Quantile Regression with Large Language Models for Price Prediction

Nikhita Vedula[1*] **Dushyanta Dhyani**[1*] **Laleh Jalali**[1]

Boris Oreshkin[1] **Mohsen Bayati**[1,2] **Shervin Malmasi**[1]

[1] Amazon.com, Inc. [2] Stanford University
{veduln, dhyanidd, lalehjal, oreshkin, bayatim, malmasi}@amazon.com

Abstract

Large Language Models (LLMs) have shown promise in structured prediction tasks, including regression, but existing approaches primarily focus on point estimates and lack systematic comparison across different methods. We investigate probabilistic regression using LLMs for unstructured inputs, addressing challenging text-to-distribution prediction tasks such as price estimation where both nuanced text understanding and uncertainty quantification are critical. We propose a novel quantile regression approach that enables LLMs to produce full predictive distributions, improving upon traditional point estimates. Through extensive experiments across three diverse price prediction datasets, we demonstrate that a Mistral-7B model fine-tuned with quantile heads significantly outperforms traditional approaches for both point and distributional estimations, as measured by three established metrics each for prediction accuracy and distributional calibration. Our systematic comparison of LLM approaches, model architectures, training approaches, and data scaling reveals that Mistral-7B consistently outperforms encoder architectures, embedding-based methods, and few-shot learning methods. Our experiments also reveal the effectiveness of LLM-assisted label correction in achieving human-level accuracy without systematic bias. Our curated datasets are made available[1] to support future research.

1 Introduction

Large Language Models (LLMs) have demonstrated remarkable capabilities across a wide range of tasks, including unstructured document processing (Zou et al., 2025). Going beyond their original purpose of text generation (Brown et al., 2020), they have recently been extended to structured numerical prediction tasks such as time series forecasting (Das et al., 2024). Recent research has shown their effectiveness in regression tasks (Garg et al., 2022; Vacareanu et al., 2024a), where they have been found to approximate numerical mappings with surprisingly strong accuracy when prompted with in-context examples.

The intersection of LLMs and regression is particularly important for the longstanding task of text regression, where unstructured language must be mapped reliably to numeric outputs (Bitvai and Cohn, 2015). Traditional regression models often struggle with applications where crucial information lies in unstructured text, such as product descriptions or financial reports (Zhang et al., 2024; Gu et al., 2024), requiring rich text understanding. This is crucial in domains like product pricing where heterogeneous features across categories (e.g., screen technology for televisions versus mileage for cars) make traditional unified feature representations inadequate for capturing category-specific dynamics.

Existing work on regression with LLMs has explored three main approaches to address these challenges: fine-tuning LLMs for specific numeric prediction tasks (Morgan and Jacobs, 2024), using LLM embeddings as features for downstream regression models (Imperial, 2021; Tang et al., 2024), and leveraging in-context learning for zero-shot or few-shot numeric estimation (Vacareanu et al., 2024a). However, these approaches, with the exception of a few (Gruver et al., 2023; Qiu et al., 2024), focus on point estimates, overlooking a key limitation: the inability to quantify uncertainty. Many real-world applications, such as price prediction, demand forecasting, and financial risk assessment, inherently require probabilistic outputs rather than single-value predictions (Qiu et al., 2024). Probabilistic modeling is essential in these applications to capture uncertainty and normal variation, mitigate risks, and improve decision-making (Gu et al., 2024). Current work has neither explored proba-

[*] Equal contribution.

[1] https://github.com/vnik18/llm-price-quantile-reg/

Findings of the Association for Computational Linguistics: ACL 2025, pages 12396–12415
July 27 - August 1, 2025

bilistic regression using LLMs in-depth nor evaluated the trade-offs between different LLM regression approaches in a single study.

This paper presents the first study of probabilistic regression using LLMs to encode unstructured text inputs, and take a step towards a systematic investigation of LLM-based regression methods. We center our study on price prediction, a task that requires both nuanced interpretation of free-form text inputs and accurate distribution estimation. Understanding the complete price distribution is essential in financial contexts where accurately modeling tail behavior is critical for effective risk management.

In sum, this paper makes three key contributions. First, we propose a novel LLM-based quantile regression approach that produces full distributions with strong calibration while maintaining sharp prediction intervals, and improves point estimation accuracy compared to traditional approaches. Qualitative analysis shows that our model produces well-calibrated distributions that adapt to different price ranges and uncertainties across datasets, with tighter distributions for standardized products and appropriately wider distributions for items with more price variability. Second, we systematically compare different LLM architectures (decoder-only vs. encoder-only vs. traditional ML on text embeddings vs. in-context learning), multiple loss functions (squared error vs. pinball), and various data scales, while investigating training data contamination. We show that fine-tuning decoder models (e.g., Mistral-7B) outperforms other approaches. Our results confirm that model size, data scaling, and clean training sets all play critical roles in robust, generalizable LLM-based probabilistic regression. Third, focusing on the task of price estimation, we release three curated datasets (Amazon products, Craigslist used cars, Used boats) with standardized splits.

2 Related Work

Distributional, Quantile and Text Regression: Quantile regression, introduced by Koenker and Bassett (1978), extends beyond traditional pointwise prediction methods by characterizing the entire conditional distribution of the target variable through estimation of conditional quantiles at different probability levels (Kneib et al., 2023). Unlike ordinary least squares regression which minimizes squared errors, quantile regression uses the *pinball loss*, making it robust to outliers and capable of capturing heterogeneous effects across the distribution. This approach is especially valuable in healthcare (Gürlek et al., 2024), finance, and economics (Dichev et al., 2023; Gu et al., 2024), e.g. where modeling the full distribution helps capture both typical and extreme valuations. Text Regression is a natural language processing task involving predicting continuous numerical values from unstructured text input with applications in domains like financial forecasting, election prediction, and box office revenue estimation (Bitvai and Cohn, 2015; Dereli and Saraclar, 2019).

LLM-based Distribution Estimation: Recent work has begun exploring LLMs' capabilities for distributional prediction tasks. Gruver et al. (2023) demonstrated probabilistic forecasting by mapping LLM token predictions to continuous distributions, while Qiu et al. (2024) developed a fine-tuned LLM that outputs discretized probability ranges for energy forecasting. However, these approaches focus primarily on structured numerical sequences rather than deriving distributions from unstructured text inputs, relying on either zero-shot prompting with specialized number formatting or domain-specific fine-tuning with predefined output ranges. Our work addresses the broader challenge of predicting full probability distributions directly from unstructured text through quantile regression. We explore and compare several approaches for text-to-distribution models: (1) computing LLM embeddings then feeding them into a separate quantile prediction model, (2) extracting embeddings and approximating distributions using outcomes of "neighboring" embeddings in training data, and (3) our proposed approach of attaching multi-quantile heads directly to the LLM's last hidden layer and fine-tuning the entire architecture end-to-end with smoothed pinball loss. To our knowledge, this is the first paper to explore price prediction as a text-to-distribution NLP task rather than as a point value prediction task, with our approach allowing the LLM's representation layers to adapt in capturing distribution-relevant features.

Regression with LLMs Embeddings: A prevalent approach involves using pre-trained LLMs to generate text embeddings for downstream regression tasks (Imperial, 2021; Gu et al., 2024). Tang et al. (2024) provide evidence that LLM embeddings maintain strong regression performance even as input dimensionality increases, where traditional feature engineering methods typically fail.

In-Context Learning for Regression: Recent research reveals the surprising capability of LLMs like GPT-4 and Claude to perform regression through in-context learning (Garg et al., 2022; Vacareanu et al., 2024a). Their work shows that regression accuracy generally improves with the number of in-context examples provided. In the domain of real estate, Chen and Si (2024) confirm this behavior for price prediction tasks. Lukasik et al. (2024) further advance this direction with Regression-Aware Inference with LLMs (RAIL), enhancing zero-shot numeric prediction through optimized decoding strategies. Their approach demonstrates that careful calibration of sampling parameters can significantly improve regression performance without requiring model fine-tuning.

Fine-Tuning LLMs for Regression: Recent studies demonstrate the effectiveness of fine-tuning LLMs for regression tasks across diverse domains. Morgan and Jacobs (2024) fine-tune a LLaMA-based model, achieving performance comparable to specialized domain models in chemical property prediction. Zhang et al. (2024) show that fine-tuned BERT-based models can effectively predict house prices from unstructured property descriptions, outperforming baselines that rely solely on structured features. Song et al. (2024) present a framework that converts various input formats into text and fine-tunes LLMs as universal end-to-end regressors, demonstrating strong cross-domain performance.

While these studies demonstrate LLMs' capabilities for regression, two gaps exist in the literature. Most critically, existing work largely overlooks probabilistic regression techniques that could leverage LLMs' rich understanding of textual data. Additionally, prior work lacks unified comparisons between fine-tuning, embedding-based methods, and in-context learning under similar conditions. We primarily address the probabilistic gap by introducing novel methods for quantile regression that enable uncertainty-aware predictions from language models, while also providing a comparative analysis of different LLM-based regression approaches.

3 Price Estimation Datasets

We experiment upon three price prediction datasets from different domains publicly available for research use: Amazon Products (Ni et al., 2019), Craigslist Used Cars listings,[2] and European Boat Sales.[3] Initial manual inspection revealed numerous instances with erroneous prices, i.e., unreasonably high or low relative to the items being sold (examples are provided in Table 7 of the Appendix). To address this, we employed the Claude-3.5-Sonnet LLM (Anthropic, 2024) in a zero-shot manner to identify and remove rows with incorrect prices from all dataset splits. Details of this process and our human evaluation (>94% agreement between human and LLM judgments) verifying that it neither removed difficult instances nor introduced bias are provided in Appendix A.1. Table 1 presents the dataset distribution and number of samples removed for each dataset.

Dataset	Train	Val	Test	Removed
Amazon Products	500K	15K	15K	100K
Craigslist Used Cars	350K	19K	19K	10K
Boats	8K	450	450	1K

Table 1: Dataset Size Distribution

4 LLM-Based Quantile Regression

4.1 Problem Statement

Given a random variable X with realizations $X = \mathbf{x} \in \mathcal{X}$ representing unstructured textual input (e.g., a product title or description) and other structured attributes, our goal is to predict the conditional distribution $F_{Y|X}(\cdot|\mathbf{x})$, where $y \in \mathbb{R}$ is a numeric outcome, such as the price of a product. More formally, we aim to learn a function $f(\cdot;\Theta) : \mathcal{X} \to \mathcal{F}$ that maps inputs to conditional distributions, where $\mathcal{F}$ is the space of cumulative distribution functions on $\mathbb{R}$. Here, Θ is a multi-dimensional parameter, such as the weights of an LLM. We represent these distributions through a vector of conditional quantiles $\mathbf{q}_{\boldsymbol{\tau}}(\mathbf{x}) = (\hat{q}_{\tau_1}(\mathbf{x}), \ldots, \hat{q}_{\tau_K}(\mathbf{x}))$ for K, prespecified quantiles $\boldsymbol{\tau} = (\tau_1, ..., \tau_K) \in (0,1)^K$. The optimal parameters Θ^*, are learned by minimizing:

$$\Theta^* = \arg\min_{\Theta} \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}[\mathcal{L}(f(\mathbf{x};\Theta), y)]$$

where $\mathcal{D}$ is the underlying data distribution and $\mathcal{L}$ is a proper loss function for probabilistic forecasts. Figure 1 shows the end-to-end training and inference pipelines of our proposed approach for distributional price prediction using LLM-based quantile regression.

[2] https://kaggle.com/datasets/austinreese/craigslist-carstrucks-data

[3] https://www.kaggle.com/datasets/karthikbhandary2/boat-sales

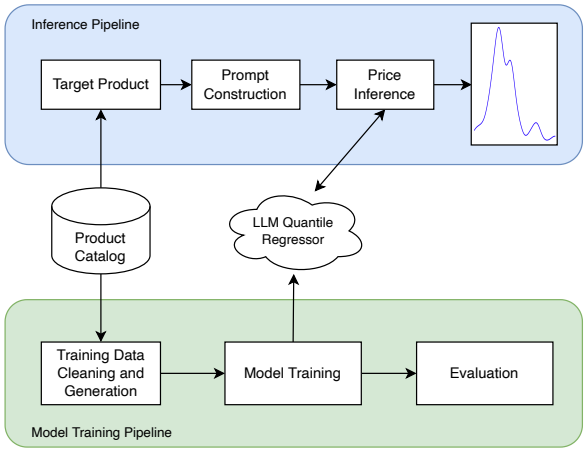

Figure 1: End-to-end training and inference pipelines of our proposed framework.

4.2 Quantile Regression Head & Pinball Loss

We propose adding a quantile regression head to both decoder-only and encoder-only LM architectures. Let $f(\cdot;\Theta)$ be the decoder-only LLM parameterized by Θ. It takes a tokenized version of the input text ($\mathbf{x}$) with sequence length T, denoted by $\tilde{\mathbf{x}} = (x_1, \ldots, x_T)$, and produces hidden states $H \in \mathbb{R}^{T\times D}$ where D is hidden state dimension. We then extract the final hidden state $\mathbf{h}_T = H[T,:]$ as a summary of the sequence. For encoder models, $\mathbf{h}_T$ is the [CLS] token representation. We then replace the language modeling head with a quantile regression head $g(\cdot;\phi)$, to predict K quantiles $\hat{\mathbf{q}} = (\hat{q}_{\tau_1}, \ldots, \hat{q}_{\tau_K})$ given $\mathbf{h}_T$, with more details in appendix B.1. Therefore, our model structure is:

$$\tilde{\mathbf{x}} = \text{tokenized } \mathbf{x}\,, \tag{1}$$

$$H = f(\tilde{\mathbf{x}};\Theta) \in \mathbb{R}^{T\times D}\,, \tag{2}$$

$$\mathbf{h}_T = H[T,:]\,, \tag{3}$$

$$\hat{\mathbf{q}} = g\big(\mathbf{h}_T;\phi\big) \in \mathbb{R}^K\,. \tag{4}$$

The pinball loss, also known as the quantile loss, enables asymmetric penalization of over- and under-predictions. For predicted quantiles $\hat{\mathbf{q}}$ obtained in (4), given quantile levels $\boldsymbol{\tau}$, and ground truth y, we implement the pinball loss as $\mathcal{L}_{\boldsymbol{\tau}}(\hat{\mathbf{q}}, y) = (1/K)\sum_{k=1}^{K}\mathcal{L}_{\tau_k}(\hat{q}_{\tau_k}, y)$, where,

$$\mathcal{L}_\tau(\hat{q}_\tau, y) = \tau(y - \hat{q}_\tau) + \mathsf{ReLU}(\hat{\mathsf{q}}_\tau - \mathsf{y})\,. \tag{5}$$

For improved optimization stability, we employ a smoothed variant:

$$\mathcal{L}^\alpha_\tau(\hat{q}_\tau, y) = \tau(y - \hat{q}_\tau) + \alpha\cdot\mathsf{SoftPlus}_\alpha(\hat{\mathsf{q}}_\tau - \mathsf{y}),$$

where $\mathsf{SoftPlus}_\alpha(\mathsf{x}) = \alpha\log(1 + \mathsf{e}^{\mathsf{x}/\alpha})$ provides a differentiable approximation to ReLU as $\alpha \to 0^+$.

5 Experimental Setup

Quantile Levels and Point Prediction: For all models that predict distributions, we take $K = 200$ and divide the interval $(0, 1)$ into K equal-length sub-intervals to obtain $\boldsymbol{\tau}$. We discuss the impact of varying K and the smoothing parameter α in Appendix C. We use models that produce a distribution both for generating probabilistic outputs and for evaluating point predictions. In the latter case, we take the predicted quantile at $\tau = 0.5$ as the point estimate. Additionally, we include baseline models trained solely with traditional squared error loss, using their direct predictions for comparison.

Baselines Using LLM Embeddings: Text features (title, description, attributes) are concatenated with appropriate field markers and converted to embeddings using the Qwen2-7B-instruct embedding model (Chu et al., 2024). All baseline models using these embeddings are denoted by the "Qwen-7B-Emb" prefix. These embeddings serve as input features for five models: Ridge Regression and XGBoost for point estimation, Quantile Regression (with two hidden layers) for distribution prediction, trained on log-transformed target,[4] and two nearest neighbor-based distribution prediction approaches. The first nearest neighbor model (kNN) predicts distributions by using the empirical distribution of target values from selected neighbors in the training set, while the second variant employs a radius-based selection criterion (rkNN) with a minimum neighbor requirement. All hyperparameters are selected using 5-fold cross-validation.

Fine-tuned LMs with Quantile Head: We fine-tune Mistral-7B (Mistral, 2023), Phi-3B (Abdin et al., 2024), Qwen-500M (Bai et al., 2023a) and XLM-RoBERTa (Conneau et al., 2019) models with a quantile regression head.

In-context Learning: We evaluate two state-of-the-art LLMs, Claude-3.5-Sonnet and Nova Pro (Anthropic, 2024; Amazon, 2024), both zero-shot and few-shot. For few-shot learning, we implement three example selection strategies: (i) random sampling; (ii) category-based stratified sampling and (iii) similar item sampling based on cosine similarity of Qwen2-7B embeddings. The latter two leverage domain similarity for potentially better price estimation (prompts in Figure 7).

[4] In all three data sets since the target was price, we used its log-transformed prices to handle the wide range of values in our datasets during training.

Evaluation Metrics: We use two sets of metrics. The first set evaluates point price estimates and includes: (i) **Mean Absolute Percentage Error (MAPE)**, (ii) **Weighted Absolute Percentage Error (WAPE)**, with weight = 1 and (iii) **Mean Percentage Error (MPE)**.

The second set of metrics measures the distributional quality of the predicted quantiles $\hat{\mathbf{q}}_{\boldsymbol{\tau}}(\mathbf{x}_i) = (\hat{q}_{\tau_1}(\mathbf{x}_i) \leq \cdots \leq \hat{q}_{\tau_K}(\mathbf{x}_i))$ for each input $\mathbf{x}_i$. These metrics include:

(i) Calibration Error (CE) measures how well predicted quantiles match their theoretical coverage. $\text{CE} = (1/K)\sum_{k=1}^{K} |\widehat{\text{coverage}}(\tau_k) - \tau_k|$, where $\widehat{\text{coverage}}(\tau_k)$ is the empirical fraction of true values in the test set, below the τ_k quantile.

(ii) Continuous Ranked Probability Skill Score (CRPSS) measures the integrated squared difference between predicted and true cumulative distribution functions.

(iii) Relative Confidence Interval Width (RCIW) measures the average width or tightness of predicted intervals relative to the true value.

For each metric, we report 95% confidence intervals with bootstrap resampling (1000 iterations): $\text{CI}_{95\%}(M) = [\hat{M}_{(0.025)}, \hat{M}_{(0.975)}]$ where $\hat{M}_{(q)}$ denotes the q-th quantile of the bootstrap distribution of metric M. Further details about all metrics and the training process are presented in Appendix C.

6 Results

6.1 Point Regression Results

Table 2 lists our main point regression results, comparing all models across three datasets.

Fine-tuned LLMs Outperform Traditional Models. The fine-tuned Mistral-7B-Quantile model notably outperforms other approaches across all datasets. For the Amazon Products dataset, Mistral-7B achieves a MAPE of 16.86%, substantially lower than the best traditional baselines (Qwen7B-Emb+RkNN-Q) at 42.68%. This pattern is particularly pronounced in the Used Cars dataset, where Mistral-7B's MAPE of 6.3% represents an order of magnitude improvement over traditional approaches, which show MAPEs up to 235%. This highlights the importance of rich text understanding for price regression. The MPE results indicate that traditional approaches tend to systematically underestimate prices, with negative biases ranging from -24% to -135% across datasets. In contrast, Mistral-7B shows minimal systematic bias, with

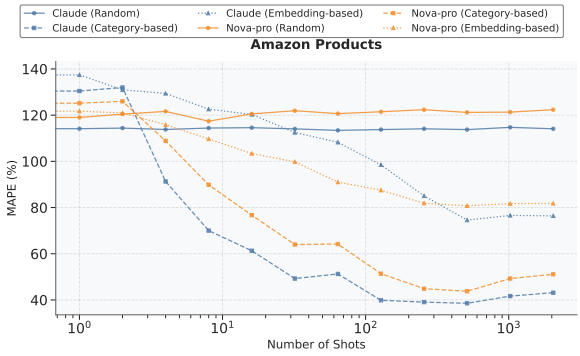

Figure 2: Few-shot learning performance of Claude-3.5-Sonnet and Nova-Pro LLMs on Amazon Products data.

MPE values close to zero: -0.88% for Amazon Products and 0.185% for Used Cars.

Better Estimates via Quantile Regression. Comparing the best model (Mistral-7B-Quantile) to a version with a point regression head shows a substantial improvement in all metrics, confirming that the median of the quantile regression distribution is a better estimate than pointwise regression.

Decoder-only architectures are better than Encoders. Comparing encoder-only (XLM-RoBERTa) and decoder-only (Mistral-7B, Phi-3B) architectures reveals interesting patterns. The larger XLM-RoBERTa model consistently outperforms its base variant, but both lag behind Mistral-7B across all datasets. On the Amazon Products dataset, XLM-RoBERTa Large achieves a MAPE of 36.52%, while XLM-RoBERTa Base shows 41.99% – both substantially higher than Mistral-7B's 16.86%. Interestingly, the performance gap between architectures varies by dataset. For Used Cars, both XLM-RoBERTa variants perform relatively well (MAPE: 11.45% - 12.84%) compared to their performance on other datasets, though still trailing Mistral-7B (6.3%). The Boats dataset shows the smallest architecture-based performance difference, suggesting that dataset characteristics may influence the relative advantages of different architectures.

Few-shot learning Underperforms. Figure 2 shows the results of few-shot learning approaches across our three datasets. Even with the best-performing category-based sampling strategy and optimal shot count, both Claude and Nova-pro lag behind fine-tuned Mistral-7B by more than 15% for Amazon Products and Used Boats, and by more than 200% for Used Cars. Our experiments also re-

Dataset	Model	MAPE (%) ↓		MPE (%) ↓		WAPE (%) ↓	
		Value	95% CI	Value	95% CI	Value	95% CI
Amazon Products	Mistral-7B-Point	20.81	[20.13, 21.22]	-3.40	[-3.53, -3.30]	22.81	[22.45, 24.67]
	Mistral-7B-Quantile	**16.86**	[16.15, 17.71]	**-0.88**	[-1.23, -0.55]	**18.32**	[17.83, 18.83]
	XLM-R Base-Quantile	41.99	[40.34, 43.86]	-21.73	[-23.72, -20.00]	40.27	[39.09, 41.45]
	XLM-R Large-Quantile	36.52	[34.64, 38.49]	-15.18	[-17.22, -13.22]	37.51	[36.20, 38.74]
	Qwen-500M-Quantile	39.19	[38.01, 40.36]	-6.33	[-7.73, -5.07]	43.15	[42.07, 44.10]
	Phi-3B-Quantile	34.17	[33.14, 35.27]	-5.41	[-6.65, -4.17]	38.17	[37.11, 39.29]
	Qwen-7B-Emb+Ridge	58.97	[57.78, 60.26]	30.36	[32.02, -29.02]	52.72	[51.93, 53.41]
	Qwen-7B-Emb+XGBoost	63.16	[62.22, 64.30]	-32.57	[-33.98, -31.50]	58.01	[57.27, 58.88]
	Qwen-7B-Emb+Quantile	77.97	[76.89, 79.10]	-27.44	[-28.99, -25.88]	76.3	[75.69, 76.99]
	Qwen-7B-Emb+kNN-Quantile	46.86	[45.88, 47.83]	-10.53	[-11.68, -9.27]	54.05	[52.95, 55.04]
	Qwen-7B-Emb+RkNN-Quantile	42.68	[41.66, 43.90]	-10.33	[-11.65, -9.06]	48.03	[46.88, 49.21]
	Claude-3.5-Sonnet (512 category-based shots)	38.50	[36.70, 39.10]	14.32	[14.29, 14.41]	41.40	[40.20, 42.16]
	Nova-Pro (512 category-based shots)	43.77	[40.78, 45.01]	19.12	[18.79, 19.81]	48.13	[46.21, 49.33]
Used Cars	Mistral-7B-Point	9.76	[9.25, 10.67]	-5.40	[-5.89, -4.01]	12.79	[12.65, 13.32]
	Mistral-7B-Quantile	**6.30**	[6.06, 6.95]	**0.19**	[0.05, 0.31]	**5.40**	[5.29, 5.51]
	XLM-R Base-Quantile	11.45	[10.68, 12.44]	-5.71	[-6.62, -4.87]	8.89	[8.62, 9.23]
	XLM-R Large-Quantile	12.84	[12.41, 13.37]	-9.70	[-10.23, -9.24]	10.46	[10.22, 10.74]
	Qwen-500M-Quantile	23.49	[20.61, 26.71]	-4.56	[-8.03, -1.48]	15.93	[15.48, 16.40]
	Phi-3B-Quantile	52.79	[51.83, 53.89]	47.09	[45.90, 48.14]	74.91	[74.50, 75.32]
	Qwen-7B-Emb+Ridge	40.46	[37.95, 43.42]	-18.04	[-21.12, -15.44]	23.04	[22.49, 23.41]
	Qwen-7B-Emb+XGBoost	39.70	[38.10, 41.75]	-16.09	[17.96, -14.41]	26.13	[25.80, 26.60]
	Qwen-7B-Emb+Quantile	235.92	[221.57, 249.55]	-192.67	[-206.41, -178.24]	58.41	[57.85, 58.85]
	Qwen-7B-Emb+kNN-Quantile	79.72	[73.18, 86.87]	-59.39	[-66.09, -52.59]	26.81	[26.33, 27.29]
	Qwen-7B-Emb+RkNN-Quantile	58.18	[52.46, 64.02]	-40.92	[-46.86, -35.39]	21.37	[20.93, 21.84]
	Claude-3.5-Sonnet (2048 random shots)	275.00	[269.12, 280.09]	189.19	[175.21, 195.62]	53.34	[50.78, 56.09]
	Nova-Pro (1024 random shots)	219.67	[167.42, 231.91]	173.07	[156.12, 189.07]	46.44	[42.13, 48.71]
Boats	Mistral-7B-Point	24.01	[23.82, 24.29]	4.10	[2.30, 7.45]	25.82	[24.20, 27.39]
	Mistral-7B-Quantile	**21.20**	[20.50, 23.39]	2.19	[1.59, 6.65]	**23.96**	[20.68, 27.69]
	XLM-R Base-Quantile	22.17	[20.26, 24.47]	**0.58**	[-2.69, 3.52]	23.59	[20.12, 26.78]
	XLM-R Large-Quantile	22.67	[20.85, 24.55]	-4.51	[-7.43, -1.78]	31.05	[24.31, 37.99]
	Qwen-500M-Quantile	62.27	[56.49, 69.12]	16.98	[8.6, 24.73]	77.23	[73.20, 80.75]
	Phi-3B-Quantile	73.83	[71.45, 76.02]	72.89	[70.32, 75.31]	93.64	[92.41, 94.64]
	Qwen-7B-Emb+Ridge	30.77	[28.06, 33.89]	-7.52	[-11.85, -3.81]	28.77	[24.57, 33.38]
	Qwen-7B-Emb+XGBoost	44.56	[40.12, 49.19]	-12.84	[-17.86, -6.93]	42.35	[38.02, 46.85]
	Qwen-7B-Emb+Quantile	131.03	[110.78, 158.77]	-67.61	[-99.21, -44.88]	82.21	[78.75, 84.98]
	Qwen-7B-Emb+kNN-Quantile	77.68	[67.39, 88.29]	-32.36	[-43.49, -20.56]	63.39	[58.06, 67.78]
	Qwen-7B-Emb+RkNN-Quantile	70.96	[61.86, 80.72]	-28.67	[-39.61, -18.10]	56.80	[51.81, 61.44]
	Claude-3.5-Sonnet (2048 random shots)	30.00	[28.97, 31.28]	17.32	[15.16, 19.23]	29.36	[26.16, 30.09]
	Nova-Pro (2048 random shots)	61.01	[55.54, 64.76]	23.22	[21.16, 25.91]	48.79	[45.03, 50.71]

Bold values indicate best performance for each metric and dataset. The ↓ indicates that lower metric values are better.

Table 2: Model point-estimate performance comparison, using median as the point estimate for quantile regression models. For the few-shot Claude-3.5 and Nova-Pro LLMs, we only show the optimal few shot example selection strategy and the corresponding number of shots that gave the best results.

veal that increasing the number of examples beyond a certain point starts degrading model performance. This substantial performance gap indicates that for precise price prediction, fine-tuning yields considerably better results than carefully crafted few-shot approaches. It would appear that the complex relationships between rich textual data and prices require more thorough model adaptation than what can be achieved through in-context learning.

6.2 Distributional Regression Results

Table 3 lists our main distributional regression results, comparing various decoder and encoder models, and embedding baselines.

Larger Fine-tuned LMs achieve the best results. As earlier, the fine-tuned Mistral-7B-Quantile model is a consistently strong performer across all distributional metrics and datasets. It achieves the best CRPSS scores ranging between 0.73-0.92 on all three datasets, indicating the high quality of the model's probabilistic predictions relative to

Dataset	Model	CE ↓		CRPSS ↑		RCIW@95%CI ↓	
		Value	95% CI	Value	95% CI	Value	95% CI
Amazon Products	Mistral-7B-Quantile	0.042	[0.039, 0.044]	**0.75**	[0.74, 0.76]	**0.92**	[0.92, 0.93]
	XLM-R Base	0.060	[0.057, 0.064]	0.49	[0.48, 0.51]	2.03	[1.99, 2.09]
	XLM-R Large	0.040	[0.037, 0.043]	0.53	[0.51, 0.55]	1.52	[1.48, 1.57]
	Qwen-500M-Quantile	0.055	[0.051, 0.059]	0.47	[0.46, 0.49]	2.89	[2.83, 3.00]
	Phi-3B-Quantile	0.041	[0.036, 0.046]	0.53	[0.52, 0.54]	2.33	[2.27, 2.38]
	Qwen-7B-Emb+Quantile	0.045	[0.042, 0.048]	0.03	[0.01, 0.04]	16.76	[16.59, 16.92]
	Qwen-7B-Emb+kNN-Quantile	**0.01**	[0.006, 0.013]	0.34	[0.31, 0.37]	6.14	[6.05, 6.26]
	Qwen-7B-Emb+RkNN-Quantile	**0.01**	[0.007, 0.012]	0.42	[0.39, 0.44]	6.12	[6.00, 6.29]
Used Cars	Mistral-7B-Quantile	0.054	[0.051, 0.055]	**0.92**	[0.91, 0.92]	**0.20**	[0.20, 0.21]
	XLM-R Base	0.157	[0.155, 0.159]	0.80	[0.79, 0.81]	1.02	[1.01, 1.03]
	XLM-R Large	0.185	[0.183, 0.187]	0.80	[0.79, 0.81]	1.01	[1.00, 1.01]
	Qwen-500M-Quantile	0.160	[0.158, 0.162]	0.66	[0.65, 0.66]	4.70	[3.76, 5.50]
	Phi-3B-Quantile	0.395	[0.393, 0.397]	0.04	[0.03, 0.05]	0.99	[0.96, 1.04]
	Qwen-7B-Emb+Quantile	**0.020**	[0.018, 0.022]	0.01	[0.01, 0.02]	13.41	[12.84, 14.45]
	Qwen-7B-Emb+kNN-Quantile	0.024	[0.020, 0.028]	0.53	[0.52, 0.53]	3.90	[3.75, 4.09]
	Qwen-7B-Emb+RkNN-Quantile	0.022	[0.019, 0.026]	0.62	[0.63, 0.64]	2.64	[2.54, 2.80]
Boats	Mistral-7B-Quantile	0.076	[0.070, 0.084]	**0.73**	[0.67, 0.77]	1.28	[1.23, 1.35]
	XLM-R Base	0.047	[0.028, 0.066]	0.73	[0.70, 0.77]	1.54	[1.50, 1.59]
	XLM-R Large	0.042	[0.030, 0.051]	0.59	[0.45, 0.69]	1.68	[1.65, 1.72]
	Qwen-500M-Quantile	0.257	[0.237, 0.275]	0.18	[0.03, 0.44]	1.24	[1.17, 1.34]
	Phi-3B-Quantile	0.453	[0.445, 0.461]	0.21	[0.13, 0.35]	**0.68**	[0.62, 0.74]
	Qwen-7B-Emb+Quantile	0.034	[0.014, 0.056]	0.20	[0.09, 0.23]	33.37	[30.03, 36.55]
	Qwen-7B-Emb+kNN-Q	**0.021**	[0.011, 0.036]	0.28	[0.19, 0.38]	11.33	[10.41, 12.25]
	Qwen-7B-Emb+RkNN-Q	0.025	[0.013, 0.042]	0.39	[0.30, 0.46]	7.87	[7.03, 8.61]

Bold values indicate best performance. ↓ indicates that lower metric values are better, and ↑ indicates that higher are better.

Table 3: Model distribution prediction performance comparison across models and datasets. CE measures how well predicted quantiles match their theoretical coverage, CRPSS evaluates the probabilistic prediction quality relative to a reference, and RCIW measures the sharpness of the distribution prediction intervals.

the references. Mistral-7B also achieves the best RCIW score on Amazon Products and Used Cars data, and a competitive score on the Boats data, indicating sharp distributions and generally precise prediction confidence intervals. Overall, Mistral-7B is much more consistent and maintains a better balance across different metrics compared to the smaller models like Phi, Qwen-500M or XLM-RoBERTa. Both Mistral-7B and Qwen-7B embedding based variants show very low CE scores on all datasets, indicating that the predicted probabilities match their theoretical coverage very well.

LLMs produce Better Calibrated Distributions. Mistral-7B-Quantile model demonstrates strong calibration (low CE between 0.04-0.07) while maintaining better confidence intervals, suggesting that fine-tuned LLMs are inherently better at producing well-calibrated probability distributions. Qwen-7B embedding variants also achieve very low calibration errors (CE of 0.01 on Amazon Products and 0.02 on Used Cars), significantly outperforming smaller models like XLM-RoBERTa which has a CE of 0.04-0.06 on Amazon Products and 0.157-0.185 on Used Cars. The RCIW patterns also reveal interesting trade-offs. While embedding-based Qwen-7B variants achieve excellent calibration, they produce much wider confidence intervals with RCIW between 6.12-16.76 on Amazon Products. However, the Mistral-7B model fine-tuned for quantile regression can achieve both sharp predictions and good calibration, evidenced by its optimal RCIW scores while also maintaining high CRPSS.

Larger Data Leads to Better Distributions. Most models achieve overall better and more consistent distributional metric scores on the larger Amazon Products and Used Cars datasets compared to the much smaller Boats data. Mistral-7B-Quantile achieves tighter confidence intervals on Used Cars with an RCIW of 0.2 compared to its RCIW of 1.28 on the Boats data. Wider confidence intervals and more variable model performance on the Boats dataset highlight the detrimental impact of smaller sample sizes on probabilistic predictions.

Qualitative Analysis of Distributions. We show in Figure 3 the predicted probability distribution function of the prices by our fine-tuned Mistral-7B-Quantile model, smoothed using a Gaussian kernel. We show examples from all three datasets having

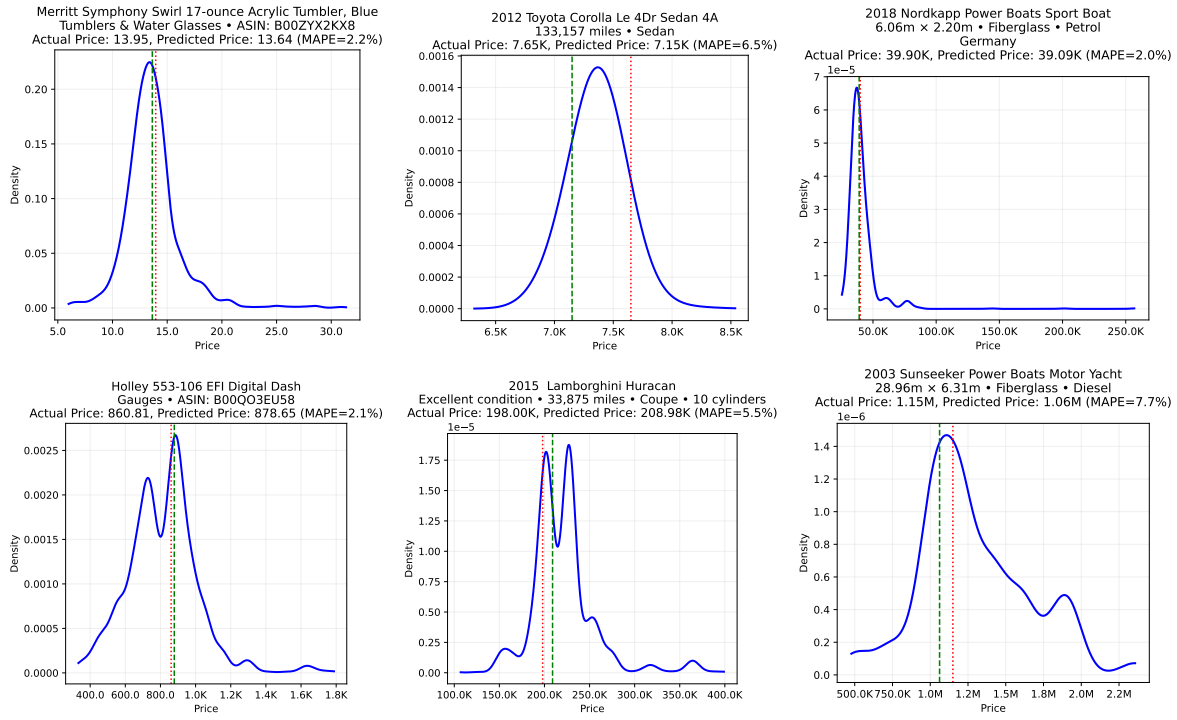

Figure 3: Probability density distribution of the prices predicted by the Mistral-7B-Quantile model across different datasets (blue curve). Each x-axis has a different scale. The red dotted line represents the ground truth price while the green dashed line is the predicted median price. As demonstrated, the model captures different distribution shapes including unimodal (top row), bimodal (bottom row), and right-skewed (right) distributions.

different MAPE values (additional examples are provided in Section D.2 of the appendix). In both the Amazon Products examples, the predicted median and actual ground truth prices are very close to each other, with the largest mode of the distribution centered around the ground truth price. We see wider distributions spanning larger price ranges for the higher priced Used Cars datasets, with the most distribution width and price uncertainty in the Boats datasets, possibly due to a greater price variability in this domain or a smaller training dataset.

6.3 Discussion

Theoretical Justification of Distributional Regression. While Table 2 shows consistent outperformance of distributional regression (Mistral-7B-Quantile) over point regression (Mistral-7B-Point) across all point metrics, we also theoretically discuss why multi-quantile LLM fine-tuning is superior to point-estimate fine-tuning in capturing uncertainty. Fine-tuning LLMs with Mean-Square Error (MSE) loss trains them to learn the conditional mean, ignoring higher-order moments and distributional shape. This is because the gradient is proportional to the raw error $(\hat{y} - y)$, and all corrections push predictions towards the conditional mean. On the other hand, Pinball loss for a quantile τ yields a consistent estimator of that specific quantile (Koenker and Bassett, 1978). For each observation, when the model under-predicts the τ-quantile, the gradient moves the prediction upward with weight τ. Conversely, for over-prediction, the gradient pushes downward, with weight 1-τ. In our multi-quantile approach, summing over multiple τ provides a discrete approximation to the integral of pinball losses over τ in (0,1). This integral corresponds to CRPS which is a strictly proper scoring rule for the entire distributions (Gneiting and Raftery, 2007), so minimizing it recovers the ground truth conditional distribution.

From a multi-task learning perspective, our approach benefits from shared representation across quantile predictions; each quantile level effectively functions as a related but distinct prediction task. The quantile approach yields better point estimates through two mathematical mechanisms: first, the $\tau = 0.5$ quantile loss's inherent robustness to outliers, and second, when training with multiple quantile levels simultaneously, the non-median quantile losses effectively serve as regularization terms for the median prediction task, constraining the model to perform well across the entire distribution.

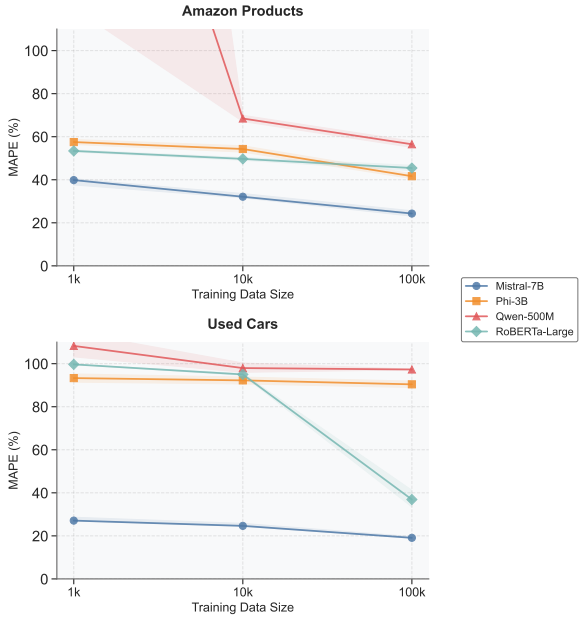

Figure 4: Impact of training data size on model MAPE on two datasets (y-axis scaled 0-100% for comparison).

Performance Breakdown by Category. Analysis of Mistral-7B's performance across different product categories in Amazon Products data reveals interesting patterns.The model excels in categories with standardized pricing structures, achieving MAPEs as low as 4.68% for Window Tinting Kits which also has a high price range spanning $200; and 5.39% for Keyrings & Keychains. While there are high performing categories with either narrow price ranges (e.g., Machine Screws: $7.35-$13.84) or well-defined market segments (e.g., Engine Management Systems), the model is also able to make good quality price predictions (MAPE within 6-12%) for widely diverse categories with high price ranges (e.g., MAPE of 10.43% for Custom Fit with a price range over $500, and 12.11% for Body with a price range over $400). More details can be found in Tables 8 and 9.

Performance Improvement with Training Data and Model Size. Figure 4 illustrates clear performance gains with increased training data. Mistral-7B-Quantile shows strong scaling benefits for the Amazon Products dataset, with MAPE decreasing from 39.84% at 1,000 samples to 24.3% at 100,000 samples. The Used Cars dataset exhibits similar scaling behavior, with MAPE reducing from 27.09% to 19.09% across the same range. Mistral-7B's superior performance over smaller models (Phi, Qwen and RoBERTa) across all metrics shows that model scale significantly impacts price estimation accuracy, particularly in complex scenarios.

Training Data Contamination. Multiple lines of evidence suggest our results are not due to contamination between LLM pre-training and our test data. State-of-the-art LLMs like Claude-3.5-Sonnet perform poorly (Table 2, Figure2) without task-specific fine-tuning, indicating limited retention of price relationships during pre-training. Additionally, our data scaling experiments (Figure 4) show consistent performance improvements with increased training data across all datasets, indicating that more data contributes to higher learning.

Practical Applications of Text-To-Distribution Modeling. Our price distribution estimation approach from unstructured text input helps in generating substantially more informative outputs than traditional point-wise regression methods. These price distributions can be used for: (i) capturing varying degrees of price uncertainty across items, (ii) providing interpretable probability bounds (e.g., 90% confidence intervals), and (iii) representing diverse distribution shapes as shown in Figure 3.

7 Conclusion

We demonstrated the effectiveness of LLMs with quantile regression heads for probabilistic price prediction from unstructured inputs which not only produce well-calibrated price distributions but also achieve superior point estimates compared to traditional approaches. Our Mistral-7B-Quantile model outperforms traditional approaches and few-shot, in-context learning across multiple datasets, with performance notably improving with larger model sizes and training volumes. Our findings establish a foundation for probabilistic regression with LLMs and showcase their capability in complex numeric prediction tasks using unstructured input.

There are several promising avenues for future research, such as hybrid architectures combining decoder-only models with traditional pricing approaches, explicitly incorporating domain knowledge about pricing and market dynamics, exploring advanced LLM reasoning techniques, and building more interpretable and reliable models that provide insights into they make pricing decisions.

The LLM-based regression approach could also be applied to a number of other existing text-based tasks, such as financial forecasting of return and volatility from news articles and social media, sentiment analysis, and text readability scoring.

8 Limitations

We acknowledge the following limitations of our study. First, we did not fine-tune LLMs larger than 7B parameters in size. Second, although we focused exclusively on the pricing task in this work, we believe that our quantile regression approach would generalize well to other domains, given that our model architecture contains no domain-specific components. However, we did not evaluate this on other general regression domains or non-price prediction tasks. Third, we do agree that our training data is old and it is possible that the LLMs we experimented upon may have seen this data during their pre-training. However, multiple experimental results discussed earlier suggest that this contamination does not significantly contribute to observed results. Finally, some of the listings in our datasets date back to 5-10 years, and we did not explore in detail how this can affect the performance of our in-context learning baselines.

Acknowledgments

The authors would like to thank Yiman Ren and Arman Akbarian for preliminary explorations of this project, and the anonymous reviewers for their helpful feedback.

References

Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, and Ammar Ahmad Awan et al. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *Preprint*, arXiv:2404.14219.

Amazon. 2024. Amazon nova foundation models. https://aws.amazon.com/bedrock/nova. Accessed February 4, 2025.

Anthropic. 2024. Claude 3.5 sonnet. Accessed February 4, 2025.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, and Kai et al. Dang. 2023a. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Yu Bai, Fan Chen, Huan Wang, Caiming Xiong, and Song Mei. 2023b. Transformers as statisticians: Provable in-context learning with in-context algorithm selection. *arXiv preprint arXiv:2306.04637*.

Zsolt Bitvai and Trevor Cohn. 2015. Non-linear text regression with a deep convolutional neural network. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 180–185, Beijing, China. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, and Jared D et al. Kaplan. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Tingting Chen and Shijing Si. 2024. Predicting rental price of lane houses in Shanghai with machine learning methods and large language models. *arXiv preprint arXiv:2405.17505*.

Yunfei Chu, Jin Xu, Qian Yang, Haojie Wei, Xipin Wei, Zhifang Guo, Yichong Leng, Yuanjun Lv, Jinzheng He, Junyang Lin, et al. 2024. Qwen2-audio technical report. *arXiv preprint arXiv:2407.10759*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale.

Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. 2024. A decoder-only foundation model for time-series forecasting. In *Forty-first International Conference on Machine Learning*.

Neşat Dereli and Murat Saraclar. 2019. Convolutional neural networks for financial text regression. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 331–337, Florence, Italy. Association for Computational Linguistics.

Ilia D. Dichev, Xinyi Huang, Donald K.K. Lee, and Jianxin Zhao. 2023. Estimating and using distributional forecasts of earnings. Working paper, SSRN. Available at SSRN, Last revised: May 6, 2025.

Shivam Garg, Dimitris Tsipras, Percy Liang, and Gregory Valiant. 2022. What can transformers learn in-context? a case study of simple function classes. In *Advances in Neural Information Processing Systems*.

Tilmann Gneiting and Adrian E Raftery. 2007. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378.

Nate Gruver, Marc Anton Finzi, Shikai Qiu, and Andrew Gordon Wilson. 2023. Large language models are zero-shot time series forecasters. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Wenjun Gu, Yihao Zhong, Shizun Li, Changsong Wei, Liting Dong, Zhuoyue Wang, and Chao Yan. 2024. Predicting stock prices with FinBERT-LSTM: Integrating news sentiment analysis. In *2024 8th International Conference on Cloud and Big Data Computing*.

Ragıp Gürlek, Francis de Véricourt, and Donald K.K. Lee. 2024. Boosted generalized normal distributions: Integrating machine learning with operations knowledge. Working paper, SSRN. Available at SSRN, Posted: August 1, 2024.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Joseph Marvin Imperial. 2021. BERT embeddings for automatic readability assessment. pages 611–618.

Thomas Kneib, Alexander Silbersdorff, and Benjamin Säfken. 2023. Rage Against the Mean – A Review of Distributional Regression Approaches. *Econometrics and Statistics*, 26:99–123.

Roger W Koenker and Gilbert Bassett. 1978. Regression quantiles. *Econometrica*, 46(1):33–50.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2021. Retrieval-augmented generation for knowledge-intensive nlp tasks. *arXiv preprint arXiv:2005.11401*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Michal Lukasik, Harikrishna Narasimhan, Aditya Krishna Menon, Felix Yu, and Sanjiv Kumar. 2024. Regression aware inference with LLMs. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 13667–13678, Miami, Florida, USA. Association for Computational Linguistics.

Mistral. 2023. Mistral 7b. Https://mistral.ai/news/mistral-7b/.

Dane Morgan and Ryan Jacobs. 2024. Regression with Large Language Models for Materials and Molecular Property Prediction (part 1 of 2).

Jianmo Ni, Jiacheng Li, and Julian McAuley. 2019. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pages 188–197.

Zihang Qiu, Chaojie Li, Zhongyang Wang, Renyou Xie, Borui Zhang, Huadong Mo, Guo Chen, and Zhaoyang Dong. 2024. EF-LLM: Energy forecasting LLM with AI-assisted automation, enhanced sparse prediction, hallucination detection. *arXiv preprint arXiv:2411.00852*.

Xingyou Song, Oscar Li, Chansoo Lee, Bangding Yang, Daiyi Peng, Sagi Perel, and Yutian Chen. 2024. Omnipred: Language models as universal regressors. *CoRR*, abs/2402.14547.

Eric Tang, Bangding Yang, and Xingyou Song. 2024. Understanding LLM embeddings for regression. *CoRR*, abs/2411.14708.

Robert Vacareanu, Victor-Andrei Negru, Vlad Suciu, and Mihai Surdeanu. 2024a. From words to numbers: Your large language model is secretly a capable regressor when given in-context examples. *arXiv preprint arXiv:2404.07544*.

Robert Vacareanu, Vlad-Andrei Negru, Vasile Suciu, and Mihai Surdeanu. 2024b. From words to numbers: Your large language model is secretly A capable regressor when given in-context examples. *CoRR*, abs/2404.07544.

Yaqing Wang, Quanming Yao, James T Kwok, and Lionel M Ni. 2020. Generalizing from a few examples: A survey on few-shot learning. *ACM Computing Surveys*, 53(3):63.

Hanxiang Zhang, Yansong Li, and Paula Branco. 2024. Describe the house and i will tell you the price: House price prediction with textual description data. *Natural Language Engineering*, 30(4):661–695.

Yi Zou, Mengying Shi, Zhongjie Chen, Zhu Deng, Zongxiong Lei, Zihan Zeng, Shiming Yang, Hongxiang Tong, Lei Xiao, and Wenwen Zhou. 2025. ESGReveal: An LLM-based approach for extracting structured data from ESG reports. *Journal of Cleaner Production*, 489:144572.

Appendix

A Dataset Details and Examples

We show in Table 4 examples of the inputs in each of our three datasets. Each data entry contains both structured and unstructured text information. The currency distribution of the Used Boats dataset is shown in Table 5. We also show in Figure 6 the LLM prompt that we used to clean up all three of our datasets and remove the rows containing erroneous (described in Section 3). We show examples of such rows in Table 7.

A.1 Validating LLM-based Data Filtering

To address potential concerns about whether the LLM filtering removed hard examples or created unintended biases, we made two key observations. First, we noted that Claude, the model performing the cleanup, shows poor performance in zero-shot and few-shot settings on the data it marked as clean, providing initial evidence that it did not selectively retain easily predictable cases.

More rigorously, our human evaluation study on a balanced subset comparing both LLM-accepted and LLM-rejected cases confirmed that the filtering criteria were appropriate and unbiased. Specifically, we selected a balanced random subset of data marked as both acceptable and unacceptable by the LLM. Independent human evaluators assessed these samples without knowledge of the LLM's decisions. As shown in Table 6, human evaluators assessed 341 samples from the Amazon dataset and 153 samples from the Cars dataset. The results demonstrate strong agreement between human and LLM judgments, with agreement rates of 95.3% and 94.1% for Amazon and Cars datasets, respectively.

To further validate the filtering effectiveness, we compared model performance on both LLM-filtered and human-validated subsets. For the Amazon dataset, Mistral-7B-Quantile achieved a MAPE of 16.3% (95% CI: [14.3%, 18.3%]) on LLM-filtered data and 43.76% (95% CI: [14.9%, 87.5%]) on human-validated data. For the Cars dataset, the model showed nearly identical performance with MAPE of 5.82% (95% CI: [4.23%, 7.43%]) and 5.79% (95% CI: [4.22%, 7.53%]) on LLM-filtered and human-validated sets, respectively.

A Fisher test with bootstrap sampling comparing the MAPEs between LLM-filtered and human-validated test sets yielded a p-value of 0.198, indicating no statistically significant difference between the two sets' prediction accuracy. This statistical evidence, combined with the high human-LLM agreement rates and Claude's poor zero-shot performance on the filtered data, strongly supports the reliability and unbiased nature of our LLM-based filtering approach.

A.2 Price Distributions

Figure 5 presents the density distributions of prices across the datasets. All distributions exhibit notable right-skewed patterns, though with varying degrees of concentration and scale. The Amazon Products prices show a sharp peak around $25 with a relatively narrow spread, suggesting most reviewed products fall within the affordable consumer goods range. The used car market displays a broader distribution centered approximately around $15,000-$20,000, with a gradual tapering toward higher price points. The used boat market demonstrates the largest price variation, with values extending into the millions of dollars, though the core distribution remains concentrated in the lower price ranges. In this visualization, all distributions are trimmed at the 95th percentile, to trim the outliers.

B Further Modeling Details

B.1 Ensuring Monotonicity and Continuous Quantile Prediction

This section describes two structural additions we can implement in the quantile regression head, previously denoted by $g(\cdot;\phi)$ in section 4.2, to ensure two properties. First, is the monotonicity of the quantile. Specifically, just a quantile regression head and the use of pinball loss provides no guarantee that predicted quantiles $\hat{q}_{\tau_1}, \hat{q}_{\tau_2}, \ldots, \hat{q}_{\tau_K}$ will satisfy the monotonicity constraint $\hat{q}_{\tau_i} \leq \hat{q}_{\tau_j}$ for $\tau_i < \tau_j$. This can lead to nonsensical predictions where, for example, the 90th percentile could be lower than the 80th percentile.

Dataset Type	Example Data Structure
Amazon Products	{<product> <title>Tubing End Cap Solid Brass Scroll End</title> <description>CAP-off your railing in style with our selection of END CAPS and PLUGS...</description> <brand>Renovator's Supply</brand> <type>Pipe Fittings</type> <attributes>Part Number: 95988, Material: Solid Brass</attributes> </product>, 'price': $34.163}
Used Cars	{<used_car> <model_type>pickup, sierra 1500 crew cab slt, gmc, 2014.0</model_type> <description>Carvana is the safer way to buy a car During these uncertain times, Carvana is dedicated to ensuring safety for all of our customers. In addition to our ...[Removed due to length]</description><size></size><color>white</color> <region>auburn, , al</region><condition>good, clean</condition> <features>cylinders: 8 cylinders, fuel: gas, odometer: 57923.0, transmission: other, VIN: 3GTP1VEC4EG551563, drive: , </features> </used_car>, 'price': $33589.548}
Used Boats	{<boat> <boat_type>Flybridge</boat_type> <boat_manufacturer>Galeon power boats</boat_manufacturer> <size>Length: 9.6, Width: 3.0</size> <condition>Used boat, Diesel</condition> <material>GRP</material> <region>Italy » Lombardia - Trentino Alto Adige » MARINA DI VERBELLA - LAGO MAGGIORE</region> <year_built>2005</year_built> <price_currency>EUR</price_currency> </boat>, 'price': €68000}

Table 4: Example data format for different datasets. Each dataset contains both unstructured and structured fields with categorical and numerical valued attributes, capturing various item attributes and price information.

Currency	Count
EUR	8,430
CHF	980
GBP	298
DKK	180

Table 5: Used Boats Currency Distribution

Dataset	Total Samples	Both Agree	LLM Acc., Human Rej.	LLM Rej., Human Acc.
Amazon	341	325	14	2
Used Cars	153	144	9	0

Table 6: Human Validation of LLM-cleaned Prices. 'Acc.' and 'Rej.' stand for Accept and Reject.

Table 7: Examples of erroneous prices across datasets that were removed

Dataset	Product Type	Description Summary	Condition	Price
Amazon Products	RAM Memory	16GB (2x8GB) DDR3 RAM for Toshiba Satellite	New	$3.28
Amazon Products	Window Insulation Kits	500 sqft (4ft x125ft) of NASA TECH Commercial Grade Reflective Insulation	New	$2.85
Used Cars	Mercedes E-Class	2015, 59,749 miles,4MATIC, Blue	Excellent	$1.00
Used Cars	Chevrolet Malibu LS Sedan	2015, 79,539 miles, Blue	Clean	$165
Boats	Rigiflex Motor Yacht	2017, 4m length, 1.9m width,Switzerland	New	3337 CHF
Boats	Whaly Pontoon boat	2018, 4.35m length, 1.73m width,Italy	New	3300 EUR

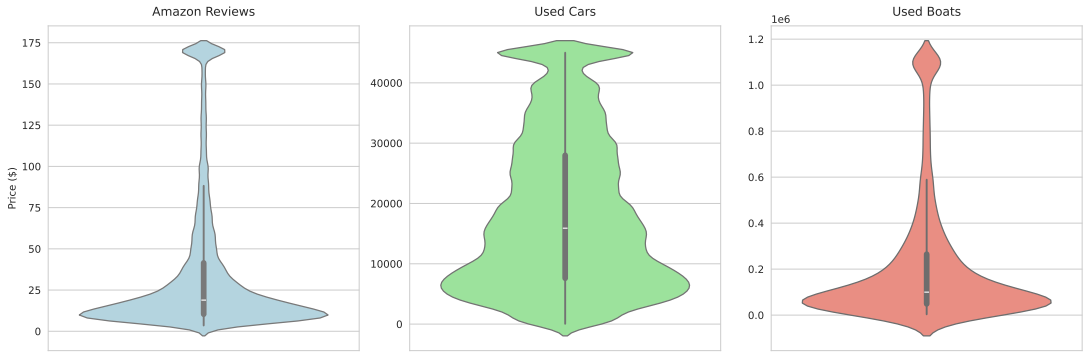

Figure 5: Density distribution of prices across three different datasets: Amazon Products, used cars, and used boats. The distributions are trimmed at the 95th percentile to handle outliers.

The second issue is limited quantile resolution. That is, training on a fixed set of K quantile levels (e.g., $\tau \in \{0.1, 0.2, \ldots, 0.9\}$) restricts predictions to these specific levels, preventing inference at arbitrary quantile levels such as $\tau = 0.73$.

Below, we describe how we can address both challenges through a combination of delta encoding and linear interpolation.

Monotonicity via Delta Encoding: Instead of directly predicting quantile values, as in eq. (4) via a regression head, we can adjust the architecture to predict the first quantile value: $\hat{q}_{\tau_1}$, and the non-negative differences between consecutive quantiles: $\Delta_i = \hat{q}_{\tau_{i+1}} - \hat{q}_{\tau_i} \geq 0$.

This can be implemented as:

$$\mathbf{z}_{\text{deltas}} = [\mathbf{z}_0, \sigma(\mathbf{z}_1), \sigma(\mathbf{z}_2), \ldots, \sigma(\mathbf{z}_{K-1})] \tag{6}$$

$$\hat{\mathbf{q}} = \mathsf{CumSum}(\mathbf{z}_{\text{deltas}}) \,, \tag{7}$$

where $\mathbf{z}$ is $\mathbf{h}_T$, $\sigma(\cdot)$ is a non-negative activation function (e.g., ReLU or SoftPlus), and CumSum denotes the cumulative sum operation. This construction guarantees $\hat{q}_{\tau_1} \leq \hat{q}_{\tau_2} \leq \ldots \leq \hat{q}_{\tau_K}$ by design.

Note that the above modification is purely an architectural modification that guarantees monotonicity by construction, while keeping the loss function and training objective exactly the same as described in section 4. The network still learns to minimize the pinball loss, it just does so through an architecture that makes it impossible to violate monotonicity.

Continuous Quantile Prediction via Interpolation: To predict quantiles at arbitrary levels $\tau \in (0, 1)$ not in our initial quantile levels used during training, one could use linear interpolation between adjacent trained quantiles. For a query quantile τ, one can find the adjacent trained quantile indices: $i = \lfloor \tau \cdot (K-1) \rfloor$ and $i+1$, then compute the interpolation weight: $w = \tau \cdot (K-1) - i$, and interpolate:

$$\hat{q}_\tau = (1 - w) \cdot \hat{q}_{\tau_i} + w \cdot \hat{q}_{\tau_{i+1}} \,.$$

```
You are an expert in understanding product details and product prices. Given the below information
about a product and its corresponding sale price, judge whether the given price is within a
reasonable range for the given product, or if it is too high or too low.

Also generate a short reason. Your final output should be a single dict within <result> tags with
two keys: price_quality and reason.

[PRODUCT INFO]
Sale Price: [PRICE INFO]
```

Figure 6: Sample LLM prompt that we used to clean up the three of our datasets to remove rows with unreasonably high or unreasonably low prices, with respect to the item contexts.)

This leads to continuous quantile predictions across the entire range $(0, 1)$ while maintaining monotonicity, as linear interpolation preserves order relationships.

B.2 Few-shot Learning

Few-shot learning enables models to make predictions with limited training examples, a capability that has proven particularly effective with LLMs (Wang et al., 2020). Recent theoretical work has demonstrated that this ability, also known as in-context learning, has roots to transformer architectures (Garg et al., 2022; Bai et al., 2023b; Vacareanu et al., 2024a).

In our pricing context, few-shot learning allows LLMs to leverage their pre-trained knowledge for price estimation with minimal additional examples. We enhance this approach by selecting prompt examples similar to the target product based on the category or manufacturer of the respective items, similar to retrieval-augmented generation (RAG) techniques (Lewis et al., 2021).

We evaluate the zero-shot and few-shot performance of two state-of-the-art LLMs, Claude-3.5-Sonnet and Nova Pro (Anthropic, 2024; Amazon, 2024). We implement three few shot example selection strategies: (i) random sampling; (ii) category-based stratified sampling and (iii) similar item sampling based on cosine similarity of Qwen2-7B embeddings. The latter two leverage domain similarity for potentially better price estimation. We vary the number of examples as $\{0, 2^0, 2^2, \ldots, 2^{11}\}$, constrained only by the available dataset size and the LLM context window length, to analyze the relationship between example count and performance. All few-shot experiments use consistent prompts, shown in Figure 7, and temperature equal to 0. Aligned with prior literature (Vacareanu et al., 2024b), we utilize these models for point estimates only, as distributional predictions require specialized decoding rules (Lukasik et al., 2024) that are limited to open-source models.

B.3 Using Cross-Entropy Loss

In preliminary experiments, we compared three fine-tuning approaches: regression with squared error loss, regression with quantile (pinball) loss, and token prediction with cross-entropy loss. The regression approaches directly optimize for price predictions, treating the task as a continuous value prediction problem, while the cross-entropy approach treats prices as text and follows the traditional next token prediction.

In our experiments, regression-based approaches significantly outperformed the cross-entropy approach, with squared error loss showing a 1.11 percentage point improvement in MAPE (95% CI: [0.40%, 1.87%]). Based on these findings, we focused on regression and quantile loss fine-tuning for all subsequent experiments.

```
You are an expert in understanding product details and product prices.
Predict the price in US dollars as a float32 number, for the given set of products. Output a JSON
dict with a key for each input product ID, and a nested dict with a key 'price' containing your
predicted price of the product, and another key 'reason' briefly explaining why your predicted
price is correct.
Put the output JSON dict in <result> tags.

Here are some examples of products and their prices.

[EXAMPLES]

Now predict the price for:
[CONTEXT]
```

Figure 7: Sample prompt for zero shot and few shot LLM based price prediction. This prompt is customized for the Amazon Products dataset, but we used very similar prompts for the other two datasets as well, with minor modifications (e.g., changing references to 'products' to 'used cars' etc.)

C Metric Definitions and Implementation Details

We clarify that all publicly available datasets as well as models that we used in this work were used in accordance with their license and terms for use. We did not use any data or model outside of its intended purpose.

Quantile Levels and Point Prediction: For all models that predict distributions, we take $K = 200$ and $\boldsymbol{\tau}$ is obtained by dividing the interval $(0, 1)$ into K equal-length sub-intervals. We studied the impact of varying the number of quantiles $K = 10, 50, 200, 500, 1000$ across the three datasets and found that initially as K increases the performance improved, but plateaued after a certain point, which in our case was $K = 200$. We therefore used this number for all our experiments involving a trained model with a quantile regression head. We use models that produce a distribution both for generating probabilistic outputs and for point predictions. In the latter case, we take the predicted quantile at $\tau = 0.5$ as the point estimate. Additionally, we include baseline models trained solely with traditional squared error loss, using their direct predictions for comparison.

We also tune the value of the smoothing parameter α, that controls how closely the SoftPlus function approximates the ReLU function. We experimented with values ranging from 10^{-5} to 10^{-1} and did not observe significant effects. We therefore settled on 10^{-2}, to achieve a balance between being closer to a true quantile loss and also achieving numerical gradient stability.

C.1 Baselines

Text Embedding Baselines: We evaluate traditional ML models with text embeddings. Text features (title, description, attributes) are concatenated with appropriate field markers and converted to embeddings using the general Qwen2-7B-instruct embedding model (Chu et al., 2024). These embeddings serve as input features for five models: Ridge Regression and XGBoost for point estimation, Quantile Regression (with two hidden layers) for distribution prediction, trained on log-transformed target[5], and two nearest neighbor-based distribution prediction approaches. The first nearest neighbor model predicts distributions by using the empirical distribution of target values from selected neighbors in the training set, while the second variant employs a radius-based selection criterion with a minimum neighbor requirement. All hyperparameters are selected using 5-fold cross-validation.

Fine-tuned Decoder LMs with Quantile Head: We fine-tune Mistral-7B (7 billion parameters), (Mistral, 2023), Phi-3B (3B parameters), (Abdin et al., 2024), and Qwen-500M (500M parameters), (Bai et al., 2023a), using LoRA (Hu et al., 2022) (rank=192, alpha=384, dropout=0.1) with the AdamW optimizer (Loshchilov and Hutter, 2019) (learning rate=$1.0\mathrm{e}{-06}$, weight decay=0.01), with the quantile head described in Section 4, on log-transformed targets.

[5]In all three data sets since the target was price, we used its log-transformed prices to handle the wide range of values in our datasets during training.

Fine-tuned Encoder LMs with Quantile Head: We fine-tune XLM-RoBERTa (Conneau et al., 2019) in both base (279M parameters) and large (561M parameters) variants, adding a regression head as described in Section 4.

Fine-tuned LLM with Regression Head: We fine-tune Mistral-7B, the largest LLM in our set, with a regression head to study the impact of quantile prediction versus point estimation.

Few-shot SOTA LLMs: We evaluate the zero-shot and few-shot performance of two state-of-the-art LLMs, Claude-3.5-Sonnet and Nova Pro (Anthropic, 2024; Amazon, 2024). For few-shot learning, we implement three example selection strategies: (i) random sampling; (ii) category-based stratified sampling and (iii) similar item sampling based on cosine similarity of Qwen2-7B embeddings. The latter two leverage domain similarity for potentially better price estimation. We vary the number of examples as $\{0, 2^0, 2^2, \ldots, 2^{11}\}$, constrained only by the available dataset size and the LLM context window length, to analyze the relationship between example count and performance. All few-shot experiments use consistent prompts (Figure 7 of Appendix) and temperature equal to 0. Aligned with prior literature (Vacareanu et al., 2024b), we utilize these models for point estimates only, as distributional predictions require specialized decoding rules (Lukasik et al., 2024) that are limited to open-source models.

For the Amazon Products and Boats dataset, even with the best-performing category-based sampling strategy and optimal shot count (256), both Claude and Nova-pro achieve MAPEs more than 35%, lagging significantly behind fine-tuned Mistral-7B's MAPE of 16.86% and 21% respectively. The performance disparity is similarly stark in the Used Cars dataset. For the Used Cars dataset, while Mistral-7B achieves a MAPE of 6.3%, few-shot approaches struggle with much higher error rates: both Claude and Nova-pro show MAPEs between 230-245% with random sampling and 290-305% with category-based sampling. Nova-pro performs similarly poorly, with error rates consistently above 220%. For the Boats dataset, the gap narrows somewhat but remains substantial. Mistral-7B's MAPE of 21.2% still outperforms the best few-shot results (Claude with random sampling at 35% MAPE) by a considerable margin. Choosing few shot examples similar to the target item based on pairwise cosine similarity using Qwen-7B-embeddings also gives a MAPE within 2-3% of the random sampling strategy.

Our experiments also reveal an intriguing pattern in few-shot learning performance. Contrary to common intuition, our experiments also reveal that increasing the number of examples beyond a certain point starts degrading model performance. This finding challenges the conventional wisdom that more examples invariably lead to better few-shot performance. The degradation might be attributed to several factors, such as models' context window size limitations, potential interference between examples, or increased complexity in extracting relevant patterns from larger sets of examples. This non-monotonic behavior suggests that careful attention must be paid to the number and quality of examples used in price prediction tasks, and there exists an optimal window for few-shot learning, beyond which additional examples may interfere with the model's ability to effectively leverage the in-context information. This observation has important implications for the practical application of few-shot learning in pricing tasks, suggesting that careful attention should be paid to the number of examples used rather than simply maximizing them.

C.2 Evaluation Metrics

We use two sets of metrics, one evaluating the estimated distributions generated by our quantile regression models and the other for point estimates. For each metric we report 95% confidence intervals with bootstrap resampling (1000 iterations): $\text{CI}_{95\%}(M) = [\hat{M}_{(0.025)}, \hat{M}_{(0.975)}]$ where $\hat{M}_{(q)}$ denotes the q-th quantile of the bootstrap distribution of metric M.

C.2.1 Distribution Quality Metrics

Assuming we have a test set of size n: $(\mathbf{x}_i, y_i)_{i=1}^n$ and for each test point $\mathbf{x}_i$, we have predicted quantiles, $\hat{\mathbf{q}}_{\boldsymbol{\tau}}(\mathbf{x}_i) = (\hat{q}_{\tau_1}(\mathbf{x}_i) \leq \cdots \leq \hat{q}_{\tau_K}(\mathbf{x}_i))$.

Calibration Error (CE): CE measures how well predicted quantiles match their theoretical coverage: $\text{CE} = (1/K) \sum_{k=1}^{K} |\widehat{\text{coverage}}(\tau_k) - \tau_k|$. where $\widehat{\text{coverage}}(\tau_k)$ is the empirical fraction of true values in the test set, below the τ_k quantile.

Continuous Ranked Probability Skill Score (CRPSS): This metrics is a scale-free version of the well-known CRPS which measures the integrated squared difference between predicted and true cumulative distribution functions:

$$\text{CRPS} = \frac{1}{n}\sum_{i=1}^{n}\int_{-\infty}^{\infty}\left(\hat{F}_{\mathbf{x}_i}(r) - \mathbf{1}_{y_i \le r}\right)^2 dr\,,$$

where $\hat{F}_{\mathbf{x}_i}$ is the estimated CDF using $\hat{\mathbf{q}}_{\boldsymbol{\tau}}(\mathbf{x}_i)$. As a proper scoring rule, CRPS converges to zero if and only if the predicted distribution matches the true distribution (Gneiting and Raftery, 2007). We report the (scale-free) skill score

$$\text{CRPSS} = 1 - \left(\frac{\text{CRPS}_{\text{model}}}{\text{CRPS}_{\text{reference}}}\right).$$

where the reference is the empirical distribution of training targets.

Relative Confidence Interval Width (RCIW). RCIW measures the average width of predicted intervals relative to the true value:

$$\text{RCIW}_\gamma = \frac{100}{n}\sum_{i=1}^{n}\frac{U_i^\gamma - L_i^\gamma}{|y_i|}$$

where $[L_i^\gamma, U_i^\gamma]$ is the predicted $(1-\gamma)$ CI for $\mathbf{x}_i$. RCIW captures the sharpness of the distribution, where smaller values indicate a tighter interval.

C.2.2 Point Estimate Metrics

We report: MAPE (Mean Absolute Percentage Error)

$$(100/n)\sum_{i=1}^{n}|(y_i - \hat{y}_i)/y_i|\,,$$

WAPE (Weighted Absolute Percentage Error),

$$\frac{100\sum_{i=1}^{n}|y_i - \hat{y}_i|}{\sum_{i=1}^{n}|y_i|}\,,$$

and MPE (Mean Percentage Error):

$$(100/n)\sum_{i=1}^{n}(y_i - \hat{y}_i)/y_i\,.$$

C.3 Computation Infrastructure

We used the AWS EC2 infrastructure for running all our experiments. We estimate the use of about 2000 GPU hours for all our model training and evaluations. We also used an AI assistant to help with some parts of code writing.

D Detailed Analysis of Model Performance

In this section, we provide an examination of our Mistral-7B-Quantile model's performance across different product categories and analyze the distributional patterns captured by the model. First, we present a breakdown of prediction accuracy by category, revealing which product types are most (least) challenging for price prediction. We then explore how the model captures various distributional shapes that reflect the underlying market dynamics of different products.

D.1 Performance Breakdown by Category

We provide detailed performance breakdown of our best model on the different categories of each dataset in Tables 8 and 9.

Category	MAPE [%]	Size	Price Range [$] [Min, Median, Max]
Camera Lenses	34.75 [20.57, 44.61]	6	[7.78, 36.60, 294.95]
Tools	33.92 [23.22, 44.19]	6	[7.99, 15.93, 84.95]
Bakeware Sets	33.29 [25.07, 40.03]	8	[3.99, 8.09, 130.48]
Compressors	33.09 [25.68, 39.96]	13	[29.99, 165.00, 395.65]
All-Purpose Labels	30.19 [15.89, 43.41]	7	[4.99, 14.95, 33.29]
Platters	29.86 [20.35, 38.93]	6	[14.99, 26.46, 69.99]
Pickups & Pickup Covers	29.86 [22.64, 36.16]	9	[6.04, 12.40, 219.00]
Lighting Assemblies	29.66 [15.11, 44.26]	6	[14.98, 35.67, 173.43]
Hard Hat Accessories	29.40 [17.66, 42.69]	6	[3.99, 4.99, 11.09]
Internal Hard Drives	29.17 [19.69, 38.61]	12	[14.99, 52.50, 599.99]

Table 8: Examples of Categories with High Mistral-7B MAPE for Amazon Products dataset (Minimum Size > 5)

Category	MAPE [%]	Size	Price Range [$] [Min, Median, Max]
Window Tinting Kits	4.68 [2.74, 6.75]	19	[24.49, 39.49, 283.94]
Keyrings & Keychains	5.39 [2.59, 8.12]	6	[5.99, 8.09, 10.19]
CV Boots & Joints	5.69 [3.38, 8.31]	11	[11.50, 11.88, 69.99]
Exhaust	5.80 [2.69, 9.22]	8	[15.72, 122.78, 719.48]
Machine Screws	6.01 [2.65, 9.60]	8	[7.35, 9.96, 13.84]
Socket Wrenches	6.30 [2.32, 11.21]	6	[8.51, 14.65, 108.38]
Engine Management Systems	6.65 [3.72, 10.82]	19	[15.22, 69.95, 69.95]
Inkjet Printer Paper	6.79 [3.66, 9.69]	6	[9.50, 29.42, 152.26]
License Plate Frames	6.94 [2.62, 13.28]	11	[5.66, 16.99, 29.99]
Highball Glasses	6.94 [3.08, 11.00]	6	[34.46, 47.27, 110.36]
Touchup Paint	7.37 [5.76, 9.05]	68	[8.25, 15.30, 71.92]
Keychains	9.70 [7.79, 11.83]	71	[5.79, 10.99, 55.99]
Frames	9.91 [8.64, 11.18]	231	[4.99, 14.99, 95.00]
Custom Fit	10.43 [9.19, 11.68]	175	[18.99, 119.00, 599.00]
Body	12.11 [10.63, 13.60]	136	[6.99, 43.49, 409.85]

Table 9: Examples of Categories with Low Mistral-7B MAPE for Amazon Products dataset (Minimum Size > 5)

D.2 Distributional Patterns in Price Predictions

The probability distributions shown in Figure 3 and Figure 8 show different patterns that reflect the underlying market dynamics of different product categories.

Unimodal Distributions: Products such as the Merritt tumbler, Toyota Corolla, Ford Mustang, and wedding guest book exhibit single-peaked distributions. These standardized products typically have well-established market prices with relatively low variance. The narrow, symmetric distributions suggest predictable pricing driven by clear market segments and standardized features, which is reflected in the model's lower prediction errors (MAPE ranging from 1.2% to 6.5%) for these items.

Bimodal Distributions: Several products display dual peaks, including the Holley EFI gauges, AC compressor, and to varying degrees, the Lamborghini Huracán and Toyota Tacoma. This bimodality likely reflects distinct market segments. For automotive parts (gauges, compressor), the two peaks may represent new versus refurbished/used markets operating at different price points. For vehicles, different trim levels, model years, or condition categories (e.g., certified pre-owned versus standard used) create separate pricing clusters. Finally, the Toyota Tacoma's bimodal pattern potentially captures the price gap between base work trucks and fully-loaded consumer models.

Right-Skewed Distributions: The luxury marine vessels (Sunseeker yacht, Storebro, and Baikai flybridge) exhibit right skew with heavier tails. This pattern aligns with the characteristics of high-end markets where, base models establish the primary peak, extensive customization options, rare features, or pristine/collector conditions create the long tail. Additionally, the extreme tail (particularly visible in the Baikai boat with prices reaching $500K+) likely represents highly customized or rare configurations.

The correlation between distribution shape and prediction accuracy is noteworthy. Standardized products with unimodal distributions achieve lower prediction errors, while luxury items with complex,

skewed distributions show higher uncertainty (MAPE up to 37%), that could be due to the inherent difficulty in pricing highly variable, customized products.

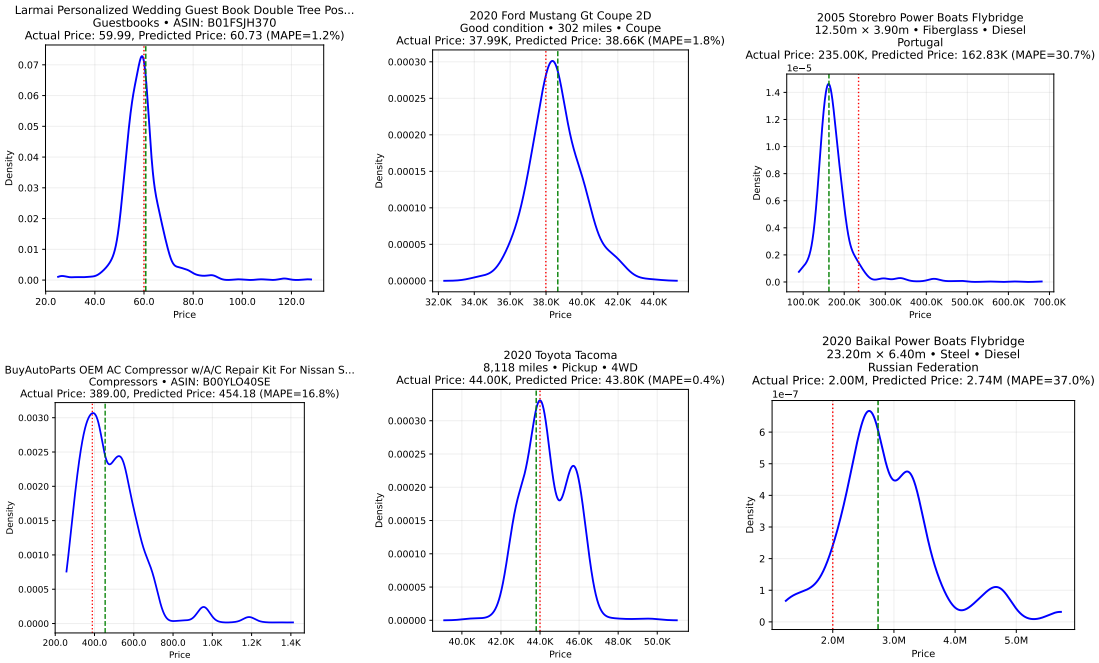

Figure 8: Probability density distribution of the prices predicted by the Mistral-7B-Quantile model across different datasets (blue curve). Each x-axis has a different scale. The red dotted line represents the ground truth price while the green dashed line is the predicted median price. As demonstrated, the model captures different distribution shapes including unimodal (top row), bimodal (bottom row), and right-skewed (right) distributions.