

# Assisted Pricing and Foundational Pricing Capabilities: A Scalable, Explainable, and Fair AI-Driven Pricing Architecture

Tanushree Gupta

San Diego, California, USA

Email: [tanushree.851\[at\]gmail.com](mailto:tanushree.851[at]gmail.com)

**Abstract:** *Static price schedules ignore how much service requests vary. The result is estimation error, revenue leakage, and prices that customers cannot interpret. This study tests whether an assisted pricing architecture can improve accuracy without giving up fairness or transparency. The system pairs gradient-boosted trees with a regularized linear model, applies a rule engine for margin, regional, and risk limits, attaches SHAP attributions, and adjusts prices with Q-learning under a fairness-penalized reward. It was run on a simulated dataset of 500,000 transactions from four service industries, repeated over five random seeds. Against a static rate card, the share of estimates within  $\pm 5\%$  of the realized price rose from 11.2% to 34.0%, mean absolute error fell by 67%, and the standard deviation of relative error dropped from 39% to 12%. Simulated price noise caps attainable accuracy at 63.7%, so the system is still well short of a perfect estimator. Simulated CSI rose by 30 points and revenue per session by 35%, but both follow from assumed customer-response models. With an 8% age-based markup injected into the training data, an ungoverned boosted-tree model gave parity ratios of 0.98–1.06. The governed system stayed within 0.99–1.01. A burst of 10,000 concurrent requests on one 4-vCPU node had a median latency of 1.55 s (p95 2.89 s, p99 3.02 s). Reinforcement learning did not reduce six-month pricing drift. The data are simulated, so all results are preliminary.*

**Keywords:** Assisted pricing; Explainable AI; Algorithmic fairness; Reinforcement learning; Dynamic pricing

## 1. Introduction

Pricing shapes value perception, competitiveness, and long-term sustainability. In service sectors such as healthcare, education, legal services, logistics, and consulting, prices are still commonly set from static schedules, heuristic rules, and manual review. These approaches are simple to administer, but they overgeneralize service complexity, leave a large gap between estimated and realized charges, ignore context such as region, urgency, and client history, and give customers no view of how a price was reached.

These weaknesses matter because customers react strongly to perceived unfairness [1] and because algorithmic pricing is under growing regulatory and public scrutiny [2]. Organizations therefore need pricing that is accurate, explainable, and demonstrably fair.

This paper proposes assisted pricing supported by foundational pricing capabilities: data pipelines, feature engineering, a dual-model predictive core, rule-based governance, explainability, and continuous optimization. The goal is not to replace human decision-making but to give operators a price that is accurate, transparent, and auditable, and that scales across industries.

The main contributions are: (i) a modular architecture that wraps a predictive core in rule governance, explanation, and constrained learning; (ii) a fairness-auditing approach that separates parity monitoring from counterfactual justification; (iii) a reinforcement learning formulation whose reward penalizes unfairness and whose actions are bounded by the rule engine; and (iv) a simulation study on 500,000 transactions with an extended protocol for replication. Section 2 surveys related work, Section 3 defines the problem, Section 4 describes the methodology, Section 5

reports results, and Sections 6 and 7 give the conclusion and future scope.

## 2. Literature Survey

Dynamic pricing research studies how a seller can learn demand while setting prices [3]. Recent work uses machine learning for personalized pricing with high-dimensional customer features and heterogeneous elasticity [4], and contextual bandits and reinforcement learning supply general tools for sequential decisions of this kind [5, 6]. This literature concentrates on revenue and regret and gives limited attention to explanation, governance, and fairness.

Algorithmic pricing is also an empirical and regulatory concern. Marketplace studies document how widely automated pricing is used and how it changes market behavior [7]. Models trained on historical data can reproduce or amplify existing disparities without any explicit intent [2], which matters most in healthcare, education, and legal services, where price differences can affect access. Customer reactions to price changes are shaped by perceived fairness [1].

The fairness literature offers several formal criteria, including individual fairness [8], equality of opportunity [9], and counterfactual fairness [10]. These criteria can conflict with one another and with accuracy [11, 12]. Interpretability has been argued to need rigorous evaluation rather than assertion [13], and SHAP is a widely used attribution method [14]. This study combines these strands in one pricing architecture and evaluates it on both accuracy and fairness.

### 3. Problem Definition

#### 3.1 Static pricing as a low-resolution estimator

Typical static structures are tiered packages, SKU-based units, flat consultation rates, and volume-based multipliers. Let  $y$  be the realized charge for a service request with full feature vector  $X$ . Static pricing predicts

$$\hat{y} = f(X_{\text{static}})$$

where  $X_{\text{static}}$  is a small set of tier, unit, or volume features. Because  $X_{\text{static}}$  omits complexity, urgency, region, and client history, the estimator has high bias and large residual variance, and it cannot explain its output. Four structural limitations follow:

- **Overgeneralization of complexity.** Requests in the same category can differ greatly in effort, resources, or risk, and static tiers cannot capture this.
- **High residual variance.** The gap between estimated and actual charges causes underpricing or overpricing and hurts revenue accuracy.
- **Contextual inconsistency.** Regional cost, urgency, client history, and complexity are often ignored.
- **Opaque price logic.** Customers cannot see how a price was set, which reduces trust and can deter engagement.

#### 3.2 Assisted pricing and foundational capabilities

Assisted pricing is a semi-automated approach that combines predictive modeling with rule-based governance to produce transparent, context-aware, and statistically validated price estimates in real time. Its attributes are predictive intelligence (machine learning over service features, client history, and external factors), rule-based governance (compliance constraints, minimum margins, and risk buffers), transparency (explainable outputs for operators and customers), and continuous feedback (reinforcement learning and operational feedback). It rests on six foundational capabilities:

- **Data pipelines** for ingestion, validation, and normalization of transactional data, client history, and contextual metadata.
- **Feature engineering** that turns raw inputs into complexity scores, behavioral features, and categorical encodings.
- **Predictive model infrastructure** pairing gradient-boosted trees [15] for accuracy with regularized linear regression for interpretability.
- **Governance layers** that enforce pricing guidelines, fairness mandates, and margin constraints through a rule engine.
- **Transparency mechanisms** such as SHAP attribution [14] and price composition views.
- **Continuous optimization loops** that recalibrate predictions and reduce long-term drift.

#### 3.3 Objectives and hypotheses

The study has six objectives:

- Improve pricing accuracy to within  $\pm 5\%$  of realized charges, reducing revenue leakage and disputes.
- Reduce estimation variance across heterogeneous requests, supporting predictable revenue.

- Increase the customer satisfaction index (CSI) through clear, explainable pricing.
- Increase revenue per session without compromising fairness or trust.
- Keep prices within parity limits across client segments.
- Maintain a median response time of about 2 s under scaled load.

Four hypotheses follow. **H1:** assisted pricing improves accuracy over static pricing ( $p < 0.001$ ). **H2:** it reduces residual variance. **H3:** CSI rises with transparency and explainability. **H4:** revenue per session rises without amplifying price disparities across the evaluated segments (region and service type).

#### 3.4 Notation

Table 1 lists the symbols used in this paper.

**Table 1:** Notation

| Symbol                 | Meaning                                                    |
|------------------------|------------------------------------------------------------|
| $\hat{y}$              | Predicted price                                            |
| $X$                    | Feature vector of a service request                        |
| $X_{\text{static}}$    | Limited feature set used by static pricing                 |
| $s, a, r$              | State, action, and reward in the RL formulation            |
| $\alpha, \gamma$       | Learning rate and discount factor                          |
| $\Delta R, D_f$        | Revenue gain and fairness deviation                        |
| $\lambda_f, \lambda_s$ | Reward weights on fairness deviation and dissatisfaction   |
| CSI                    | Customer Satisfaction Index (0–100)                        |
| MAE, RMSE              | Mean absolute error, root mean square error                |
| Parity ratio           | Demographic price parity ratio; acceptable range 0.95–1.05 |

## 4. Methodology

#### 4.1 Architecture

The system is a modular pipeline (Fig. 1). A user input layer accepts forms, documents, account history, and geographic indicators. Ingestion validates the schema, imputes missing values, and flags outliers. Feature engineering builds model-ready variables. The predictive engine estimates a base price, the rule engine adjusts it, and the explainability layer attaches SHAP attributions and a confidence interval. Outcomes, operator overrides, and satisfaction signals are logged and drive a reinforcement learning loop that updates pricing adjustments.

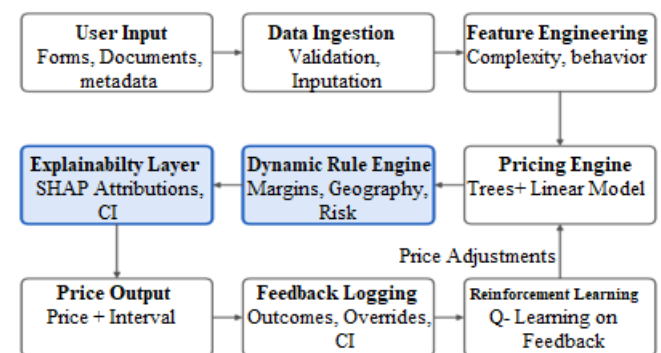


**Figure 1:** Assisted pricing architecture. Shaded blocks are the governance and explanation stages

## 4.2 Data and features

Inputs are structured request forms (service requests, durations, pricing history), uploaded contracts or invoices, historical account metadata, and geographic indicators such as regional regulations and labor costs. Validation includes schema enforcement for completeness, missing-field imputation to prevent model failure, and outlier detection to guard against anomalous or fraudulent entries. Features fall into four groups (Table 2). Normalization, categorical encoding, and scaling make them model-ready.

**Table 2:** Feature categories

| Type         | Examples                               | Purpose                               |
|--------------|----------------------------------------|---------------------------------------|
| Quantitative | Document length, transaction frequency | Service scale and workload            |
| Categorical  | Service type, region                   | Service class and geography           |
| Derived      | Complexity index, urgency score        | Combine several factors in one metric |
| Behavioral   | Past acceptance rate                   | Client engagement and expectations    |

## 4.3 Predictive engine

A dual-model design is used. Gradient-boosted regression trees [15] are the primary model for accuracy, and a regularized linear regression is the secondary model for interpretability and benchmark comparison. Both minimize squared error, with L2 regularization on leaf weights for the trees and on coefficients for the linear model. Models are validated with 10-fold cross-validation stratified by quantile bins of the continuous price target, and hyperparameters are tuned by Bayesian optimization [16]. In the simulation the models predict log price; tuning used a Tree-structured Parzen Estimator (a Bayesian optimizer) with 12 trials on a 40,000-transaction subsample, and the final model was refit on the full training split with 1.5 times the mean early-stopped number of rounds.

## 4.4 Rule engine and explainability

The rule engine enforces policy-driven constraints on model outputs, keeping them inside pricing policy, margin floors, and regional regulation:

- Geographic compliance rules
- Minimum margin thresholds
- Volume discounts
- Risk buffers based on uncertainty or client profile

The explainability layer provides SHAP feature attributions [14] showing each input's contribution to price, price composition visualizations for operators and clients, and confidence interval displays that quantify uncertainty. Together these improve trust and give operators an audit trail.

## 4.5 Reinforcement learning

To adapt over time, the system applies Q-learning [17]. The state  $s$  is the transaction context (service type, client behavior, and environmental factors), the action  $a$  is a bounded pricing adjustment applied to the base prediction, and the reward  $r$  balances revenue against fairness and satisfaction:

$$r = \Delta R - \lambda_f D_f - \lambda_s (1 - \text{CSI}/100) \quad (1)$$

where  $\Delta R$  is the revenue gain,  $D_f$  the fairness deviation, and  $\lambda_f$  and  $\lambda_s$  are non-negative weights set by governance policy. The value update is

$$Q(s, a) \leftarrow Q(s, a) + \alpha [r + \gamma \max_{a'} Q(s', a') - Q(s, a)] \quad (2)$$

with learning rate  $\alpha$  and discount factor  $\gamma$ . Tabular Q-learning needs discrete states and actions, so context features are binned and adjustments use fixed increments; for continuous contexts, deep Q-networks [18] are a natural extension. Every adjusted price passes through the rule engine, so no action can breach a margin floor, regional rule, or parity threshold, and exploration stays inside this feasible set (Fig. 2). In the simulation, the reward in Eq. (1) is measured relative to the unadjusted base quote, so that  $\Delta R$  and the change in CSI are gains over making no adjustment, and  $D_f$  is the absolute difference between the chosen adjustment and the mean adjustment in the same mini-batch. The contextual bandit used for comparison is the same learner with  $\gamma = 0$ .

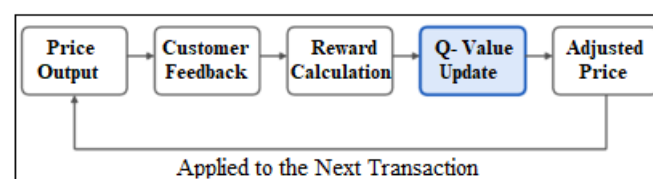


**Figure 2:** Reinforcement learning feedback loop.

## 4.6 Governance and fairness monitoring

Fairness is monitored in two complementary ways (Fig. 3):

- **Demographic parity checks** regularly measure price ratios across age, region, and client type, with an acceptable range of 0.95–1.05, to detect systemic over- or undercharging.
- **Counterfactual simulations** test, for each transaction, whether changing a sensitive attribute would change the price [10].
- **Bias drift alerts** notify operators automatically when fairness metrics exceed predefined thresholds, enabling timely intervention.

The two criteria can conflict [11, 12]: parity expects similar outcomes even when legitimate cost drivers differ, whereas counterfactual fairness permits differences that arise from legitimate features. The framework therefore treats parity ratios as a monitoring signal and uses counterfactual tests to judge whether a difference is justified.

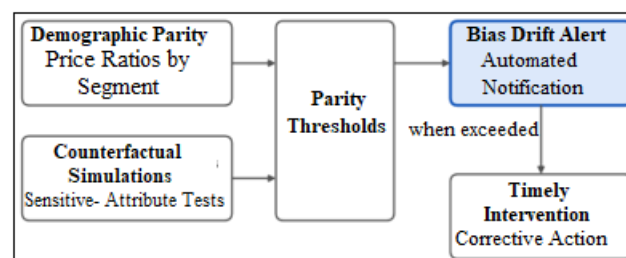


**Figure 3:** Fairness monitoring workflow

Privacy protections include differential privacy noise injection to limit reverse engineering of individual transactions [19], anonymized feature hashing that encodes client identifiers while preserving modeling utility, and secure retention policies that limit exposure of sensitive

features while keeping traceability for audits. The effect of these controls on accuracy was not separately evaluated.

#### 4.7 Ethical and societal impact

Automated pricing in healthcare, education, and legal services can affect access to essential services, so the framework treats the following as design requirements:

- **Price discrimination:** prices may differ only for legitimate cost drivers such as complexity, urgency, and regional cost, and counterfactual tests and parity monitoring check that sensitive attributes do not drive differences.
- **Human oversight:** operators can review, override, and explain any price, and overrides are logged as feedback.
- **Learning on customers:** reinforcement learning explores prices, which can expose customers to worse outcomes. Actions are bounded, subject to the rule engine, and tested in simulation and shadow mode before any live use.
- **Contestability:** customers receive the price composition and confidence interval and have a route to dispute a price.
- **Regulation and privacy:** rule sets must be adapted to local pricing, consumer protection, and data protection law, and privacy controls apply to all logged data.
- **Misuse:** the same machinery could be used to extract maximum willingness to pay. Parity thresholds and price caps in the rule engine can limit this, but they are policy choices that the deploying organization must set and publish.

#### 4.8 Experimental Setup

A simulated multi-domain dataset was constructed: 500,000 transactions from healthcare, logistics, education, and consulting, with 42 structured and 7 derived complexity features. Simulation allows control over variance, complexity, and fairness testing without proprietary data. Details needed for replication are listed in Table 3. Every experiment was run over five random seeds (0–4), each of which regenerates the data, retrains every model, and reruns

reinforcement learning; tables report means over seeds. The evaluation metrics are:

- **Accuracy:** share of estimates within  $\pm 5\%$  of the actual price.
- **MAE and RMSE:** mean absolute error and root mean square error of predicted vs. actual prices.
- **CSI:** Customer Satisfaction Index (0–100), produced by a simulated customer-response model and therefore reflecting its assumptions rather than surveyed opinion.
- **Parity ratio:** demographic price parity ratio, acceptable range 0.95–1.05.
- **Response time:** median API latency.

Assisted and static pricing are compared with paired t-tests, bootstrap confidence intervals [20], Levene's test for variance [21], and a fairness disparity index. With 500,000 transactions even small differences give very small p-values, so effect sizes should accompany them.

An extended protocol, each part repeated over several random seeds, adds:

- **Baselines:** linear-only, tree-only, and rules-only models.
- **Ablations:** the full system with the rule engine, explainability layer, or reinforcement learning removed, and with a contextual bandit in place of reinforcement learning.
- **Calibration:** the share of realized prices inside the reported confidence intervals, compared with their nominal level.
- **Sensitivity:** accuracy under noisier inputs, missing features, and distribution shift.
- **Per-industry reporting:** accuracy and error for each industry, with an error analysis of the largest residuals.
- **Fairness:** parity ratios for age and client type, intersections of segments, and a numerical report of the counterfactual tests.
- **Explanation stability:** rank agreement of SHAP attributions across retrains.

**Table 3:** Simulation and training specification. All generator parameters were fixed before the reported runs; the customer-response and reward settings were adjusted in pilot runs (see Section 7)

| Item                         | Specification                                                                                                                                                                                                                                                                                                                                                                                                                          |
|------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Transactions                 | 500,000 per seed                                                                                                                                                                                                                                                                                                                                                                                                                       |
| Industries                   | Healthcare, logistics, education, consulting (mix 30/25/25/20%)                                                                                                                                                                                                                                                                                                                                                                        |
| Features                     | 42 structured (5 categorical, 16 quantitative, 6 behavioral, 15 contextual, of which 6 are pure noise), 7 derived                                                                                                                                                                                                                                                                                                                      |
| Price generation process     | Log price = industry base (120, 60, 200, 300) + service-type premium + $0.12c + 0.025c^2$ (latent complexity $c$ , observed only through noisy proxies) + 0.18 log scope + urgency terms + labor-cost pass-through (0.30–0.55 of log regional index) + industry terms (fuel, distance, insurance, seats) + seasonality, volume discount (3% per tier), client effect, risk premium. The cost-justified price is exp of this sum.       |
| Noise model                  | Realized price = cost-justified price $\times$ Student-t(5) log noise (sd 0.04, 0.05, 0.025, 0.06 by industry, $\times 1.5$ for service type 2) $\times$ a $\pm 20\%$ rework shock on 1.5% of transactions. The cost-book feature has 12% noise and does not drift.                                                                                                                                                                    |
| Segment attribute assignment | Region A–D with industry-specific probabilities; service type 1/2 (P(type 2) 0.30–0.45); age band <35/35–64/65+ (0.30/0.50/0.20); client type individual/organization (P(org) 0.20–0.85). Tenure and transaction frequency are generated from age and client type, so they act as proxies.                                                                                                                                             |
| Injected bias, if any        | +8% markup on realized prices for age 65+, unrelated to cost, in all splits. Parity is measured against the cost-justified price, which is available only in simulation.                                                                                                                                                                                                                                                               |
| CSI response model           | $CSI = 92 - 120 e  - 80 \max(e, 0) + 3.5$ (if an explanation is shown) + $N(0, 6^2)$ , clipped to 0–100, where $e = \text{quote}/\text{realized}$ – 1. Acceptance is logistic (slope 25) in quote/WTP, with $WTP = \text{realized price} \times \exp(N(0.20, 0.08^2))$ ; 50% of an under-quote shortfall is collected, and revenue per session = acceptance $\times$ billed amount. The 3.5-point explanation uplift is an assumption. |
| Drift simulation (6 months)  | Unobserved monthly cost drift of 0.3–1.0% by industry plus 0–1.2% by region (up to 13% by month 6 for consulting in region C), observable fuel index +0.8% per month, and a rising urgency share. 50,000 new transactions per month; all models are frozen except the RL and bandit agents, which keep learning with $\epsilon = 0.05$ .                                                                                               |

|                                                                            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|----------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Train/validation/test split                                                | 300,000 / 100,000 / 100,000 (random). Validation data calibrate confidence intervals and train the RL agent; all reported results use the test split or fresh drift samples.                                                                                                                                                                                                                                                                                                                                  |
| Seeds and number of runs                                                   | 5 seeds (0–4); data, models, and RL are regenerated for each. Hyperparameters were tuned once, on seed 0.                                                                                                                                                                                                                                                                                                                                                                                                     |
| Model hyperparameters                                                      | LightGBM, L2 loss on log price: 24 leaves, learning rate 0.032, 833 trees, min 169 samples per leaf, feature fraction 0.79, bagging 0.79, L2 0.30; chosen by TPE over 12 trials with 10-fold CV stratified by price-quantile bin. Ridge: $\alpha = 0.1$ . Rule engine: margin floor $1.05 \times$ cost-book estimate, band $0.5\text{--}2 \times$ rules-only price, +2% risk buffer, age band and client type masked from model inputs. Intervals: split conformal by industry and service type, nominal 90%. |
| RL parameters ( $\alpha$ , $\gamma$ , $\lambda f$ , $\lambda s$ , actions) | Tabular Q-learning; 288 states (industry $\times$ region $\times$ service type $\times$ complexity tercile $\times$ past-acceptance tercile); 7 actions ( $-6\%$ to $+6\%$ in 2% steps); $\alpha = 0.10$ , $\gamma = 0.90$ , $\lambda f = 1.0$ , $\lambda s = 2.0$ ; $\epsilon$ from 0.30 to 0.05 over 4 epochs of mini-batches of 2,000 on the validation stream.                                                                                                                                            |
| Hardware and software                                                      | 4 vCPU aarch64 Linux sandbox, 3 GB RAM; Python 3.10.12, LightGBM 4.7.0, scikit-learn 1.7.2, SciPy 1.15.3, NumPy 2.2.6. SHAP values come from LightGBM's built-in TreeSHAP.                                                                                                                                                                                                                                                                                                                                    |
| Code and data availability                                                 | Simulation script and per-seed outputs accompany this submission; data are regenerated from the seeds.                                                                                                                                                                                                                                                                                                                                                                                                        |

## 5. Results and Discussion

### 5.1 Accuracy and variance

Table 4 and Fig. 4 compare the two systems on the held-out test split, averaged over five seeds. Accuracy within  $\pm 5\%$  rose from 11.2% to 34.0%, a gain of 22.8 percentage points (seed SD 0.2 pp; bootstrap 95% CI on seed 0, 22.5 to 23.2 pp). MAE fell from 54.4 to 17.8 ( $-67\%$ ). A paired t-test on absolute errors gives  $p < 0.001$  and Cohen's  $d$  of 0.65. RMSE dropped 66%. The standard deviation of relative error went from 39.1% to 11.8%, about a 91% cut in variance (Levene's test  $p < 0.001$ ). Mean CSI rose from 50.9 to 81.2, and revenue per session by 35.1%. H1 and H2 hold in simulation. H3 holds only by construction, because CSI here is a function of quote error plus an assumed explanation uplift. The revenue gain has the same weakness, since it comes from the assumed acceptance and under-quote leakage model. Accuracy is also capped. An estimator that knew the cost-justified price exactly would reach only 63.7% (Table 8), so the assisted system captures about 53% of what is attainable. At 500,000 transactions  $p$ -values say little, and the effect sizes and intervals matter more.

**Table 4:** Performance comparison, static vs. assisted pricing

| Metric                 | Static | Assisted | Change   | Seed SD |
|------------------------|--------|----------|----------|---------|
| Accuracy ( $\pm 5\%$ ) | 11.20% | 34.00%   | +22.8 pp | 0.2 pp  |
| MAE                    | 54.4   | 17.8     | -67%     | 0.09    |
| RMSE                   | 83.2   | 28.6     | -66%     | 0.26    |
| SD of relative error   | 39.10% | 11.80%   | -70%     | 0.06 pp |
| CSI (assumed model)    | 50.9   | 81.2     | 30.3     | 0.04    |
| Revenue/session        | 0%     | 35.10%   | 35.10%   | 0.3 pp  |

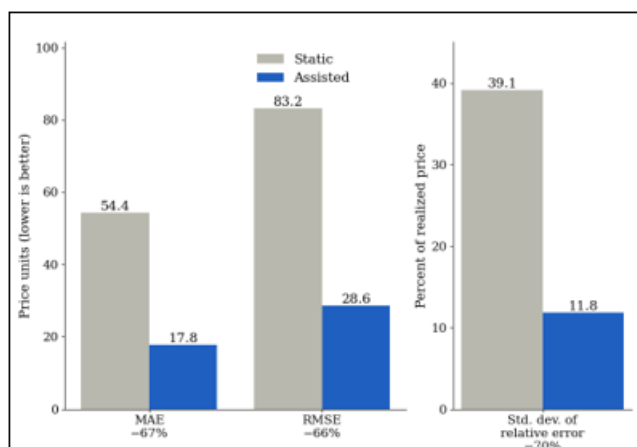


**Figure 4:** Error and variability metrics, static vs. assisted pricing (means over five seeds); percent change is shown

under each metric. MAE and RMSE are in price units; the right panel shows the standard deviation of relative error

Lower variance means steadier revenue and fewer disputes. The rule engine bounds extreme quotes. Even so, the residual analysis in Section 5.5 shows that the largest errors come mostly from rework and dispute events that no input feature predicts.

### 5.2 Fairness

Table 5 gives price parity ratios for the eleven single segments. A segment's ratio is its mean quote-to-cost-justified-price ratio divided by the overall mean. The static rate card ignores regional cost and falls outside 0.95–1.05 for two segments, Region C (0.945) and Region D (1.051). The other nine are inside, so the static baseline is only mildly unfair here. The ungoverned boosted-tree model learns the injected 8% markup from historical prices and reproduces it: 1.062 for age 65+. The governed system masks age band and client type and applies the rule engine. Its ratios span 0.993–1.009 across the eleven segments and 0.990–1.014 across the 24 region, age, and client-type intersections. That costs 2.7 pp of accuracy against the ungoverned model (Table 8), because the markup is part of the realized prices used for scoring.

**Table 5:** Price parity ratios (acceptable range 0.95–1.05), relative to the cost-justified price; means over five seeds

| Segment        | Static | Trees only | Full system |
|----------------|--------|------------|-------------|
| Region A       | 1.026  | 1          | 0.999       |
| Region B       | 0.982  | 1          | 1.001       |
| Region C       | 0.945  | 0.999      | 1           |
| Region D       | 1.051  | 1.001      | 1           |
| Service type 1 | 0.999  | 1.001      | 1.001       |
| Service type 2 | 1.001  | 0.999      | 0.999       |
| Age <35        | 1      | 0.984      | 0.993       |
| Age 35–64      | 1      | 0.985      | 1           |
| Age 65+        | 0.999  | 1.062      | 1.009       |
| Individual     | 0.999  | 1          | 1           |
| Organization   | 1.001  | 1          | 1           |

Counterfactual tests flip one sensitive attribute and shift its proxy using the generating equations: tenure for age band, transaction frequency for client type. For the full system, an age flip changes 43% of quotes by more than 1%, with a mean absolute change of 1.0%. A client-type flip changes 0.1% of quotes. The ungoverned tree model changes 60% of quotes on an age flip (mean 4.6%). The static rate card never changes. Masking therefore reduces the dependence on age but does

not remove it, because tenure still carries age. The earlier claim that remaining differences come only from legitimate features is not supported. H4 holds for the eleven segments tested, with that caveat.

### 5.3 Reinforcement learning

Table 6 summarizes the six-month drift run. Unobserved cost inflation moved the frozen full system's signed pricing error from +1.2% to -5.7% and cut its accuracy from 34.7% to 29.2%. Q-learning did not fix this. It began at -1.5% because of its price cuts and ended at -5.8%, the same place as the system without RL. The bandit ended at -5.8%. Month-6 accuracy was 29.2%, 29.1%, and 29.0% for no RL, bandit, and Q-learning, and the maximum segment parity deviation was 4.2%, 4.2%, and 3.8%. The learned policy chose cuts of 2% to 6% almost everywhere (mean -2.8% at the start). In the assumed response model, a lower quote raises acceptance and CSI by more than it loses in leakage. At month 0 that lifted revenue by 2.6 pp and CSI by 1.1 points over the same system without RL, at a cost of 0.7 pp of accuracy. By month 6 the differences were gone, with revenue at +27.3% over static in all three. The claim that RL cuts drift is not supported. A reward built on revenue and satisfaction does not push quotes toward the realized charge, and actions limited to  $\pm 6\%$  cannot follow cost drift that reached 13% in some segments.

**Table 6:** Effect of reinforcement learning after six months of simulated drift (means over five seeds; RL and bandit rows include exploration).

| Metric (month 6)         | No RL  | Bandit | Q-learning |
|--------------------------|--------|--------|------------|
| Signed pricing error (%) | -5.7   | -5.8   | -5.8       |
| Shift since month 0 (pp) | -6.9   | -4.6   | -4.3       |
| Accuracy ( $\pm 5\%$ )   | 29.20% | 29.10% | 29.00%     |
| Max parity deviation (%) | 4.2    | 4.2    | 3.8        |
| Revenue vs. static       | 27.30% | 27.30% | 27.30%     |
| CSI                      | 81.2   | 81.1   | 81.1       |

### 5.4 Scalability

Load was raised from 100 to 10,000 concurrent requests. Each level arrives as one burst at a single process with four worker threads, which serve requests in micro-batches of up to 32 (Table 7). With explanations, median latency was 39 ms at 100 requests, 0.18 s at 1,000, and 1.55 s at 10,000. That meets the 2 s target. At 10,000 requests p95 was 2.89 s and p99 was 3.02 s. SHAP is the expensive step. One request takes 1.29 ms with attributions and 0.12 ms without, against 3  $\mu$ s for the static rate card. Without explanations the 10,000-request burst has a median of 54 ms, and static pricing has 8 ms. CPU averaged 97% across the four vCPUs during the burst (peak 100%), and process memory peaked at 0.8 GB, including the test data held in memory. Disk and network throughput were not measured because requests are in-process.

**Table 7:** Latency percentiles for a burst of concurrent requests on one 4-vCPU node (four workers, micro-batches of up to 32)

| System, concurrent requests | p50 (s) | p95 (s) | p99 (s) |
|-----------------------------|---------|---------|---------|
| Assisted + SHAP, 100        | 0.039   | 0.04    | 0.04    |
| Assisted + SHAP, 1,000      | 0.176   | 0.33    | 0.33    |
| Assisted + SHAP, 10,000     | 1.55    | 2.891   | 3.019   |
| Assisted, no SHAP, 100      | 0.002   | 0.002   | 0.002   |

|                           |       |       |       |
|---------------------------|-------|-------|-------|
| Assisted, no SHAP, 1,000  | 0.008 | 0.015 | 0.016 |
| Assisted, no SHAP, 10,000 | 0.054 | 0.097 | 0.101 |
| Static, 100               | 0.001 | 0.001 | 0.001 |
| Static, 1,000             | 0.001 | 0.002 | 0.002 |
| Static, 10,000            | 0.008 | 0.013 | 0.014 |

Throughput with explanations was 836 requests/s on one worker, 1,674 on two, and 3,311 on four, so scaling inside one node was close to linear. Scaling across containers, network latency, and a production feature store were not tested. The earlier statement that the system scales near-linearly under container orchestration is therefore not established here. The measured path covers feature engineering, the model, the rule engine, the RL lookup, the interval, and SHAP, on records that are already validated.

### 5.5 Extended Evaluation

Tables 8 to 10 report the extended analyses from Section 4.8, averaged over five seeds. Parity in Table 8 is the minimum and maximum ratio across the eleven single segments. Acc. (M6) is accuracy at month 6 of the drift run. Seed standard deviations were at most 0.2 pp for accuracy and 0.2 for MAE in every row.

**Table 8:** Baselines and ablations

| Configuration                        | Acc.   | MAE   | Parity    | CSI  | Acc. (M6) |
|--------------------------------------|--------|-------|-----------|------|-----------|
| Static pricing                       | 11.20% | 54.39 | 0.94–1.05 | 50.9 | 10.70%    |
| Linear model only                    | 36.80% | 16.21 | 0.98–1.06 | 78   | 29.60%    |
| Boosted trees only                   | 36.70% | 16.3  | 0.98–1.06 | 78   | 29.40%    |
| Rules only                           | 11.20% | 54.04 | 0.99–1.01 | 51.1 | 10.90%    |
| No rule engine                       | 35.60% | 16.8  | 0.98–1.06 | 82.3 | 29.40%    |
| No explainability                    | 34.00% | 17.85 | 0.99–1.01 | 77.8 | 29.00%    |
| No RL                                | 34.70% | 17.46 | 0.99–1.01 | 80.1 | 29.20%    |
| Bandit instead of RL                 | 34.20% | 17.72 | 0.99–1.01 | 81.2 | 29.10%    |
| Full system                          | 34.00% | 17.85 | 0.99–1.01 | 81.2 | 29.00%    |
| Noise ceiling (cost-justified price) | 63.70% | 9.45  | 1.00–1.00 | n/a  | n/a       |

**Table 9:** Results by industry

| Industry   | Static acc. | Assisted acc. | Static MAE | Assisted MAE |
|------------|-------------|---------------|------------|--------------|
| Healthcare | 11.20%      | 34.70%        | 38.6       | 12.3         |
| Logistics  | 11.10%      | 33.10%        | 20.8       | 6.9          |
| Education  | 11.30%      | 36.00%        | 66.3       | 20.3         |
| Consulting | 11.10%      | 31.40%        | 105.3      | 36.9         |

**Table 10:** Calibration and sensitivity of the full system (nominal interval coverage 90%). Models are frozen; the shift scenario changes the industry and region mix, raises complexity and urgency, and adds 4% unobserved cost inflation, so its MAE is not comparable with the baseline

| Test                            | Acc.   | MAE  | Coverage |
|---------------------------------|--------|------|----------|
| Baseline simulation             | 34.00% | 17.8 | 89.90%   |
| Missing features (10% of cells) | 32.50% | 19.6 | 87.30%   |
| Increased input noise (15%)     | 26.10% | 23.1 | 79.00%   |
| Distribution shift              | 21.90% | 38.7 | 81.10%   |

Boosted trees were no more accurate than the regularized linear model (36.7% vs. 36.8%). The generator is close to

additive in the observed features, and its nonlinear complexity term is visible only through noisy proxies, so the dual-model design is not justified by accuracy here. The rules-only baseline did no better than the static rate card (11.2%), because simple regional and urgency rules miss the cost structure. Removing the rule engine raised accuracy against realized prices (35.6% vs. 34.0%) and brought back the age-based parity gap (Table 8). Explainability does not change the quote, so accuracy, MAE, and parity are identical with and without it. Its only simulated effect is the assumed 3.5-point CSI uplift. RL changed accuracy by -0.7 pp, MAE by +0.39, and CSI by +1.1. The bandit matched Q-learning, so the sequential structure added nothing in this setting.

Interval coverage was near nominal at baseline (89.9% against 90%) and fell under perturbation: 87.3% with missing features, 79.0% with noisier inputs, and 81.1% under distribution shift. Accuracy dropped most under noise (26.1%) and shift (21.9%) but stayed above the static rate card, which held near 11% throughout. By industry (Table 9), assisted accuracy ranged from 31.4% to 36.0% and MAE fell 65–69%. Of the largest 1% of relative errors (above 33%), 16% were injected rework or dispute shocks, against a 1.5% base rate. Logistics (30%) and consulting (31%) were over-represented relative to their 25% and 20% shares of traffic, and service type 2 made up 52% of these cases. SHAP attributions were stable across the five retrains, with a mean pairwise Spearman correlation of 0.971 (minimum 0.961) and a mean top-5 overlap of 92%. The top-ranked features were estimated cost, industry, average past price, service type, workload.

### 5.6 Cross-industry applicability

The architecture is designed for cross-industry use (Table 11). Only healthcare, logistics, education, and consulting were evaluated; e-commerce and legal are proposed applications that were not tested. Industry-agnostic foundational capabilities allow rapid adaptation to sector-specific requirements, and the transparency and fairness mechanisms apply broadly.

**Table 11:** Cross-industry use cases.

| Industry   | Application                                                                                                          | Tested |
|------------|----------------------------------------------------------------------------------------------------------------------|--------|
| Healthcare | Real-time cost estimation for patient services, considering complexity, insurance coverage, and regional adjustments | Yes    |
| Logistics  | Route-based adaptive pricing using delivery time, distance, and fuel cost volatility                                 | Yes    |
| Education  | Context-aware tuition modeling by program type, course complexity, and student history                               | Yes    |
| Consulting | Engagement-based estimates reflecting scope, urgency, and complexity                                                 | Yes    |
| E-commerce | Bundled dynamic pricing using product combinations, seasonality, and inventory levels                                | No     |
| Legal      | Document-based consultation estimates with urgency, complexity, and geographic adjustments                           | No     |

### 5.7 Comparison with previous research

Direct numerical comparison with published studies is not possible because their data differ and ours are simulated, so the comparison is qualitative. Dynamic pricing studies [3, 4]

typically emphasize revenue or regret rather than accuracy against realized charges, explanation, or parity, which this study treats as first-class requirements. Fairness studies [8, 9, 10] define criteria but do not themselves prescribe an operational pricing pipeline; here parity monitoring, counterfactual tests, and drift alerts run alongside the predictor. Compared with static schedules, the assisted system is more accurate, less variable, and explainable at the level of individual prices.

Rules bound extreme quotes. Together with masking of sensitive attributes, they removed a parity gap that the ungoverned model inherited from biased history. Reinforcement learning was meant to limit long-term drift and did not (Section 5.3). The SHAP price composition and interval displays are supported here only by stable attributions, not by evidence that customers understand them. Modular pipelines should ease deployment across sectors, but only the four simulated industries were tested.

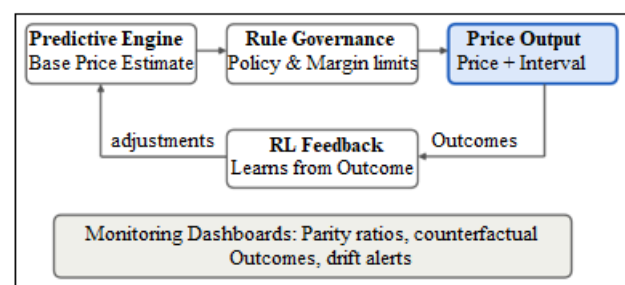


**Figure 5:** Integration of the predictive engine, rule governance, reinforcement learning feedback, and monitoring

## 6. Conclusion

This study built an assisted pricing architecture around a predictive core, with rule governance, explanation, fairness monitoring, and constrained reinforcement learning. It was tested in a seeded simulation with baselines, ablations, calibration, sensitivity, and latency runs. The predictive core with rules was more accurate and less variable than a static rate card: accuracy rose from 11.2% to 34.0% and MAE fell 67%. It reached only about 53% of the 63.7% ceiling set by simulated price noise, and boosted trees gave no gain over a regularized linear model. Masking sensitive attributes and applying rules kept parity ratios within 0.99–1.01, where an ungoverned model reached 1.06, at a cost of 2.7 pp of accuracy. Counterfactual tests still show residual dependence on age through proxies. SHAP attributions were stable across retrains (Spearman 0.97). The serving path met a 2 s median for a 10,000-request burst on one 4-vCPU node (1.55 s; p95 2.89 s, p99 3.02 s), and SHAP is the main cost. Reinforcement learning added 2.6 pp of revenue and 1.1 CSI points at a small accuracy cost. It did not reduce six-month drift and did no better than a contextual bandit.

On 500,000 simulated transactions per seed, accuracy within  $\pm 5\%$  rose from 11.2% to 34.0%, the standard deviation of relative error fell 70%, and simulated CSI rose 30 points. Accurate, explainable, and fairer pricing looks achievable without removing human oversight. These results depend on the generator and on assumed customer-response models, and

they do not support the reinforcement learning component. They are preliminary and need validation on real data.

## 7. Future Scope

### Limitations

- **Simulated data:** the evaluation used simulated multi-domain data rather than live deployments, so real-world conditions may add variability. All reported gains are preliminary.
- **Simulated satisfaction scores:** CSI and revenue come from simulated customer-response models. The 3.5-point explanation uplift and the acceptance and leakage parameters are assumptions, and the CSI and revenue gains mostly follow from them.
- **Generator dependence and tuning:** Absolute numbers depend on the generator. Price noise caps accuracy at 63.7%, the static baseline is a simple rate card, and the generator is near-additive, which is why trees match the linear model. The customer-response and reward settings (willingness-to-pay mean,  $\lambda$ s, and a reward measured against the unadjusted quote) were changed in pilot runs after an earlier setting led the agent to undercut prices. Other generator parameters were fixed beforehand. Hyperparameters were tuned once on seed 0, and seed-to-seed variation is small. Parity is measured against the cost-justified price, which a real deployment does not observe.
- **Reinforcement learning scope:** RL was tested only in simulation and did not reduce drift. Its effect on real customers through price exploration is untested.
- **Limited fairness coverage:** Fairness was checked across region, service type, age band, client type, and their intersections, but against one injected bias. Dependence on age through proxies remains.
- **Privacy trade-offs:** the effect of privacy controls on model accuracy was not evaluated.
- **Latency:** Latency was measured on one 4-vCPU node with in-process requests. Scaling across nodes, network effects, and disk or network throughput were not measured.
- **Compliance and features:** regulatory nuances differ across regions and require custom rule sets, and performance depends on high-quality feature extraction.
- **Untested industries:** e-commerce and legal applications were not evaluated.

### Benefits over earlier approaches

Relative to static schedules, the framework gives higher accuracy, lower variance, and price-level explanations. Relative to revenue-focused dynamic pricing, it adds governance, fairness monitoring, and bounded learning.

### Future Directions

- **Validation on real data:** test on a public or partner pricing dataset, and pilot in shadow mode before live use.
- **User study of explanations:** measure whether price composition and confidence intervals change customers' trust and understanding, replacing the simulated CSI.
- **Causal inference integration:** understand cause-effect relationships between service attributes and price outcomes for more robust decisions.

- **Market elasticity modeling:** incorporate demand sensitivity into dynamic pricing to optimize revenue without undermining fairness.
- **Federated learning architectures:** enable cross-organization model training without sharing sensitive data.
- **Real-time fairness-aware optimization:** dynamically adjust price outputs to maintain parity under evolving conditions.
- **Adaptive feature selection:** automatically identify the most predictive and fair features in high-dimensional service datasets.

## References

- [1] Kahneman D, Knetsch JL, Thaler RH. Fairness as a constraint on profit seeking: entitlements in the market. *Am Econ Rev.* 1986;76(4):728-41.
- [2] Barocas S, Selbst AD. Big data's disparate impact. *Calif Law Rev.* 2016;104(3):671-732.
- [3] den Boer AV. Dynamic pricing and learning: historical origins, current research, and new directions. *Surv Oper Res Manag Sci.* 2015;20(1):1-18.
- [4] Ban GY, Keskin NB. Personalized dynamic pricing with machine learning: high-dimensional features and heterogeneous elasticity. *Manage Sci.* 2021;67(9):5549-68. doi:10.1287/mnsc.2020.3680
- [5] Li L, Chu W, Langford J, Schapire RE. A contextual-bandit approach to personalized news article recommendation. In: *Proceedings of the 19th International Conference on World Wide Web.* New York: ACM; 2010. p. 661-70.
- [6] Sutton RS, Barto AG. *Reinforcement learning: an introduction.* 2nd ed. Cambridge (MA): MIT Press; 2018.
- [7] Chen L, Mislove A, Wilson C. An empirical analysis of algorithmic pricing on Amazon Marketplace. In: *Proceedings of the 25th International Conference on World Wide Web.* New York: ACM; 2016. p. 1339-49. doi:10.1145/2872427.2883089
- [8] Dwork C, Hardt M, Pitassi T, Reingold O, Zemel R. Fairness through awareness. In: *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference.* New York: ACM; 2012. p. 214-26.
- [9] Hardt M, Price E, Srebro N. Equality of opportunity in supervised learning. In: *Advances in Neural Information Processing Systems* 29. 2016.
- [10] Kusner MJ, Loftus J, Russell C, Silva R. Counterfactual fairness. In: *Advances in Neural Information Processing Systems* 30. 2017.
- [11] Kleinberg J, Mullainathan S, Raghavan M. Inherent trade-offs in the fair determination of risk scores. In: *8th Innovations in Theoretical Computer Science Conference (LIPIcs; vol. 67).* 2017. p. 43:1-43:23.
- [12] Chouldechova A. Fair prediction with disparate impact: a study of bias in recidivism prediction instruments. *Big Data.* 2017;5(2):153-63.
- [13] Doshi-Velez F, Kim B. Towards a rigorous science of interpretable machine learning. *arXiv:1702.08608 [Preprint].* 2017.
- [14] Lundberg SM, Lee SI. A unified approach to interpreting model predictions. In: *Advances in Neural Information Processing Systems* 30. 2017.

- [15] Friedman JH. Greedy function approximation: a gradient boosting machine. *Ann Stat.* 2001;29(5):1189-232. doi:10.1214/aos/1013203451
- [16] Snoek J, Larochelle H, Adams RP. Practical Bayesian optimization of machine learning algorithms. In: *Advances in Neural Information Processing Systems* 25. 2012.
- [17] Watkins CJCH, Dayan P. Q-learning. *Mach Learn.* 1992;8(3-4):279-92. doi:10.1007/BF00992698
- [18] Mnih V, Kavukcuoglu K, Silver D, Rusu AA, Veness J, Bellemare MG, et al. Human-level control through deep reinforcement learning. *Nature.* 2015;518(7540):529-33. doi:10.1038/nature14236
- [19] Dwork C, Roth A. The algorithmic foundations of differential privacy. *Found Trends Theor Comput Sci.* 2014;9(3-4):211-407. doi:10.1561/04000000042
- [20] Efron B, Tibshirani RJ. *An introduction to the bootstrap.* New York: Chapman & Hall; 1993.
- [21] Levene H. Robust tests for equality of variances. In: Olkin I, et al., editors. *Contributions to probability and statistics.* Stanford (CA): Stanford University Press; 1960. p. 278-92.