ON THE COMPARATIVE PERFORMANCE OF MACHINE LEARNING AND ECONOMIC MODELS FOR RISKY DECISION-MAKING

MATEUS HIRO NAGATA

APRIL 30, 2025

Abstract. What constitutes the pinnacle of decision under risk models? This paper addresses this question by comparing a range of models, from popular economic models to black-box machine learning algorithms, as well as hybrid approaches in predicting the choice of certainty equivalents of risky prospects. The findings demonstrate that there is a relevant gain in descriptive prowess in using machine learning techniques. However, this indicates heterogeneity in the population rather than inadequacy of the economic models.

1 Introduction

Economic models capture key regularities of human decision-making, while also being easy to interpret. Can they be improved upon, at the expense of interpretability? To answer this question, I analyze the prediction power of different models of choices under risk, ranging from established economic models to black-box algorithms, and compare their performance on forecasting certainty equivalents of a binary risky prospect. From that analysis, I find out that there is indeed a little gain in using machine learning algorithms, but at the expense of risking overfitting and interpretability.

My primary contribution is technical. By leveraging the universally approximating capability of neural networks (Hornik et al., 1989), I derive optimal functional forms for both probability weighting function and utility function. However, this non-parametric freedom does not translate into greater predictive power. Adding to that, the found utility form is strikingly similar to a linear utility with a loss aversion parameter of $\lambda = 2$, which is often seen in the literature (Barberis, 2013).

Given the flexibility of neural networks, the simplicity of the resulting probability weighting and utility functions is unexpected. Both functions are continuous, piecewise linear, with two or three breakpoints. This suggests that, for the data considered, simple models adequately captures risk attitudes.

Additionally, I propose a new method for estimating the utility in this context. Exploring free-form utility functions is technically complicated and computationally expensive, particularly because calculating certainty equivalents requires both the utility function and its inverse. My proposed method efficiently and effectively estimate both at the same time by introducing in the loss function a term that accounts for the performance of the inverse function.

Finally, incorporating individual-level identification significantly improves model performance,

indicating that heterogeneity in the population's risk attitudes can be better captured by focusing on individuals or subgroups. Importantly, these subgroups are unlikely to be identifiable based on observable characteristics (Vieider et al., 2019), but are reflected in their choice behavior (Bruhin et al., 2010).

The focus on predicting decisions under risk is certainly not new (Plonsky et al., 2017; Peterson et al., 2021). Most notably, the Choice Prediction Competition that started in 2010, highlights the pursuit of primarily predictive models (Erev et al., 2010). This portion of the literature has focused on generating random "representative samples" of risky prospects to investigate if models are focusing too much on anomalies (such as the Allais' paradox, Ellsberg's paradox, reflection effect) (Erev et al., 2017) and finding models that forecast decision among those random lotteries.

It is worth mentioning also the work of Ellis et al. (2023) that explores the predictivity of machine learning and economic models in the setup in which agents' problem is to choose Arrow securities to hedge against well-defined risk.

Despite sharing the goal of predicting choices under risk, my approach differs. In machine learning terms, I focus on the regression problem of predicting the numeric value of the certainty equivalent for a risky prospect, similar to (Fudenberg & Puri, 2021; Peysakhovich & Naecker, 2017). What sets my work apart is that I "open up" the neural network black box to extract interpretable insights. This is akin to Peysakhovich & Naecker (2017), where they "rediscover" expected utility with nonlinear probability weighting.

Shifting the goal from description to explicit prediction is useful (Yarkoni & Westfall, 2017). It serves as an indication for external validity, favors simpler models, reduces the risk of p-hacking and serves as a good upper bound for assessing whether there is still explainable variance in the model.

In section 2, I describe the dataset and the metrics used in this study. In section 3, I define the models employed. Section 4 presents the results, and section 5 concludes the paper.

2 Setup, Data and Metrics

The experimental dataset was collected by Vieider et al. (2015) and comprises 28 risky binary prospects $(x, p; y)$ along with their correspondent certainty equivalents ce for 2939 individuals, all of whom are students. A risky binary prospect, also referred to a lottery, is a scenario in which with probability p, the outcome is x, and with probability $1-p$, the outcome is y. x is always defined to be the most extreme value. The reference point is framed to be 0 and gains and losses are relative to this point. For a positive prospect, $x > y \geq 0$; and for a negative prospect, $x < y \leq 0$; and for a mixed prospect $x > 0 > y$. The choice is incentived, so, at the end, the individual received a randomly assigned question and final prospect is determined according to its choice.

The 28 prospects are categorized into 14 positive prospects, 13 negative prospects, and 1 mixed prospect, as shown in Tables 1, 2, and 3, respectively. Let $\mathcal{D} = ((x_i, p_i; y_i), ce_i)_i$ represent the dataset, containing all 28 prospects for each of the 2939 subjects.

For the positive and negative prospects, the certainty equivalents are elicited via a choice list

method proposed by Holt & Laury (2002). For each risky lottery $(x, p; y)$, the subject compares the lottery to a series of evenly spaced sure amounts, differing by increments of €0.5, within the range of x to y. The certainty equivalent is defined as the midpoint between the amounts at which the subject switches preference from the lottery to the sure amount. The probability p takes values $p = \frac{i}{8}, i = 1, \ldots, 7$. In practice, this means that the estimated probability weighting function may not provide information for probabilities outside these values. This limitation is especially relevant when discussing certainty and possibility effects (Tversky & Kahneman, 1992).

For the mixed prospect, which is critical for observing loss aversion, the certainty equivalent is fixed at 0. In this case, the subject selects a negative amount y_{28} such that a sure amount of 0 is as desirable as the prospect $(20, \frac{1}{2}; y_{28})$.

For the mixed prospect, which is essential to identify loss-aversion, the certainty equivalent is fixed at 0 and the subject chooses the negative amount y_{28} such that a sure amount of 0 is as desirable as the prospect $(20, \frac{1}{2}, y_{28})$.

The data is highly heterogeneous in terms of nationality[1] represented (30 countries chosen taking into account cultural variety according to Hofstede (1980) cultural attitude scales), study major, family composition, height, and other characteristics that can partially explain risk attitude (Dohmen et al., 2011; Vieider et al., 2019, 2015; Bruhin et al., 2010). Those characteristics *are not* contemplated in this study. However, it does imply that there is substantive heterogeneity that cannot possibly be captured by any model that that considers only the binary prospect's information and that any result obtained here reflects an average over a diverse population.

2.1 Metrics

A model is a function $f : \mathcal{X} \to \mathcal{Y}$ that takes as input $\mathbf{x}$, which includes the binary prospect $(x, p; y)$ and may also include the identification *id*. The output is a prediction of the certainty equivalent $ce \in \mathcal{Y}$. Let P represent the joint distribution of X and Y. The quality of a function's forecasting power is assessed by the mean squared error (MSE)

$$\mathbb{E}_P[\ell(f(\mathbf{x}), ce)] = \mathbb{E}_P[(f(\mathbf{x}) - ce)^2].$$

Now, let us focus on the model selection problem. An ideal model should be interpretable, exhibit descriptive/predictive power, and maintain simplicity. Even though it is not possible to derive meaningful insights from evaluating the weights and biases of neural networks, if embedded in an economic model, it is possible to estimate graphically a probability weighting function and an utility function from it. Therefore, it can be as interpretable as usual economic models.

The latter two properties—predictive power and simplicity—can be measured by completeness and restrictiveness, respectively. In line with the bias-variance tradeoff, a good model must have enough complexity to capture any meaningful structure in the data (thereby reducing bias) while avoiding excessive complexity, which would lead to overfitting and sensitivity to noise (increasing variance). Both properties are only meaningful when compared to other models, so they are defined

[1]The prices are adjusted to their equivalent in purchasing power parity equivalents in the local currency.

Table 1: Positive Prospects

Prospect	Equivalent
$(5, \frac{1}{2}; 0)$	ce_1
$(10, \frac{1}{2}; 0)$	ce_2
$(20, \frac{1}{2}; 0)$	ce_3
$(30, \frac{1}{2}; 0)$	ce_4
$(30, \frac{1}{2}; 10)$	ce_5
$(30, \frac{1}{2}; 20)$	ce_6
$(20, \frac{1}{8}; 0)$	ce_7
$(20, \frac{1}{8}, 5)$	ce_8
$(20, \frac{2}{8}, 0)$	ce_9
$(20, \frac{3}{8}, 0)$	ce_{10}
$(20, \frac{5}{8}, 0)$	ce_{11}
$(20, \frac{6}{8}, 0)$	ce_{12}
$(20, \frac{7}{8}, 0)$	ce_{13}
$(20, \frac{7}{8}, 5)$	ce_{14}

Table 2: Negative Prospects

Prospect	Equivalent
$(-5, \frac{1}{2}; 0)$	ce_{15}
$(-10, \frac{1}{2}; 0)$	ce_{16}
$(-20, \frac{1}{2}; 0)$	ce_{17}
$(-20, \frac{1}{2}; -5)$	ce_{18}
$(-20, \frac{1}{2}; -10)$	ce_{19}
$(-20, \frac{1}{8}; 0)$	ce_{20}
$(-20, \frac{1}{8}; -5)$	ce_{21}
$(-20, \frac{2}{8}; 0)$	ce_{22}
$(-20, \frac{3}{8}; 0)$	ce_{23}
$(-20, \frac{5}{8}; 0)$	ce_{24}
$(-20, \frac{6}{8}; 0)$	ce_{25}
$(-20, \frac{7}{8}; 0)$	ce_{26}
$(-20, \frac{7}{8}; -5)$	ce_{27}

Table 3: Mixed Prospect

Prospect	Equivalent
$(20, \frac{1}{2}; y_{28})$	0

relative to the optimal model and the worst model. Let us define the best mapping $f^* : \mathcal{X} \to \mathcal{Y}$ as the one that satisfies

$$f^*(\mathbf{x}) \in \underset{ce' \in \mathcal{Y}}{\operatorname{argmin}} \mathbb{E}_P[\ell(ce', ce)|\mathbf{x}].$$

The mean squared error of this optimal function $\mathcal{E}_P(f^*) = \mathbf{E}_P[\ell(f^*(\mathbf{x}), ce)]$ is referred to as the irreducible error.

Completeness (Fudenberg & Liang, 2020; Fudenberg et al., 2022, 2023) is a characteristic of a parametric family of functions $\mathcal{F}_\Theta$, let us first define the best function in the family as

$$f^*_\Theta(x) \in \underset{f \in \mathcal{F}_\Theta}{\operatorname{argmin}} \mathbb{E}_P[\ell(f(\mathbf{x}), ce)],$$

with the corresponding mean squared error

$$\mathcal{E}_P(f^*_\Theta) = \mathbb{E}_P[\ell(f^*_\Theta(\mathbf{x}), ce)].$$

Completeness is then defined as the relative advantage of the best function in the parametric family of models $\mathcal{F}_\Theta$ over a naive benchmark f_{base} divided by the relative advantage of the best model in the whole function space

$$\frac{\mathcal{E}_P(f_{base}) - \mathcal{E}_P(f^*_\Theta)}{\mathcal{E}_P(f_{base}) - \mathcal{E}_P(f^*)}.$$

If the family of functions have the same predictive power as the baseline, completeness equals 0. If the family of functions achieves the irreducible error, completeness equals 1. In this study, the baseline is defined as the expected-value prediction of each prospect. In practice, I use the best model as a proxy of the ideal function. Note that this definition depends on the input, so the completeness for the class of functions that take $\mathbf{x} = (x, p; y)$ as input differs from that for $\mathbf{x} = (x, p; y; id)$.

Now, to define restrictiveness, we focus on the set of eligible rules $\mathcal{F}$, which consists of all functions that satisfy conditions that should appear in any relevant dataset. In this context, we limit our attention to functions that satisfy first-order stochastic dominance (FOSD) and monotonicity. Let $\lambda_\mathcal{F}$ be the uniform distribution on $\mathcal{F}$.

Restrictiveness (Fudenberg et al., 2023) is defined as the ratio of the average distance between the parametric model and a randomly generated function from $\mathcal{F}$ to the average distance between this randomly generated function and a naive baseline:

$$r(\mathcal{F}_\Theta, \mathcal{F}) = \frac{\mathbb{E}_{\lambda_\mathcal{F}}[d(\mathcal{F}_\Theta, f)]}{\mathbb{E}_{\lambda_\mathcal{F}}[d(f_{base}, f)]}.$$

The idea is to assess how well a model can fit any synthetic (but uninformative) data within the given context.

The dataset is randomly split into training and testing sets in an 80:20 ratio, a standard approach in the literature to prevent overfitting. This method tests how well the model generalizes to unseen data. Furthermore, the training data is divided for 5-fold cross-validation, and early-stopping methods are employed to avoid overfitting, which is common practice in machine learning.

3 Models

The models used in this study are divided into three groups; 1) economic models, 2) neural network models, and 3) tree-based models. Neural networks are employed as universal approximators, making them closer to economic models than to black-box algorithms.

3.1 Economic Models

Firstly, the simplest of all models is the Expected Value Theory (EVT) model. According to this model, $\hat{ce} = px + (1 - p)y$. Any sensible model should surpass this one in terms of prediction.

The primary economic model employed in this study is the Cumulative Prospect Theory (CPT). I assume here that the utility of a risky prospect $(x, p; y)$ is

$$CPU = \begin{cases} w^+(p)u(x) + (1 - w^+(p))u(y) \text{ if } x, y \geq 0 \\ w^-(p)u(x) + (1 - w^-(p))u(y) \text{ if } x, y \leq 0 \\ w^+(p)u(x) + w^-(1-p)u(y) \text{ if } x > 0, y < 0 \end{cases}$$

Let us simplify notation and define

$$CPU = \pi_x u(x) + \pi_y u(y),$$

where $\pi_x = w^+(p)$ if x is positive and $\pi_x = w^-(p)$ if negative and $\pi_y = (1 - \pi_x)$ if both are the same signal and $\pi_y = w^-(1-p)$ if it is a mixed prospect.

The utility function is defined as:

$$u(x) = \begin{cases} x \text{ if } x \geq 0 \\ -\lambda(-x) \text{ if x} < 0. \end{cases}$$

This utility function highlights loss aversion, "losses loom larger than gains", if the $\lambda > 1$ (typically, it is around $\lambda = 2$), being convex in gains and concave for losses. Risk attitudes vary with stakes - risk seeking for small stakes, risk neutrality for middle stakes and risk aversion for high stakes (Bouchouicha & Vieider, 2017). Given the scope of our experiments, a linear utility for both gains and losses is reasonable. In fact, our results corroborate with this assumption, showing no significant room for improvement.

The model is examined using maximum likelihood estimation (MLE) and neural networks, with more details in the next subsection. For now, let us focus on the MLE estimation, which is the common method in the literature (Bruhin et al., 2010; Vieider et al., 2015). The model assumes that choice is decomposed into a stochastic component ε_i and a deterministic component, the cumulative prospect theory utility CPU_i given a binary prospect $(x_i, p_i; y_i)$. The certainty equivalent is then modeled as $ce_i = CPU_i + \varepsilon_i$, where $\varepsilon_i \sim \mathcal{N}(0, \sigma_i^2)$.

For the probability weighting function, I use Prelec's weighting function (Prelec, 1998):

$$w^+(p) = \exp(-\beta^+(-\ln(p))^{\alpha^+})$$
$$w^-(p) = \exp(-\beta^-(-\ln(p))^{\alpha^-}),$$

where α is the sensitivity parameter and β is the pessimism for gains and optimism for losses parameter. An $\alpha = 1$ returns a $w(p) = p$, the function is totally sensitive to the probability p. As α decreases, the certainty and possibility effects are more pronounced. In the limit, $w(p) = 1/e^{\beta}$. Now, the β determines the point at which the graph crosses the 45 degree line. If $\beta = 1$, it crosses at $\frac{1}{e}$. A higher β draws back this point and a lower β pushes forward.

Let $\theta = (\alpha^+, \beta^+, \alpha^-, \beta^-, \lambda, \sigma)$. Given the assumptions, $CPU - \varepsilon$ should behave according to a normal distribution:

$$\psi(\theta,(x_i,p_i;y_i)) = \phi\left(\frac{CPU_i(\alpha^+,\alpha^-,\beta^+,\beta^-,\lambda,\sigma) - ce_i}{\sigma}\right)$$

Consequently, we estimate the parameter vector θ by maximizing the log-likelihood function:

$$LL(\theta,\gamma) = \sum_{i=1}^{N} log[\psi(\theta,(x_i,p_i;y_i))]$$

3.2 Neural Networks

In addition to the already defined cumulative prospect theory utility (CPU), I also consider a simpler Rank-Dependent Utility (RDU) model:

$$RDU = w(p)u(x) + (1-w(p))u(y)$$

By testing this specification, I can evaluate whether differences in probability weighting function in the gains and losses domain pack reasonable explanatory power or not. Alternatively, I also experiment with the linear utility specification $u_{linear}(x) = x$.

A neural network operates as follows: the input vector $\mathbf{x}$ (k-dimensional) undergoes a linear transformation $\mathbf{W}^1\mathbf{x} + b^1$, where $\mathbf{W}^1$ is a weight matrix of dimension and $\mathbf{b}^1$ is a bias vector of dimension $m_1 \times 1$. This linear combination is then passed through a nonlinear activation function $\sigma(\cdot)$, producing the first hidden layer $\mathbf{h}_1$,

$$\mathbf{h}_1 = \sigma(\mathbf{W}^1\mathbf{x} + \mathbf{b}^1) = \begin{bmatrix} h_{11} \\ h_{12} \\ \vdots \\ h_{1m} \end{bmatrix} = \begin{bmatrix} \sigma(\mathbf{w}_1^1\mathbf{x} + b_1^1) \\ \sigma(\mathbf{w}_2^1\mathbf{x} + b_2^1) \\ \vdots \\ \sigma(\mathbf{w}_m^1\mathbf{x} + b_m^1) \end{bmatrix}.$$

Any further hidden layers $i = 2,\ldots,n$ are defined similarly:

$$\mathbf{h_i} = \sigma(\mathbf{W}^i\mathbf{h}_{i-1} + \mathbf{b}^i)$$

The final hidden layer generates the output

$$z = \sigma(\mathbf{w}^n\mathbf{h}_{n-1} + b^n)$$

A schematic representation is shown in Figure 1, illustrating the most "off the shelf" neural network model. There is no economic insight and the algorithm is free to employ any nonlinear transformation of the input.

There is flexibility in choosing the number of hidden layers n and the amount of nodes $m_1, m_2, \ldots$. A bigger and longer neural network can capture more complex patterns (Hornik et al., 1989; Park

et al., 2020), but is more prone to overfitting. Careful choice of parameters is necessary and exploring is crucial. In this study, the hidden layer nodes are always $m = 10$ and the number of hidden layers is either 1 or 2, chosen because they were the simplest ones that performed well on the data. Heuristically, this setup allows for sufficiently complex to emerge, meaning that the simplicity of the outcomes is a consequence of the data, not necessarily a modelling choice. The function σ chosen is the rectified linear unit (ReLU) activation function $\sigma(x) = \max\{0, x\}$.

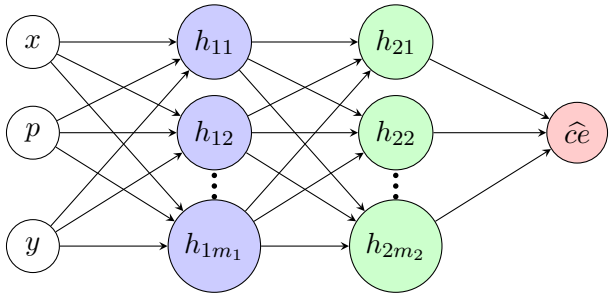

Figure 1: Neural Network with Two Hidden Layers for Certainty Equivalent

One issue that may arise with neural networks is their tendency to converge to a local optimum. However, recent studies (Choromanska et al., 2015) suggest that the difference between a local optimum and a global optimum is minimal in practice.

In this paper, I use neural networks, denoted as $NN(\mathbf{x})$ to approximate different functions. For instance, the probability weighting function $w : [0, 1] \to \mathbb{R}$, the utility function $u : \mathbb{R} \to \mathbb{R}$, or a non-linear certainty equivalent prediction function $f : \mathbb{R}^3 \to \mathbb{R}$ that uses the binary prospect $(x, p; y)$ as the input. Let w_{NN}, u_{NN} and f_{NN} be their estimations using neural networks. In all cases, the primary objective is to predict certainty equivalents, so the performance of the neural network is naturally evaluated by how well it can predict these values.

For example, in the model I refer to as Neural Rank-Dependent Utility (RDU) with Linear Utility, the estimated certainty equivalent is:

$$\widehat{ce} = w_{NN}(p)x + (1 - w_{NN}(p))y$$

The neural network algorithm dynamically adjusts the weights and biases $\mathbf{W}^1, b^1, \mathbf{W}^2, b^2, \ldots$ that minimize the mean squared error

$$\mathbb{E}_P[((w_{NN}(p)x + (1 - w_{NN})(p)y) - ce)^2]$$

In our case, the ADAM algorithm Kingma (2014) is used to find the optimal parameter combination. ADAM has proven to be stable, efficient, and relatively insensitive to hyperparameter choices, making it a reliable choice for this context.

3.3 Types of Neural Networks

I consider neural network models with economic insights as follows:

1. Neural Rank-Dependent Utility with Linear Utility: $\hat{ce} = w_{NN}(p)x + (1 - w_{NN}(p))y$
2. Neural Cumulative Prospect Theory with Linear Utility $\hat{ce} = \pi^x_{NN}x + \pi^y_{NN}y$
3. Neural Rank-Dependent Utility $\hat{ce} = u^{-1}_{NN}[w_{NN}(p)u_{NN}(x) + (1 - w_{NN}(p))u_{NN}(y)\ ]$
4. Neural Cumulative Prospect Theory: $\hat{ce} = u^{-1}_{NN}[\pi^x_{NN}u_{NN}(x) + \pi^y_{NN}u_{NN}(y)]$

I also consider pure neural network models:

5. Neural Network: $\hat{ce} = NN(x, y, p)$
6. Neural Network with Individual Identification: $\hat{ce} = NN(x, y, p, id)$.

A novel contribution of this paper is the development of a new technique for simultaneously estimating both the utility function and its inverse, a problem that has been technically challenging and previously unaddressed in the literature. Past studies have typically defined a parametric form for the utility function to avoid this issue, which leaves open the question of whether these specifications could be improved.

The proposed method consists in defining another neural-network for the inverse function $v_{NN} = u^{-1}_{NN}$. Let me exemplify this framework by dealing with the most complicated model Neural Cumulative Prospect Theory, which is also the best among the neural network models:

$$\widehat{ce} = v[\pi^x u(x) + \pi^y u(y)]$$

Not only we want $\hat{ce}$ to be close to ce, but also to guarantee that the composition of functions $v \circ u$ actually returns the original outcome. So, I introduce a new loss function that needs to be minimized

$$\mathbb{E}_P[(v(u(x)) - x)^2].$$

Finally, the neural network makes a compromise between maximizing the predictive power of the model and ensuring that v_{NN} works as an inverse function. This compromise is captured by the parameter α. Therefore, the weights and biases should be chosen to minimize the following loss

$$\mathbb{E}_P[(v[\pi^x u(x) + \pi^y u(y)] - ce)^2] + \alpha \left[\frac{\mathbb{E}_P[(v(u(x)) - x)^2 + (v(u(y)) - y)^2 + (v(u(ce) - ce)^2)]}{3}\right]$$

Experimentally, the value of $\alpha = 0.2$ was good for attaining these two objectives. $v(u(\cdot))$ is evaluated for x, y and ce because their diverge a lot, but the model should be able to generalize to the whole set.

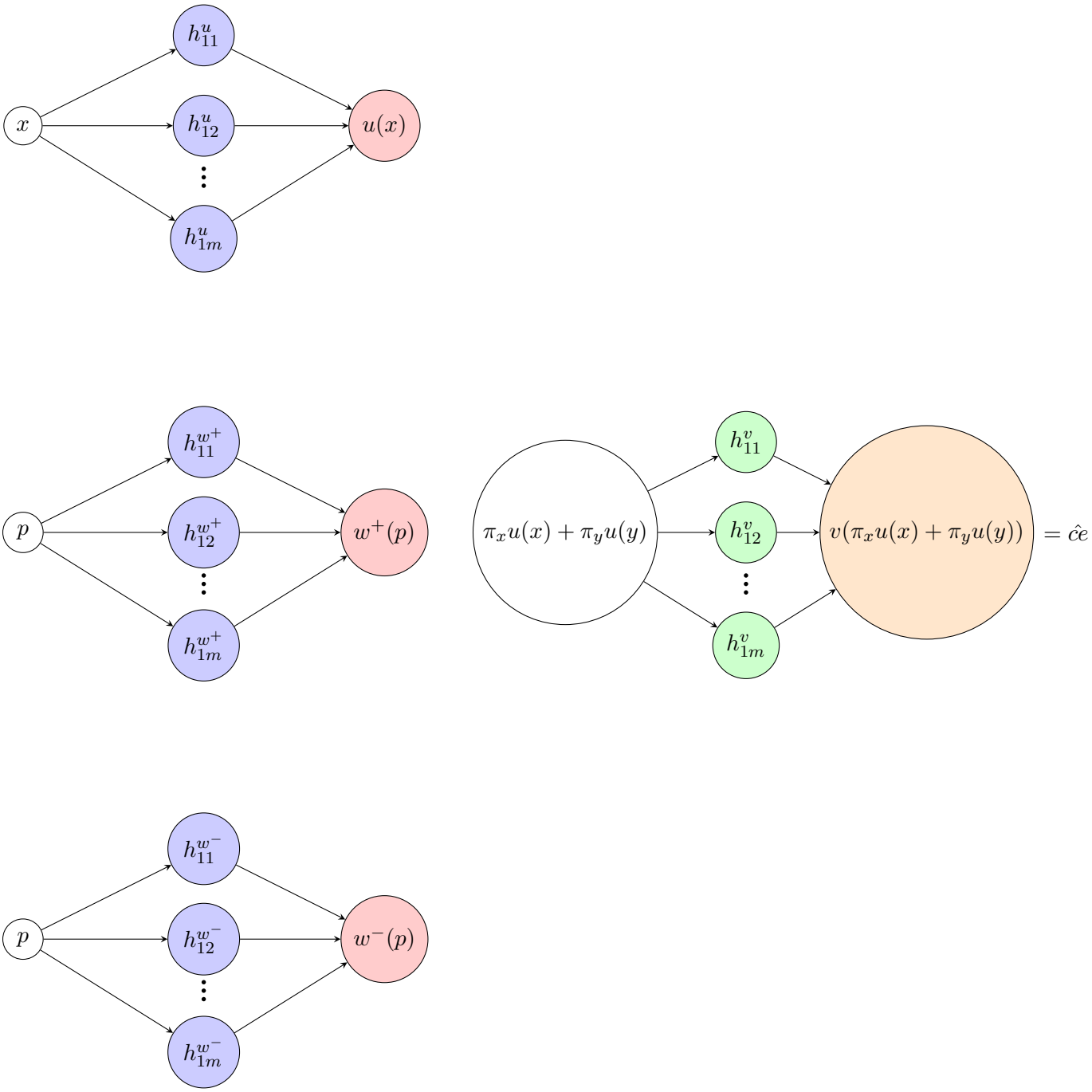

Figure 2: Neural Network Architecture for Cumulative Prospect Theory

3.4 Tree-based Algorithms

The second type of machine learning algorithms used are tree-based algorithms. All of these algorithms follow a similar approach, as outlined in Algorithm 1, where the feature space $\mathcal{X}$ is split according to some optimal criteria and prediction consists on the average value within each split. The employed algorithms are Random Forests, XGBoost and CatBoost and they were used in an "off-the-shelf" manner, without any economic insights. Let the prediction given an input $\mathbf{x}$ be $rf(\mathbf{x}), XGBoost(\mathbf{x}), CatBoost(\mathbf{x})$, respectively. More details on the functioning of each algorithm can be found in the Appendix A.

Algorithm 1 Generic Tree-Based Algorithm Framework

Input: Training data $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$

Parameters: Number of trees T, learning rate η, max depth d, minimum samples per leaf $n_{\min}$, etc.

for $t = 1, \ldots, T$ **do**

a) Compute the gradient $\mathbf{g}_t$ and hessian $\mathbf{h}_t$ (if applicable)

b) Build a tree T_t by recursively splitting the data:

- i. Select the best split point (based on a criterion like Gini, entropy, MSE, or gradient/hessian gain)
- ii. Split the node into child nodes
- iii. Repeat until a stopping criterion (max depth, min samples, etc.) is met

c) Update the predictions $\hat{y}_t$:

$$\hat{y}_t = \hat{y}_{t-1} + \eta T_t(x)$$

end for

Prediction at a new point x**:**

$$\hat{f}(x) = \sum_{t=1}^{T} \eta T_t(x)$$

4 Results

At the risk of repeating myself for the sake of clarity, these are the models I test:

1. EVT: $CE = px + (1-p)y$
2. CPT MLE: $CE = w^{s(x)}_{MLE}(p)u_{MLE}(x) + w^{s(y)}_{MLE}(1-p)u_{MLE}(y)$
3. Neural RDU (Linear): $\hat{ce} = w_{NN}(p)x + (1 - w_{NN}(p))y$
4. Neural CPT (Linear): $\hat{ce} = \pi^{x}_{NN}x + \pi^{y}_{NN}y$
5. Neural RDU: $\hat{ce} = v_{NN}[w_{NN}(p)u_{NN}(x) + (1 - w_{NN}(p))u_{NN}(y)\ ]$

Model	**MSE**	**MAE**	**Completeness**	**Restrictiveness**
EVT	19.85	3.04	0	1
CPT MLE	15.87	2.92	0.78	0.59
Neural RDU (Linear)	16.63	2.97	0.64	0.52
Neural CPT (Linear)	16.37	2.98	0.69	0.46
Neural RDU	16.62	3.03	0.64	0.45
Neural CPT	15.62	2.92	0.84	0.39
$NN(x, p; y)$	15.81	2.92	0.801	0.36
Random Forest	14.81	2.74	1	0.25
XGBoost	14.81	2.74	1	0.25
CatBoost	14.81	2.74	1	0.25

Table 4: Predictive performance of aggregate economic and machine learning models.

6. Neural Cumulative Prospect Theory: $\hat{ce} = v_{NN}[\pi^x_{NN}(p)u_{NN}(x) + \pi^y_{NN}u_{NN}(y)]$
7. NN: $\hat{ce} = NN(x, y, p)$
8. NN with ID: $\hat{ce} = NN(x, y, p, id)$
9. Neural RDU with ID (Linear): $\hat{ce} = NN(w_{NN}(p)x + (1 - w_{NN}(p))y, id)$
10. Random Forest: $\hat{ce} = RF(x, y, p)$
11. XGBoost: $\hat{ce} = XGBoost(x, y, p)$
12. CatBoost: $\hat{ce} = CatBoost(x, y, p)$

The predictions are divided into two frameworks. The first one is the aggregated version, which is shown in Table 4, and the individual version, which is shown in Table 6. In the former case, the comparison does not take into account individual identification, so it treats the whole sample as one unique individual.

In the latter case, individual identification is provided and it is accounted for in the architecture of the neural network. The identification number is transformed by a process of embedding, transforming it into a vector in a high-dimensional space (in our case, 32 dimension). Individuals that have similar behavior are assigned close locations in the space. By adding the identification of variable, we are actually adding 32 variables and not 1. Interpretation becomes much more difficult in this case and it becomes technically more complicated to perform some procedures, which is why I did not estimate a Neural Cumulative Prospect Theory with Individual Identification model.

The estimated CPT MLE model is described by the parameters shown in Table 5, which shows insensitivity and loss-aversion. A comparison of the estimated probability weighting functions between CPT MLE and Neural CPT can be found in the Figure 11.

Parameter	α^+	β^+	α^-	β^-	λ	σ
Estimate	0.60 (0.01)	0.90 (0.02)	0.64 (0.04)	0.94 (0.09)	1.96 (0.02)	0.22 (0.01)

Table 5: Parameter estimates of CPT MLE

Unsurprisingly, cumulative prospect theory (CPT) performs better than expected value theory (EVT). The results from the neural economic models align with existing literature. When comparing Neural RDU with Linear Utility and Neural RDU, we find minimal gains from using a more complex utility function. This suggests that risk attitudes may stem not from utility curvature alone but from the interaction between utility and different weighting functions in the gain and loss domains. This is particularly relevant given that the stakes in this context are not large (Rabin, 2013). The fact that Neural CPT is the most predictive model among the neural economic models also supports this view.

Both CPT MLE and Neural-CPT does show similar performance which indicates that there is not much gain from imposing a much more flexible utility function and weight probability function. Naturally, it entails the fact that the former is more restrictive than the latter. Moreover, arguably, CPT MLE is more interpretable than Neural CPT which the only interpretation we can get is by visual assessment of the derived function approximations. In this sense, CPT MLE should be superior. It also shows that most of the empirical content is already included in the CPT MLE.

Additionally, the $NN(x, p; y)$ model does not perform better. It implies that other nonlinear black-box functions of the binary prospect have no edge, and are actually inferior if the interpretability and flexibility are taken into consideration, as decision models in this context.

Given our choice of prediction, tree-based algorithms scored the best in our analysis. It is common knowledge that in low to middle complexity predictions with medium sized data, this phenomenon occurs frequently (Grinsztajn et al., 2022). Which is a little bit unsatisfying, since there is not much to analyze, since those models are black-boxes and hard to interpret.

At first glance, the results do seem to be a little strange, since their predictive performance is too similar. However, after careful consideration, I conclude that the results are similar due to the nature of the problem, specially its simplicity. Given that there are only 3 features, most of the time, the algorithms partitioned the space in optimal solutions and it turns out that the optimal way is similar for all cases, even if using different methods to do so. This is further evaluated when comparing their performance to the randomly generated functions when studying the restrictiveness.

Despite tree-based algorithms displaying the best performance, it shares few insights on how to improve our models on, given that these models are actually black-boxes and usual Explainable Machine Learning algorithms here are not that much insightful, given the amount of the features.

Now, for Table 6. It is apparent that incorporating individual identification is informative for

Model	MSE	MAE	Completeness	Restrictiveness
$NN(x, p; y, id)$	10.28	2.22	1	0.53
Neural RDU with ID (Linear)	14.93	2.85	0.48	0.47

Table 6: Predictive performance of individual economic and machine learning models.

predictive power especially if the neural network is free-form. Neural RDU with ID has some edge over the other models. Ideally, I would like to estimate a CPT version of this model, but given the inverse utility problem, there are many technical issues to solve to make it possible.

It is worth noting that there should be an issue in the restrictiveness metric for the individual models. Given that there is another variable (id) taken into account, these models should be less restrictive then their aggregate equivalents. Restrictiveness is estimated on how well it can generalize randomly generated data (given some reasonable conditions), so perhaps it shows that the randomly generated id is not much informative.

4.1 Analysis of the Neural Networks

The utility function that comes out from the neural network model is the result of the algorithm to find the continuous piecewise linear function that minimizes the distance between the output and the certainty equivalent. It can be seen at the figures 3, 4. It can be seen that the optimal utility function is monotonic and is very similar to a piecewise linear utility function centered around 0 and $u(x) = x$ if $x \geq 0$, $-\lambda(-x)$ if $x < 0$. Those are not assumptions of the model but the outcomes thereof. Linearity in the gains domain and in the loss domain is also in accordance with the literature with the size of stakes (Rabin, 2013).

Note that estimated utility function behaves bizarrely in the negative vicinity of 0. This reflects the fact that we have a little amount of data in that region, which is composed by the choices of the negative outcomes y_{28} that make the subject indifferent with $0 \sim (20, 0.5, y_{28})$. It also implies the algorithm found optimal to fine-tune the function in this area rather than changing the curvature in other segments of the utility function, further contributing to the optimality idea of a linear utility for gains and losses being the optimal.

It is so far natural to think that continuous finite piecewise linear functions should be "close" to the real utility function, but also it is interesting to see that it can actually represent an underlying preference if (and only if) it satisfies weak order, continuity and weak local bi-independence (Ke & Zhao, 2024). This last axiom is a weakened form of Machina's bi-independence axiom (Machina, 1982).

The composition of the utility and its inverse $v(u(x))$ in figures 5, 6 shows that this novel method is successful in terms of modeling tool and that it can be applied to measure utility functions in different contexts.

Now, for the probability weighting functions, they can be found in figures 7, 8, 9, 10.

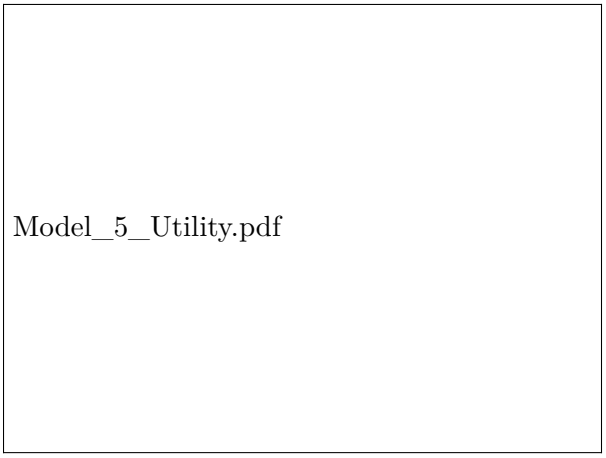

Figure 3: Utility function estimated by the Neural RDU model

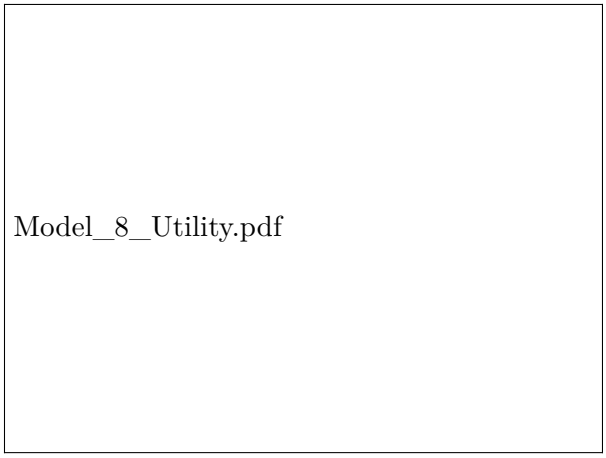

Figure 4: Utility function estimated by the Neural CPT model

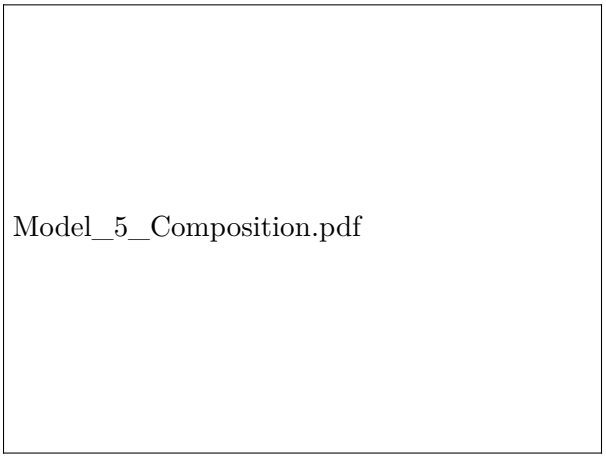

Figure 5: The composition $v(u(x))$ estimated by the Neural RDU model

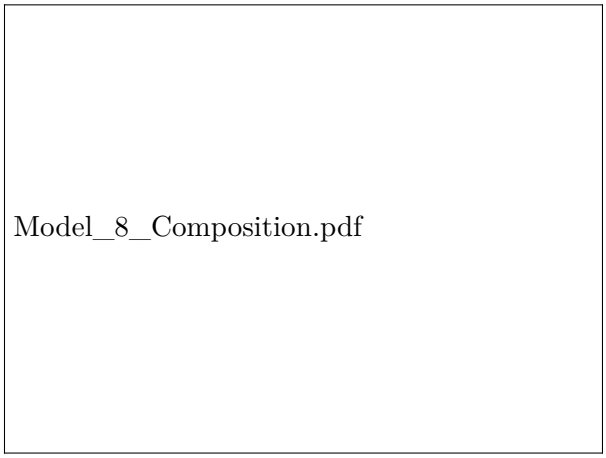

Figure 6: The composition $v(u(x))$ estimated by the Neural CPT model

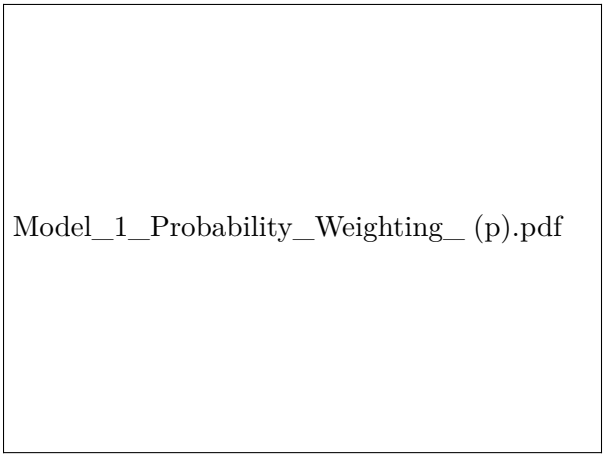

Figure 7: Probability Weighting for Neural RDU with Linear Utility

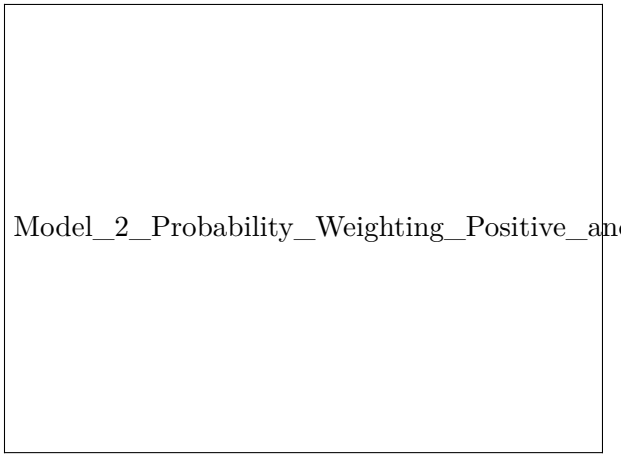

Figure 8: Probability Weighting for Neural CPT with Linear Utility

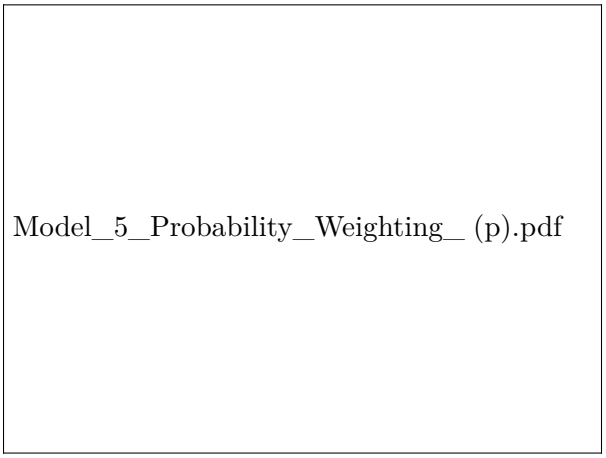

Figure 9: Probability Weighiting for Neural RDU

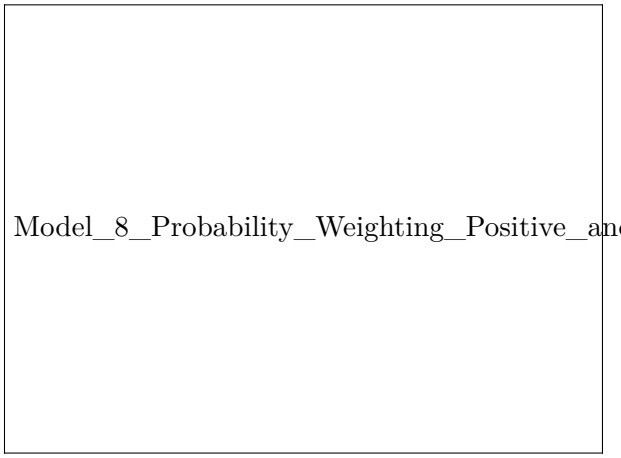

Figure 10: Probability Weighiting for Neural CPT

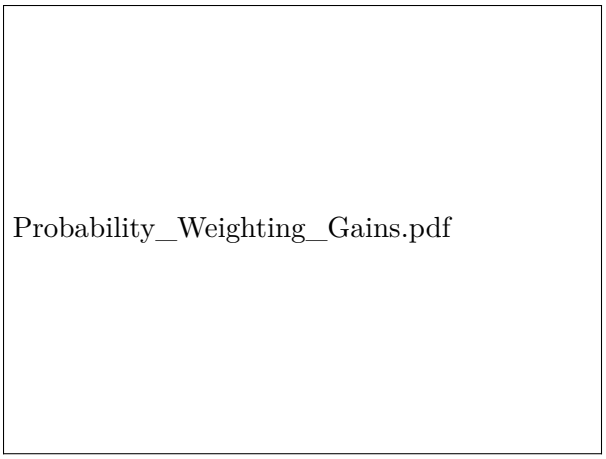

Figure 11: Probability Weighting for Gains of CPT Models

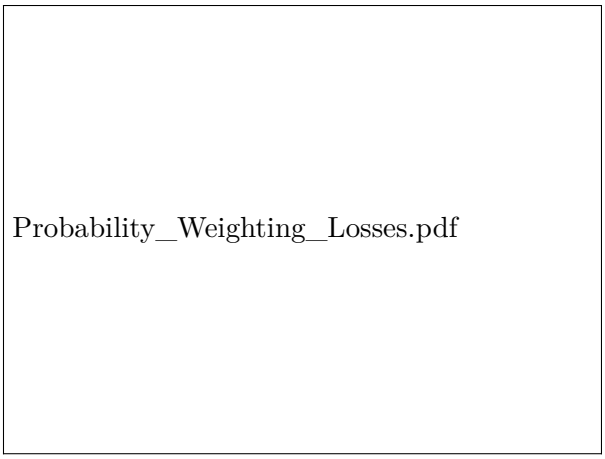

Figure 12: Probability Weighting for Losses of CPT Models

In order to interpret the probability weighting function, it is useful to notice that we only have data for data points in the range $p = \frac{i}{8}, i = 1, \ldots, 8$. The outcome of the function is defined for all points in $p \in [0, 1]$ but it is hard to say any estimation above the $\frac{7}{8} = 0.875$ mark and any mark below $\frac{1}{8} = 0.125$ are reliable. Again, very reasonable outcomes such as the monotonicity and the weight function $w(p)$ be contained in [0,1], are present, which is not hard-wired in the model, but an outcome thereof. It is to say that the algorithm probably worked smoothly. We verify the possibility effect by the portion of the graph that is above the 45 degree line before the $p = 1/8$ mark. The model also harmonize with the certainty effect if we assume $w(1) = 1$, but more data in the extremities is needed to analyze more carefully the curvature in those areas.

Finally, given that the utility for the CPT-MLE model and Neural CPT models and performance are extremely similar, one would expect the learnt probability weighting functions to look very similar. Which is not clear from figure 11, but very similar in case of losses 12. Do note that in practice, both models share the same weights for both $p = \frac{1}{8}$ and $p = \frac{7}{8}$ and there is some, but not much divergence in the middle.

In general, it seems that the simplest models that describe the stylized facts: loss aversion, certainty effect, possibility effect, separate probability weighting functions are the best among the interpretable models. Economics-based Machine Learning algorithms do work, but may not improve upon classical methods. Tree-based algorithms reign in predictive performance, but lack any interpretability.

5 Conclusion

In this study, I compare various machine learning techniques and economic models in terms of their forecasting power. Using data from 2939 individuals, the results indicate that among the explainable models, the integration of machine learning and economic models is not superior to classical models in economics, despite the full flexibility in defining the probability and utility

functions. On the other hand, tree-based models were superior in predictive power, but those raise concern for being too flexible and potentially prone to overfitting. Nevertheless, when considering the "Pareto-frontier" of completeness-restrictiveness, the cumulative prospect theory estimated by maximum likelihood estimation seems to have the edge, with the additional benefit of being interpretable.

Additionally, individual characteristics are informative in pinning down risk attitude, but it is not straightforward integrating it in an economic framework.

The message is still the same as portrayed in the literature, people do have more sensitivity to losses over gains, little probabilities are over-weighted while close to one probabilities are under-weighted and utility is linear when stakes are small.

This work displays a good framework in which black-box algorithms could be integrated in economic analysis, suggesting benchmarks for descriptive models. Decision under risk is a fairly simple and ordinary task cognitively and hence maybe there is not much room for improvement. Perhaps, decision under uncertainty, under different sizes of cognitive load must require more functional flexibility to deal with.

References

Barberis, N. C. (2013). Thirty years of prospect theory in economics: A review and assessment. *Journal of economic perspectives*, 27(1), 173–196.

Bouchouicha, R. & Vieider, F. M. (2017). Accommodating stake effects under prospect theory. *Journal of Risk and Uncertainty*, 55, 1–28.

Bruhin, A., Fehr-Duda, H., & Epper, T. (2010). Risk and rationality: Uncovering heterogeneity in probability distortion. *Econometrica*, 78(4), 1375–1412.

Choromanska, A., Henaff, M., Mathieu, M., Arous, G. B., & LeCun, Y. (2015). The loss surfaces of multilayer networks. In *Artificial intelligence and statistics* (pp. 192–204).: PMLR.

Dohmen, T., Falk, A., Huffman, D., Sunde, U., Schupp, J., & Wagner, G. G. (2011). Individual risk attitudes: Measurement, determinants, and behavioral consequences. *Journal of the european economic association*, 9(3), 522–550.

Ellis, K., Kariv, S., & Ozbay, E. (2023). The predictivity of theories of choice under uncertainty.

Erev, I., Ert, E., Plonsky, O., Cohen, D., & Cohen, O. (2017). From anomalies to forecasts: Toward a descriptive model of decisions under risk, under ambiguity, and from experience. *Psychological review*, 124(4), 369.

Erev, I., Ert, E., Roth, A. E., Haruvy, E., Herzog, S. M., Hau, R., Hertwig, R., Stewart, T., West, R., & Lebiere, C. (2010). A choice prediction competition: Choices from experience and from description. *Journal of Behavioral Decision Making*, 23(1), 15–47.

Fudenberg, D., Gao, W., & Liang, A. (2023). How flexible is that functional form? quantifying the restrictiveness of theories. *Review of Economics and Statistics*, (pp. 1–50).

Fudenberg, D., Kleinberg, J., Liang, A., & Mullainathan, S. (2022). Measuring the completeness of economic models. *Journal of Political Economy*, 130(4), 956–990.

Fudenberg, D. & Liang, A. (2020). Machine learning for evaluating and improving theories. *ACM SIGecom Exchanges*, 18(1), 4–11.

Fudenberg, D. & Puri, I. (2021). *Evaluating and extending theories of choice under risk*. Technical report, Working paper, MIT Economics.

Grinsztajn, L., Oyallon, E., & Varoquaux, G. (2022). Why do tree-based models still outperform deep learning on typical tabular data? *Advances in neural information processing systems*, 35, 507–520.

Hofstede, G. (1980). Culture and organizations. *International studies of management & organization*, 10(4), 15–41.

Holt, C. A. & Laury, S. K. (2002). Risk aversion and incentive effects. *American economic review*, 92(5), 1644–1655.

Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5), 359–366.

Ke, S. & Zhao, C. (2024). From local utility to neural networks. *Journal of Mathematical Economics*, (pp. 103003).

Kingma, D. P. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Machina, M. J. (1982). " expected utility" analysis without the independence axiom. *Econometrica: Journal of the Econometric Society*, (pp. 277–323).

Park, S., Yun, C., Lee, J., & Shin, J. (2020). Minimum width for universal approximation. *arXiv preprint arXiv:2006.08859*.

Peterson, J. C., Bourgin, D. D., Agrawal, M., Reichman, D., & Griffiths, T. L. (2021). Using large-scale experiments and machine learning to discover theories of human decision-making. *Science*, 372(6547), 1209–1214.

Peysakhovich, A. & Naecker, J. (2017). Using methods from machine learning to evaluate behavioral models of choice under risk and ambiguity. *Journal of Economic Behavior & Organization*, 133, 373–384.

Plonsky, O., Erev, I., Hazan, T., & Tennenholtz, M. (2017). Psychological forest: Predicting human behavior. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31.

Prelec, D. (1998). The probability weighting function. *Econometrica*, (pp. 497–527).

Rabin, M. (2013). Risk aversion and expected-utility theory: A calibration theorem. In *Handbook of the fundamentals of financial decision making: Part I* (pp. 241–252). World Scientific.

Tversky, A. & Kahneman, D. (1992). Advances in prospect theory: Cumulative representation of uncertainty. *Journal of Risk and uncertainty*, 5, 297–323.

Vieider, F. M. et al. (2019). All over the map: A worldwide comparison of risk preferences. *Quantitative Economics*, 10(1), 185–215.

Vieider, F. M., Lefebvre, M., Bouchouicha, R., Chmura, T., Hakimov, R., Krawczyk, M., & Martinsson, P. (2015). Common components of risk and uncertainty attitudes across contexts and domains: Evidence from 30 countries. *Journal of the European Economic Association*, 13(3), 421–452.

Yarkoni, T. & Westfall, J. (2017). Choosing prediction over explanation in psychology: Lessons from machine learning. *Perspectives on Psychological Science*, 12(6), 1100–1122.

A Tree-based Algorithms

Algorithm 2 Random Forest Algorithm

1: **for** $b = 1, \ldots, B$ **do**

2: a) Draw a bootstrap sample $\mathbf{Z}^*$ of size N from the training data

3: b) Grow a random-forest tree T_b to the bootstrapped data, by recursively repeating the following steps for each terminal node of the tree, until the minimum node size $n_{\min}$ is reached:

- i. Select m variables at random from the p variables
- ii. Pick the best variable/split-point among the m
- iii. Split the node into two daughter nodes

4: **end for**

5: **Output:** The ensemble of trees $\{T_b\}_{b=1}^{B}$

6: **Prediction at a new point x:**

$$\hat{f}_{\mathrm{rf}}^{B}(x) = \frac{1}{B}\sum_{b=1}^{B} T_b(x)$$

Algorithm 3 XGBoost Algorithm

Input: Training data $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$, regularization parameters λ, γ

Parameters: Number of trees T, learning rate η, max depth d, minimum samples per leaf $n_{\min}$

for $t = 1, \ldots, T$ **do**

a) Compute gradients and hessians for each data point

b) Build a tree T_t using gradient-based splitting:

- i. For each node, choose the best split point to maximize the gain using the gradient and hessian information
- ii. Apply regularization to avoid overfitting (penalizing complex trees)
- iii. Repeat until stopping criteria are met

c) Update the predictions $\hat{y}_t$ using the learning rate η:

$$\hat{y}_t = \hat{y}_{t-1} + \eta T_t(x)$$

end for

Output: The ensemble of trees $\{T_t\}_{t=1}^{T}$

Prediction at a new point x**:**

$$\hat{f}(x) = \sum_{t=1}^{T} \eta T_t(x)$$

Algorithm 4 CatBoost Algorithm

Input: Training data $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$, categorical feature support

for $t = 1, \ldots, T$ **do**

a) Apply ordered boosting to prevent target leakage

b) Handle categorical features using target statistics and a prior

c) Build a tree T_t by recursively selecting the best split point based on loss reduction (gradients and hessians):

- i. Use gradient-based loss and regularization
- ii. Apply symmetry constraints to ensure robust handling of categorical features

d) Update predictions:

$$\hat{y}_t = \hat{y}_{t-1} + \eta T_t(x)$$

end for

Output: The ensemble of trees $\{T_t\}_{t=1}^{T}$

Prediction at a new point x**:**

$$\hat{f}(x) = \sum_{t=1}^{T} \eta T_t(x)$$