches uniform random choice over the budget line. As γ approaches infinity, the distribution approaches deterministic utility maximization. We simulate 1000 choices from $\gamma \in \{0.25, 0.5, 1, 5, 10\}$, and calculate completeness estimates for each simulation, with random uniform choice over the budget line as the naive model and perfect prediction as the irreducible error.[24]

Because of the changed size of the data set, we modify the hyperparameter values of tree-based models to handle the increased size. We extend the minimum depth and minimum observations per partition to search over 2, 4, 6, 8, 10, 12, 14, 16, 18, and 20 for both hyperparameters. Note that linear and support vector regression trees are omitted due to computational limitations.

Table F.4 shows the results. Overall, the completeness of EUT and RDU slightly outperform the best ML algorithm at all levels of noise. As somewhat expected, the performance gap is reduced when introducing orders of magnitude more data.

Table F.4: The completeness of economic and machine learning models for simulated agents with 1000 choices.

			Regularized	Tree-based	Neural	
γ	EUT	RDU	Regressions	Models	Networks	Best ML
0.25	68.6%	68.5%	57.6%	67.9%	67.6%	67.9%
0.5	74.2%	74.1%	60.3%	73.4%	73.7%	73.7%
1	79.5%	79.5%	64.6%	79.3%	78.5%	79.3%
5	95.0%	94.9%	71.8%	94.8%	94.6%	94.8%
10	97.0%	97.0%	71.9%	96.9%	96.7%	96.9%

[24] We can also use a noiseless model as the irreducible error. This would serve to only increase the completeness of each model, as the noiseless model will make errors in prediction. However, the order of model performances will not be affected.

G Lowest quartile of consistency analysis

We further investigate subjects in the lowest quartile of consistency, to determine whether behavioral inconsistencies are due to either randomness or systematic violations of GARP and FOSD. Randomness in responses can generally be viewed as mistakes; in the most extreme cases, subjects would be no more consistent with CCEI than a Bronars (1987) test. For subjects that have systematic violations, consider subjects allocating all tokens to one security regardless of price, which is consistent with GARP but not with FOSD, or always allocating all tokens to the more expensive security (violating both GARP and FOSD). In the case of pure randomness, we expect model performance to be approximately equally poor, as the out-of-sample data is uncorrelated with training data. However, in the case of systematic and regular violations out of the scope of standard economic models that satisfy GARP and FOSD, we expect machine learning model performance to be better than economic models.

Figure G.5 plots completeness of EUT and ML for each subject, additionally fitting polynomials of degree three to the scatterplot for both EUT and ML. The three vertical lines within the image denote the cutoffs between quartiles of consistency scores. We emphasize that while the lowest quartile visually takes up more space in the figure, it still contains the same number of subjects as the other quartiles. For the three highest quartiles, there is a close relationship between consistency scores and the performances of EUT and ML. However, when considering subjects whose consistency is less than 0.5, the completeness of ML is estimated to be consistently higher than that of EUT. This corresponds to 59 subjects, or approximately 6.2% of our subject pool. For these subjects, we reject the notion that choices are due to randomness.

First, we examine these subjects in the aggregate. The average completeness among these subjects is 68.7% for ML and 59.2% for EUT. Additionally, lower consistency scores are associated with higher numbers, not just magnitudes, of WARP and FOSD violations: subjects below 0.5 have on average 14 FOSD violations and 49 WARP violations, compared to an average of 3 FOSD violations and 7 WARP violations for subjects above 0.5.

Second, we examine the three subjects with the highest difference in completeness between ML and EUT. The average completeness among these subjects is 97.8% for ML and 28.4% for EUT. Additionally, all subjects belong to the lowest quartile of rationality scores: the average consistency score is 0.167. Demand curves are shown in Figure G.6.

It is easy to visually classify the three subjects as subscribing to a simple decision rule, yet their rules are mistakes. For subjects ID 66 and 81, security 2 is always (nearly) fully invested in, regardless of the price. These decision rules are nearly consistent with utility maximization. Unlike other subjects below consistency score of 0.5, ID 66 has 1 WARP violation and ID 81 has 0 WARP violations. For example, the

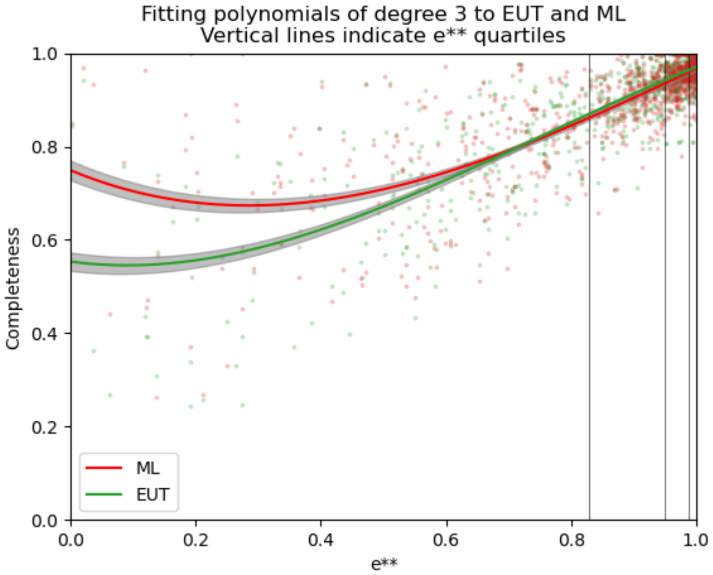


Figure G.5: Completeness and rationality score of all subjects. Polynomials of degree three are fit to the scatterplot for both EUT and ML.

non-expected utility function $u(x_1, x_2) = x_2$ will rationalize these choices. However, such choices are indeed mistakes due to the nature of the goods. Because the goods associated with the experiment are Arrow securities, any choice allocating more to the more expensive state than the cheaper state violates monotonicity with respect to FOSD. Therefore, any choice such that $\log(p_1/p_2) < 0$ and $x_1/(x_1 + x_2) < 0.5$ violates FOSD. Unsurprisingly, ID 66 has 23 FOSD violations and ID 81 has 27 FOSD violations.

Subject ID 221 exclusively chooses the more expensive of the goods if prices are sufficiently different, and equates the two goods when prices are similar. Like ID 66 and ID 81, this rule again consistently violates FOSD. However, it also consistently violates WARP, as shown in Figure G.7. In the figure, two actual price configurations of Subject ID 221 and associated decisions are plotted: round 12 in black and round 25 in blue. The log price ratio for the configurations are nearly identical in magnitude, yet with opposite sign. Choosing the more expensive alternative generates a cycle in revealed preference. Both points are available under the black price configuration, implying that the black point is preferred to the blue point, and vice versa when considering revealed preference from the blue price configuration. Subject ID 221 generates a total of 224 WARP violations, the most of any subject in the sample.

Overall, among the subjects considered, the rule explaining behavior is simple, yet they violate basic principles of utility maximization in a way that does not appear to be solely due to higher levels of randomness. Additionally, the behavior does not appear to stem from a behavioral bias, but potentially instead as a mistake or

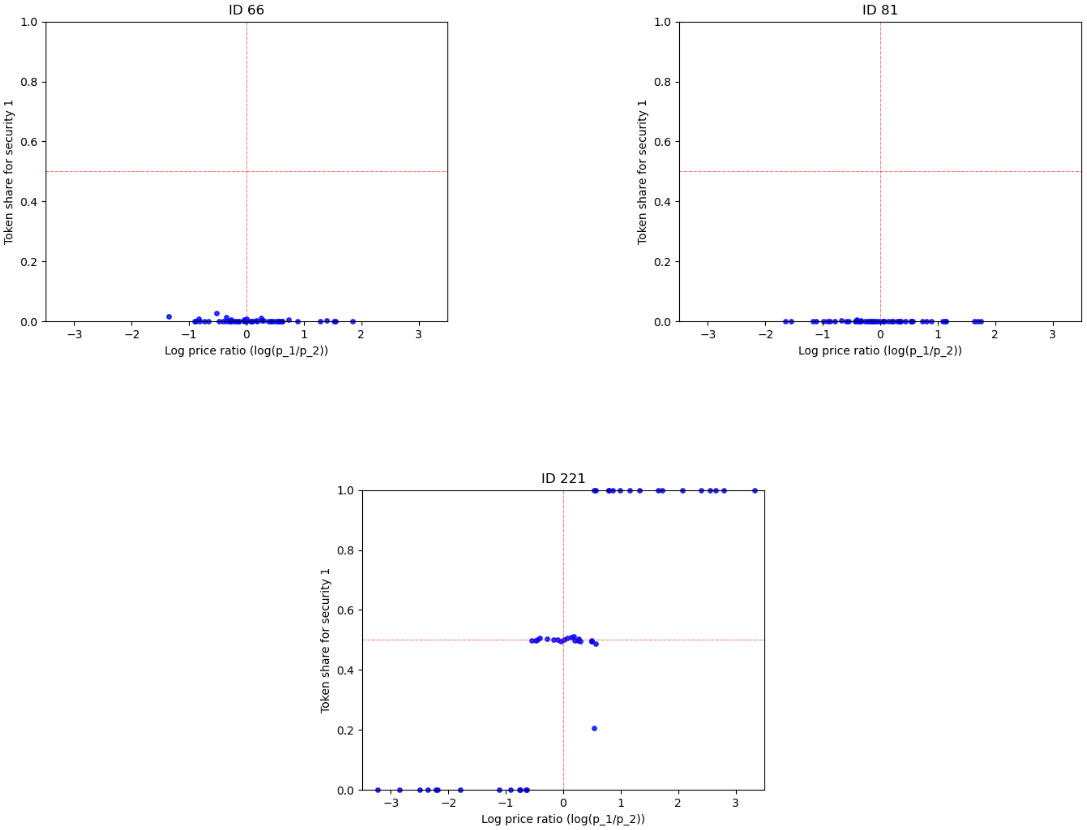


Figure G.6: The relationship between the log-price ratio and the token share for subjects with the highest completeness differential between ML and EUT, in favor of ML.

misunderstanding of the experiment.

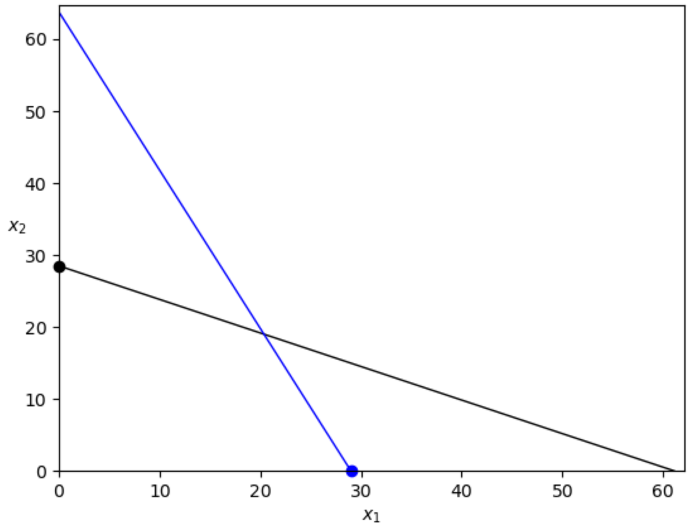


Figure G.7: Price configurations and choices of rounds 12 (black) and 25 (blue) for subject ID 221.

H Significant completeness differences

We address statistical uncertainty regarding win rates by calculating individual-level completeness standard errors and conducting a two-sample t-test. The estimates of standard errors come from Fudenberg et al. (2022a). Let k denote an arbitrary fold of cross-validation, and let $k(i)$ be a function mapping data observations to folds. For a model $\mathcal{F}_\Theta$, define the average fold k error as

$$\bar{\Delta}_{\Theta,k} = \frac{1}{5} \sum_{k(i)=k} \ell \left[f^*_\Theta(\mathcal{B}^i), x^i \right]$$

The fold k sample variance is then defined as

$$\hat{\sigma}^2_{\Delta_{\Theta,k}} = \frac{1}{4} \sum_{k(i)=k} \left[\ell \left[f^*_\Theta(\mathcal{B}^i), x^i \right] - \bar{\Delta}_{\Theta,k} \right]^2$$

The average sample variance across folds is

$$\hat{\sigma}^2_{\Delta_\Theta} = \frac{1}{10} \sum_{k=1}^{10} \hat{\sigma}^2_{\Delta_{\Theta,k}}$$

Define analogous measures $\bar{\Delta}_{f_{naive},\ k}$, $\hat{\sigma}^2_{f_{naive},\ k}$, and $\hat{\sigma}^2_{f_{naive}}$ for the naive random uniform decision rule.

The covariance estimator is defined as

$$\hat{\sigma}_{\Delta_\Theta \Delta_{f_{naive}}} = \frac{1}{10}\sum_{k=1}^{10}\frac{1}{4}\sum k(i) = k\left[\ell\left[f^*_\Theta(\mathcal{B}^i), x^i\right] - \bar{\Delta}_{\Theta,k}\right]\left[\ell\left[f_{naive}(\mathcal{B}^i), x^i\right] - \bar{\Delta}_{naive,k}\right]$$

The variance estimator for completeness is then

$$\hat{\sigma}^2_\kappa = \frac{\hat{\sigma}^2_{\Delta_\Theta} - 2\hat{\kappa}\hat{\sigma}_{\Delta_\Theta \Delta_{f_{naive}}} + \hat{\kappa}^2\hat{\sigma}^2_{f_{naive}}}{\left[\hat{\mathcal{E}}_{CV}(f^*_\Theta)\right]^2},$$

where $\hat{\mathcal{E}}_{CV}(f_\Theta)$ is the cross-validated mean squared error for f^*_Θ and $\hat{\kappa}$ is the estimate of completeness.

We naively assume independence between the completeness distributions – while it is clear that the model estimates are correlated because they are evaluated on the same data set, this bias should increase the number of subjects with "significant wins" due to the assumed covariance of zero lowering the estimate of standard error. Of the 956 subjects in the data set, only 13 have significant differences in completeness, even with the naive independence assumption.

However, the main result that RDU outperforms economic models can still be tested at the aggregate level. Of the 956 subjects in the data, 625 point estimates indicate that RDU "wins" over the most complete machine learning model. A binomial test rejects the null hypothesis that win rates are at least as high for the most complete machine learning model than for RDU at the 1% level.