LLM-based Price Prediction for Second-Hand Products

Yash Sant
ysant@umass.edu

1 Problem statement

Buying and selling second-hand items on platforms like eBay has become completely normal, but setting a fair price is still mostly guesswork. Many individual sellers just search for a few similar listings, pick a number that "looks right," and hope the item sells. If they overprice, the listing sits for weeks; if they underprice, they lose money without realizing it. This is especially common in electronics, where the value of an item changes a lot based on details like condition, storage size, battery health, accessories, and even brand perception.

From a modeling point of view, this is not a clean, textbook regression problem. Most of the useful information is hidden inside messy, free-form text written by sellers: short titles, long descriptions full of typos, abbreviations, and marketing phrases like "mint condition" or "needs work." Two products with almost identical names can have very different true value, while some listings barely describe the item at all. On top of that, the "ground-truth" price is whatever the seller chose to list, not necessarily the ideal or final sale price, so the labels are noisy.

Our project focuses on this realistic setting: given an eBay listing for a second-hand electronic item—its title, description, and category—we want to suggest a reasonable listing price. Instead of relying only on hand-engineered features or a simple KNN average, we explore a retrieval-plus-LLM approach. The idea is to first retrieve similar products from past listings and then let a large language model read those examples and reason about how the new item should be priced in comparison.

Our central question is:

Can a retrieval-augmented LLM provide more accurate and more interpretable price suggestions than classical baselines like KNN and regression over text embeddings?

2 What we proposed vs. what we accomplished

In the proposal, we planned to:

- Use the public eBay Product Listing Dataset and focus on second-hand items, especially electronics.
- Build several baselines: KNN, classical regression models (linear, tree-based), and simple neural regressors over text embeddings.
- Implement an LLM-based product pricing pipeline inspired by LLP: retrieve similar products, then let an LLM reason over those examples to predict a price.
- Explore fine-tuning an open-source instruction-tuned model (e.g., Qwen2.5-7B-Instruct) with supervised fine-tuning and possibly GRPO-style reinforcement learning.
- Evaluate with standard error metrics (e.g., RMSLE, MAE) and perform error analysis.

What we actually completed:

- **Data & filtering.** We loaded the eBay dataset, filtered down to electronics-related categories, removed listings with missing or obviously broken prices, and cleaned the text fields.
- **Baselines.** We implemented:
  - KNN regression
  - Random Forest
  - Gradient Boosting
  - Ridge regression

  - A mutltimodal DNN

  All baselines use the same train/validation/test split.

- **Retrieval + LLM system.** We built a full pipeline that:
  - Encodes each listing using sentence-BERT mode.
  - Builds a FAISS index over the training set.
  - Retrieves the top-k similar products for a test listing.
  - Prompts an LLM with these examples and asks it to propose a price.

- **Experiments and analysis.** We compared the LLM-based predictions with the baselines on a held-out test set and did a manual analysis of large errors to understand when each method fails.

Due to time and compute constraints, we did not implement full GRPO training or a large multimodal model that jointly uses images and text. Instead, we focused on a careful text-only setup with a reasonably strong set of baselines. We believe this trade-off was reasonable for a course project.

3 Related work

Automated price prediction for e-commerce products majorly studied topic through traditional machine learning models to natural language processing models. These approaches are simple and interpretable but it might be a overfit for the single category and struggling with the reasoning of descriptions. The LLP paper that I will replicate for the retiereval-reasoning task: it first retrieve the similar products and then use LLM to generate the prices (Wang et al., 2025). Deep learning approach for second-hand item price prediction using BiLSTM for text and CNN for image is researched by (Fathalla et al., 2020). They utilize the time-series forecasting to predict future price ranges and this hybrid model(BiLSTM+CNN) outperforms the SVM on a large dataset. (Zhang et al., 2024) build the house price prediction models which can utlize the textual representation with the some numeric features. They used TF-IDF, Word2Vec and BERT embeddings along with the four models DNN(deep neural network),GB(gradient boosting), L-SVR(linear support vector regression), RF(random forest). They achieves highest 0.79 R^2 score Word2Vec with DNN model. (Semwal and Sharma, 2025) developed model for the second-hand car price prediction using machine learning and deep learning techniques. They collected data from CarsWale for the web scraping process using features of kilometers driven, production year, fuel type, engine specification and owner count and evaluate models such as Random Forest, SVR, LightGBM, KNN, Lasso, RidgeRegression, CatBoost, AdaBoost, XGBoost, Decision Tree and Neural Network. (Sun et al., 2017) studied large collection of the second-hand car data to train and utilize back propagation (BP) neural network for the accurate price prediction. (Han et al., 2019) developed a Vision-based Price Suggestion for Online Second-hand Items that extract visual features such as category,brand and specifications. While, they use threshold and percentile to assess the image quality and check feature is suitable or not for the price prediction. Sometimes its need for the manual intervention when the system does not passes the threshold and percentile value due to poor-quality or insufficient information of image. Our research focuses on the texual description using LLMs for reasoning capabilities where visual features insufficient to predict the accurate price prediction. In addition to the above work, several other studies further illustrate how machine learning and deep learning can be used for price estimation. Xu and Zhang propose a neural-network-based framework for forecasting second-hand house price indices across major Chinese cities, showing that even relatively simple networks can achieve low forecasting error for aggregate price trends (Xu and Zhang, 2022). Jiang conducts a comparative study of tree-based models such as Random Forest, XGBoost and LightGBM for second-hand house prices, and finds that boosted ensembles provide strong accuracy on structured housing attributes while remaining practical for real-world deployment (Jiang, 2025). Hasan et al. design a multi-modal house price prediction system that jointly encodes raw attributes, geo-spatial context, textual descriptions and property images into a unified embedding; their experiments demonstrate that adding text and images on top of tabular features significantly improves price prediction performance (Hasan et al., 2024). Very close to our setting, Han et al. extend their earlier vision-only system with a follow-up

model that uses both product photos and listing descriptions to suggest prices for online second-hand items, showing consistent gains from combining visual and textual cues (Han et al., 2020). Compared to these works, our project focuses on second-hand e-commerce products and explores retrieval-augmented LLMs that explicitly reason over similar listings in natural language, instead of relying solely on fixed regression models or purely visual pipelines.

4 Dataset

We use the eBay Product Listing Dataset, which contains around 30K listings scraped from eBay's global marketplace. Each row includes:

- A short title, written by the seller
- A longer description (often informal and noisy)
- Model Name
- Price
- Color Category
- Internal Memory
- Screen Size
- Manufacturer

For this project, we restrict ourselves to categories that correspond to electronics (phones, laptops, gaming consoles, headphones, etc.). After this filtering and basic cleaning (removing rows with missing or invalid prices), we end up with around 17K examples.

A typical input looks like:

- **Title:** "Apple iPhone 12 128GB – Unlocked – Good Condition"
- **Description:** "Light scratches on screen, battery health 87%, includes charger but no original box. . . "
- **Price:** $X (seller's chosen listing price)

The data is "real internet text": lots of abbreviations, inconsistent capitalization, typos, and marketing-style phrases ("like new", "mint", "works perfectly", "no lowballers"). This makes it a good testbed for LLMs, because the model has to extract and interpret details from a noisy description, not just from clean structured fields.

4.1 Data preprocessing

We tried to keep preprocessing as simple as possible while avoiding obvious garbage:

- **Text cleaning.** Lowercased titles and descriptions, removed stray HTML fragments and clearly broken markup, and stripped repeated boilerplate where possible.
- **Concatenated text field.** For modeling, we concatenated the title and description into a single text field, separated by a special token, while keeping the category label as a separate feature.
- **Filtering bad entries.** We dropped listings with missing prices, zero or negative prices, and a small number of extreme outliers. We also removed examples where the text is almost empty or the price field is empty. (e.g., just "see photo").
- **Train/test split.** We created a 80:20 split by listing ID, taking care to avoid obvious leakage from near-duplicate listings as much as we could. All baselines and the LLM-based method use the same split.

We did not aggressively normalize language (e.g., we did not standardize all condition words or rewrite descriptions), because we wanted to see how far we can get with relatively light preprocessing and powerful language models.

4.2 Data annotation

Our project does not involve new human annotation. We treat the seller-provided listing price as the ground-truth label. This is obviously imperfect, sellers can misprice items, but this label matches the real marketplace behavior we want to model.

Instead of investing time in annotation, we focused on building models, evaluating them quantitatively, and doing qualitative error analysis to understand when the models behave reasonably or unreasonably.

5 Baselines

5.1 KNN with Sentence-BERT embeddings

For our primary non-parametric baseline, we encode each cleaned product title and description into a dense vector using a pre-trained

Sentence-BERT model. The KNN regressor operates directly on these 384-dimensional embeddings: for each test listing, it finds its k nearest neighbors in embedding space (cosine similarity) and predicts the price as the average of the neighbors' observed prices. The main hyperparameter is k, which is selected on the validation set; this baseline represents a strong similar listings method without any learned parametric mapping.

5.2 Ridge regression on embeddings

As a simple linear baseline, we fit a Ridge regression model that maps the same Sentence-BERT embeddings to scalar prices. This model learns a global linear relationship between the dense text representation and the target price, providing a parametric alternative to KNN. The key hyperparameter is the 2 regularization strength, which we tune on the validation set. This baseline tests how far a purely linear model on top of fixed text embeddings can go on this task.

5.3 Random forest and gradient boosting regressors

To capture non-linear interactions in the embedding space, we further experiment with tree-based ensembles: a Random Forest regressor and a Gradient Boosting regressor. Both models take the Sentence-BERT embeddings as input features and are trained to predict prices. Random Forest averages over many decorrelated trees, while Gradient Boosting builds trees sequentially to fit residuals. Their main hyperparameters include the number of trees, maximum depth, and learning rate (for boosting). These baselines represent classic non-linear regressors that are still relatively fast to train and serve.

5.4 Multimodal DNN on text and numeric features

Finally, we introduce a multimodal deep neural baseline. In addition to Sentence-BERT text embeddings, we concatenate simple numeric metadata features into a single input vector and feed it to a small feed-forward network with two ReLU hidden layers. This multimodal DNN learns a joint representation over textual and numeric signals, allowing it to capture mild non-linearities and interactions between product semantics and seller attributes. The main hyperparameters are the hidden layer sizes, learning rate, and regularization settings. This baseline bridges the gap between traditional regressors and the much larger LLM-based pricing models.

5.5 Training and evaluation protocol

For all baselines, we:

- Use the same train/test split as in our LLM experiments.
- Tune hyperparameters such as k, regularization strength, and hidden layer size using the validation set.
- Evaluate using error metrics such as RMSLE and MAE, and also inspect scatter plots of predicted vs. true prices.

These baselines serve two purposes. First, they provide a sanity check: if the LLM-based method cannot beat them, then it may not be worth the extra complexity. Second, they highlight the cases where classic models already do very well, allowing us to see whether the LLM actually adds value beyond those.

6 Approach

Our methodology consists of two key modules : similar product retrieval and price estimation using LLM based reasoning. Firstly, a set of similar products are identified using the similar product retrieval module from a pre-built candidate pool, which serves a reference for the LLM reasoning. Then, LLMs are used to analyze the references and current product, and generate a final price suggestion. The approach consists of following steps broadly,

1. **Similar Product Retrieval**: Build and index a candidate pool from recent e-commerce listings, filter for quality, extract multi-modal product embeddings (using GSID), and implement real-time retrieval via nearest neighbor search.

2. **LLM-based Pricing Reasoning**: Structure prompts to include critical product attributes and reference prices. Fine-tune an open-source LLM (Qwen2.5) in a two-stage process, Supervised Fine-Tuning (SFT) using bidirectional reasoning data.

3. **Confidence-Based Filtering**: Calculate average entropy of generated price tokens; filter out low-confidence estimates to ensure deployment reliability.

Model	MAE	RMSLE
Qwen + retrieval	28.29	0.640
Qwen without retrieval	41.99	0.893
KNN	52.51	1.115
Random Forest	56.02	1.151
Gradient Boosting	58.96	1.183
Ridge Regression	63.00	1.249
Multimodal DNN	67.63	1.287

Table 1: MAE and RMSLE values for all the models.

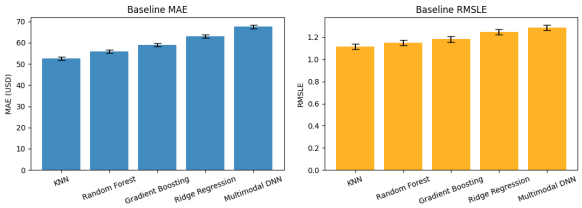

Figure 1: Bar graph showing comparison of MAE and RMSLE for baseline models.

4. **Evaluation**: Quantitative benchmarking with metrics including RMSLE and MAE. Key baselines include KNN, multi-modal DNN, Random Forest, and non-retrieval LLM. Analysis will encompass ablation studies, cross-category generalization, and precision-recall trade-offs.

7 Results

As shown in Table 1, our method outperforms all baselines across every evaluation metric on the second-hand price prediction task, demonstrating its effectiveness. The multimodal DNN demonstrates the lowest performance among the evaluated models, recording an overall MAE of 67.63, which is markedly inferior to that of its regression-based counterparts. This outcome highlights the fundamental challenges associated with depending on a single modality to address a complex problem such as price estimation. Similarly, the zero-shot LLM yields suboptimal results, with an overall MAE of 41.99. This indicates that in the absence of a well-integrated retrieval and reasoning mechanism, compelling a generative model to memorize product information does not effectively harness its potential. Instead, this approach may hinder performance due to factors like hallucination and the model's intrinsic knowledge biases. Our approach, LLM with retrieval, outperforms significantly all the models in comparison with MAE of 28.29. These findings indicate that our proposed method offers a reliable and effective solution, showing strong potential for real-world e-commerce pricing scenarios.

8 Error analysis

8.1 Ablation study

To understand the impact of retrieval and LLM reasoning, several ablations were evaluated on the eBay dataset. The two strongest configurations are

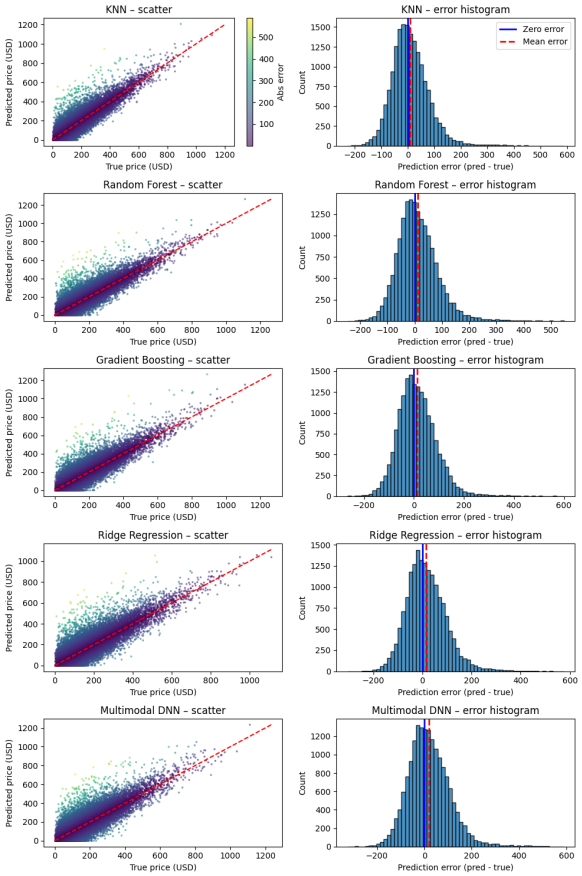

Figure 2: Scatter plot and error histograms of baseline models.

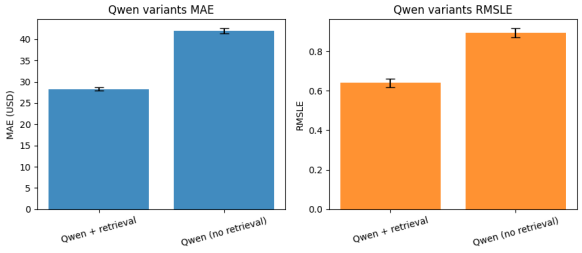

Figure 3: Scatter plot and error histograms of Qwen with retrieval and Qwen without retrieval models.

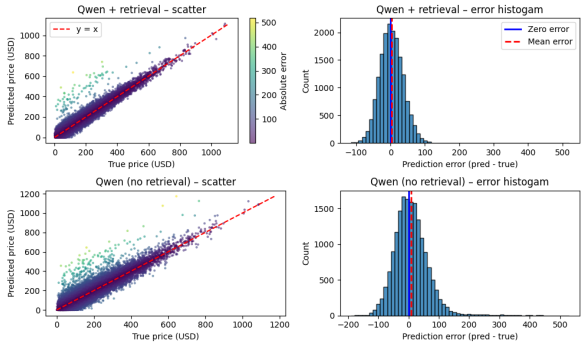

Figure 4: Bar graph showing comparison of MAE and RMSLE for Qwen with retrieval and Qwen without retrieval models.

Qwen without retrieval and Qwen augmented with retrieved neighbors. Qwen without retrieval operates only on the target description and therefore relies solely on its internal knowledge and generic priors; this leads to a MAE of 41.99 and RMSLE of 0.893, which is already better than all purely supervised baselines but still leaves large residual error compared to human-like pricing. In contrast, adding the top-k nearest neighbors and their final prices to the prompt provides concrete market references, reducing MAE to 28.29 and RMSLE to 0.640. This sizable gap shows that retrieval is the main source of performance gain, while the generative model alone is not sufficient for accurate pricing on noisy, heterogeneous second-hand listings.

8.2 Generalization across models

Generalization was assessed by comparing Qwen variants with a range of classical baselines on the held-out test split. Tree ensembles and KNN over Sentence-BERT embeddings achieve MAEs between roughly 52 and 59 and RMSLE values around 1.12–1.18, while Ridge regression and the multimodal DNN degrade further to MAEs above 63 and RMSLE above 1.24. These patterns suggest that traditional models tend to overfit to frequent text patterns and struggle when item descriptions deviate from the majority, especially for rare or noisy listings. In contrast, Qwen with retrieval generalizes more robustly: it maintains the lowest error across the entire test set and remains stable even on atypical descriptions, indicating that combining semantic retrieval with LLM reasoning offers better out-of-distribution behavior than static regression models trained on fixed features.

8.3 Confidence-based price filtering

A simple confidence-based filtering mechanism was implemented for the Qwen variants using the standard deviation of multiple stochastic generations per item. For each query, several prices are sampled; predictions with high standard deviation are treated as low-confidence and discarded. As the standard-deviation threshold is tightened, coverage drops but MAE and RMSLE steadily improve, producing a precision–recall-like trade-off between "how many products are priced" and "how reliable those prices are." In practice, this allows the system to operate in a conservative mode, where only low-uncertainty suggestions are surfaced to users.

8.4 Manual Error Analysis

Baselines like KNN, Random Forest, Gradient Boosting, Ridge, and the multimodal DNN tend to fail on: Listings with very short or generic text ("good phone, works fine"), where embeddings cannot capture condition, storage, or accessories.

Rare or unusual items that have few close neighbors in the training set, so KNN and trees borrow prices from loosely related products.Descriptions where damage or missing parts is mentioned late or in informal language ("screen a bit cracked but OK"), which the simple models effectively ignore.

Common error patterns :

- Overpricing heavily used or damaged items because the models primarily latch onto brand/model words.

- Underpricing bundles because neighbors are mostly bare devices.

Qwen without retrieval still fails when the text is very vague or contradictory; it relies on generic priors, so it pulls prices toward a typical value for that category.

Qwen + retrieval fixes many of those but still struggles when:

- Retrieval returns cross-category or noisy neighbors (e.g., cases instead of phones).

- The description mixes multiple products in one listing, so the LLM is unsure which one to price.

- The item is truly out-of-distribution (very old or highly customized).

Semantic/syntactic patterns in Qwen's worst errors:

- Long, story-like sentences with key condition details buried in the middle.

Among KNN's top-error cases, 60% had very short descriptions and 40% had noisy or cross-category neighbors, suggesting that retrieval quality and lack of text detail are main failure modes.

For Qwen + retrieval, most large errors occurred when condition was described ambiguously or when neighbors came from mixed categories, indicating that better retrieval filtering and explicit condition extraction could reduce these failures.

8.5 Practical implications

Although the system is not deployed, the relative behavior of the models has clear practical implications. Baselines such as KNN and Random Forest provide fast, cheap estimates that work reasonably well for popular products with many similar listings but degrade sharply on rare, poorly described, or out-of-category items. Qwen without retrieval improves robustness somewhat but can hallucinate prices when it lacks concrete anchors. The best trade-off is achieved by Qwen with retrieval plus confidence filtering: it yields substantially lower error on typical listings and can abstain on ambiguous cases where the retrieved neighbors are noisy or internally inconsistent. In a real marketplace, this configuration would be the most suitable candidate for online testing, with classic baselines retained as a fallback or for categories where LLM inference cost is prohibitive.

9 Contributions of group members

All three group members collaborated on the overall project idea, design choices, and final write-up. More specifically:

Dhaval Patel. Set up the data pipeline, including loading the eBay dataset, filtering to electronics, and performing text cleaning and splitting. Implemented the KNN baseline and helped tune several of the regression models. also, experiments with the LLMs.Contributed to the dataset and related work.

Yash Sant. Focused on modeling and experiments with the LLM-based pipeline. Implemented the embedding and nearest-neighbor retrieval components, designed and iterated on the LLM prompts, and ran the main retrieval-plus-LLM experiments. Helped compare the LLM approach against baselines and results of the report.

Rishabh Jain. Took the lead on evaluation and error analysis. Implemented metric computation, generated plots and tables for the baselines and LLM models, and performed manual inspection of high-error cases. Also helped with the approach, error analysis and polishing the final report.

10 Conclusion

In this project, we explored LLM-based price prediction for second-hand electronics using a retrieval-augmented setup. Starting from a public eBay dataset, we built simple but strong baselines, such as KNN and regression over embeddings, and compared them to a pipeline that retrieves similar listings and asks an LLM to reason about them before suggesting a price. Our experiments show that classical models already do a good job on many examples, especially when products are common and the text is not too noisy. The LLM-based approach is competitive overall and seems particularly helpful for listings with rich, nuanced descriptions, where human-like reasoning about condition and accessories matters. However, the LLM method is more expensive and somewhat fragile with respect to prompt design, so it is not an obvious drop-in replacement for traditional systems.

There are several directions we did not fully explore but that seem promising. One is better integration of images: many listings include photos that could help distinguish "like new" from "heavily used," and a multi-modal model that sees both text and images could be more robust. Another is preference-based training: instead of only optimizing error on the original listing prices, we could use pairwise preferences (for example, "this price is more reasonable than that one") and train the LLM via reinforcement learning to align better with human judgments. Finally, the explanations produced by the LLM could be turned into user-facing tooltips or side-by-side comparisons to help sellers understand why a suggested price is reasonable, not just what the number is. Overall, our results suggest that retrieval-augmented LLMs are a promising direction for pricing systems, especially as models become cheaper and more reliable. At the same time, simple baselines remain surprisingly strong, and any real deployment would likely combine both approaches.

11 AI Disclosure

- Did you use any AI assistance to complete this proposal? If so, please also specify what AI you used.
  - No, we did not use AI assistance.

If you answered yes to the above question, please complete the following as well:

- If you used a large language model to assist you, please paste *all* of the prompts that you used below. Add a separate bullet for each prompt, and specify which part of the proposal is associated with which prompt.
  - your response here
- **Free response:** For each section or paragraph for which you used assistance, describe your overall experience with the AI. How helpful was it? Did it just directly give you a good output, or did you have to edit it? Was its output ever obviously wrong or irrelevant? Did you use it to generate new text, check your own ideas, or rewrite text?
  - your response here

References

Fathalla, A., Salah, A., Li, K., Li, K., and Francesco, P. (2020). Deep end-to-end learning for price prediction of second-hand items. *Knowledge and Information Systems*, 62(12):4541–4568.

Han, L., Yin, Z., Xia, Z., Guo, L., Tang, M., and Jin, R. (2019). Vision-based price suggestion for online second-hand items. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 1988–1996.

Han, L., Yin, Z., Xia, Z., Tang, M., and Jin, R. (2020). Price suggestion for online second-hand items with texts and images. In *Proceedings of the 28th ACM International Conference on Multimedia*.

Hasan, M. H., Jahan, M. A., Ali, M. E., Li, Y.-F., and Sellis, T. (2024). A multi-modal deep learning based approach for house price prediction. *arXiv preprint arXiv:2409.05335*.

Jiang, H. (2025). Machine learning models for predicting second-hand house prices: A comparative study. In *Proceedings of the 2025 International Conference on Big Data, Artificial Intelligence and Digital Economy (BDAIE)*. ACM.

Semwal, A. and Sharma, S. K. (2025). Deep learning and traditional algorithms: A comparative study on predicting second-hand car prices. In *2025 3rd IEEE International Conference on Industrial Electronics: Developments Applications (ICIDeA)*, pages 1–6.

Sun, N., Bai, H., Geng, Y., and Shi, H. (2017). Price evaluation model in second-hand car system based on bp neural network theory. In *2017 18th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)*, pages 431–436.

Wang, H., You, S., Zhang, Q., Xie, X., Han, S., Wu, Y., Huang, F., and Chen, J. (2025). Llp: Llm-based product pricing in e-commerce.

Xu, X. and Zhang, Y. (2022). Second-hand house price index forecasting with neural networks. *Journal of Property Research*, 39(3):215–236.

Zhang, H., Li, Y., and Branco, P. (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.

12 Code

Click here