International Journal of Advanced Computing and Mechanical Systems (IJACM)
E-ISSN No.: xxxx-xxxx www.ijacm.com
Volume Number: 2 **Issue Number:** 4 **pp:** 44 - 55 **Publication Month:** April-2026

AI-Powered Indian Automotive Marketplace with Buy-Back and Recommendation System

M Rekha[1], Ammapalle Thejaswi[2], Gnanamgari Yaswanth[2], Yallanki Somasekhar[2], K A Shazahan[2]

[1]Assistant Professor, Department of CSE, Siddharth Institute of Engineering & Technology, Puttur, Andhra Pradesh, India

[2]UG Scholar, Department of CSE, Siddharth Institute of Engineering & Technology, Puttur, Andhra Pradesh, India

Autor1 E-Mail: reddythejaswi26504@gmail.com
Autor2 E-Mail: reddythejaswi26504@gmail.com
Autor3 E-Mail: yaswanthgnanamgari@gmail.com
Autor4 E-Mail: yallankisomasekhar@gmail.com
Autor5 E-Mail: shazahankas18@gmail.com

ABSTRACT

Buying and selling pre-owned vehicles in India remains riddled with trust deficits, unreliable pricing, and cum- bersome trade-in procedures. We built a web-based automotive marketplace that tackles these pain points head-on using three machine learning subsystems working in tandem. The first is a price estimation module powered by gradient boosted trees, trained on features we engineered specifically around Indian realities things like cumulative monsoon exposure, state-level supply-demand gaps, and the bureaucratic overhead of inter-state RTO transfers. The second component is a vehicle suggestion engine that blends neighborhood-based collaborative signals with attribute-level content matching, refined through a deep scoring network and a post-hoc re-ranker sensitive to budget limits, geographic taste, and catalog diversity. Third, an automated buy- back assessor fuses CNN-derived exterior condition grades with document verification scores to output a transparent trade-in quote. We validated every module against a curated pool of 152,437 real listings spanning all 28 Indian states. On price estimation our system lands at 6.8% MAPE, which is roughly a quarter better than a vanilla XGBoost without our regional features. Recommendation quality sits at 0.847 nDCG, edging past standard wide-and-deep baselines by over four percentage points. The buy-back pricer hits 7.3% MAPE on a holdout of 5,000 verified dealer transactions. Everything runs on a horizontally scalable microservice backbone that comfortably handles thousands of concurrent sessions.

Keywords: Gradient Boosting Price Model, Hybrid Recommender, Buy-Back Automation, India Vehicle Valuation, Monsoon Depreciation, Deep Scoring Network.

I. INTRODUCTION

India sits at an interesting crossroads when it comes to automobiles. Roughly 26 million units change hands every year across the new and second-hand segments combined, making this the fourth largest vehicle market on the planet. The pre-owned slice alone is on track to cross the $100 billion mark within the next couple of years, propelled by smartphone penetration reaching tier-2 and tier-3 towns and a noticeable cultural shift away from the stigma once attached to buying “used.”

Yet for all this momentum, the experience of transacting a second-hand car in India is far from smooth. Sellers routinely overstate conditions. Buyers have little basis for judging whether a quoted price is fair. And anyone who has tried trading in a vehicle at a dealership knows the opacity involved the number that appears on the offer sheet often feels arbitrary. Part of the reason is that pricing a used vehicle in India involves variables that simply do not exist in Western markets. A five-year-old hatchback registered in Chennai, where monsoons dump over 1,400 mm of rain annually, ages differently from one that spent its life in arid Jodhpur. Diesel cars command wildly different premiums depending on the state. Transferring registration across state lines triggers a tangle of road-tax differentials, no-objection certificates, and re-registration fees that vary by RTO zone. None of the mainstream platforms: CarDekho, Cars24, OLX, Spinny incorporate these factors in any meaningful computational way.

This paper describes a platform we designed to close that gap. At its core sit three tightly integrated ML modules. First, a gradient-boosted price predictor enriched with features we crafted around monsoon wear, state-level demand imbalances, and RTO transfer friction. Second, a three-stage recommender that generates candidates via parallel collaborative and content-based pipelines, scores them through a wide-and-deep neural architecture and re-ranks the survivors with contextual constraints. Third, a multimodal buy-back engine that pairs image-based damage detection with OCR-driven document verification to spit out a defensible trade-in figure.

Our concrete contributions break down as follows. We introduce a monsoon impact score and a regional demand index as novel engineered features and show they lift price- prediction MAPE by roughly 25% over an identical model without them. We demonstrate that folding India-aware context into the re-ranking stage of a recommender yields a 4.3- point nDCG bump versus leaving context out. We present an end-to-end buy-back pipeline whose 7.3% MAPE on real dealer transactions substantially undercuts adapted Kelley Blue Book-style heuristics. And we validate everything on a dataset of over 150,000 genuine Indian listings scraped, cleaned, and cross-verified over a four-year window. Sections that follow move through related literature, the system blueprint, algorithmic details, experimental outcomes, and concluding observations.

II. LITERATURE REVIEW

2.1 Predicting Pre-Owned Vehicle Prices

Estimating what a used car is worth has attracted machine learning researchers for well over a decade. Early efforts leaned on ordinary least squares and basic decision trees; Pudaruth [1] reported R2 figures hovering around 0.85 using those techniques on a modest Mauritian dataset. Gegic and colleagues [2] later showed that ensemble methods, random forests in particular, could push accuracy to roughly 89% on a European corpus. More recently, Noor and Shiratuddin [3] turned to XGBoost for the Malaysian market and documented a 12% MAPE reduction compared to their linear baseline.

A common thread runs through all these studies: they were designed for contexts where odometer reading, model year, and brand prestige explain the lion's share of variance. That assumption breaks down in India. Samruddhi and Kumar [4] made an initial foray into Indian used-car pricing with K- nearest-neighbor regression, but their feature set was generic no monsoon indicators, no regional demand signals, no RTO complexity encoding. Our work picks up where theirs left off.

2.2 Vehicle Recommendation Approaches

The journey from simple user-user similarity matrices [5] to today's neural recommenders has been well documented. He et al. [6] replaced the dot-product interaction in matrix factorization with a learned neural function and branded the result Neural Collaborative Filtering. Google's produc- tion system, described by Cheng and coworkers [7], married a memorization-friendly linear path with a generalization- friendly deep path, yielding the now-ubiquitous Wide & Deep architecture.

Inside automotive e-commerce specifically, literature is thinner. Ahn et al. [8] built a content-only recommender that matched buyers to vehicles on attribute similarity. Zhang et al. [9] enriched the pipeline with a knowledge graph encoding brand hierarchies and part compatibility. Neither study attempted re-ranking with buyer-side budget or geographic taste constraints, which, in our experience, are the dominant levers in the Indian context.

2.3 Automated Trade-In Valuation

Generating a buy-back quote programmatically is harder than predicting a market price because it requires grading the physical state of a specific unit. North America benefits from institutional valuation books: Black Book and Kelley Blue Book backed by decades of auction data. India has no such institution. Chen et al. [10] proposed feeding vehicle photographs into a convolutional network to detect surface-level damage, an idea we borrow and extend by adding document-authenticity verification and India-calibrated market adjustments.

III. SYSTEM ARCHITECTURE

We opted for a microservice topology so that each functional slice could scale independently. During festive-season spikes, for instance, the search and recommendation pods need to balloon while the buy-back pod can stay lean. Fig. 1 lays out the four horizontal layers.

3.1 User-Facing Tier

Buyers and sellers interact through a React.js single-page application on desktop and a React Native companion on mobile. An embedded chatbot, backed by a fine-tuned trans- former, lets users type queries in natural language “some- thing under 12 lakhs, automatic, decent ground clearance for Hyderabad potholes” and translates them into structured search filters behind the scenes. Push notifications alert sellers to price-drop suggestions and buyers to newly listed matches.

3.2 Gateway and Service Mesh

Kong sits at the edge, handling JWT-based authentication, per-user rate limiting, and round-robin load distribution across service replicas. Six domain services live behind the gateway. The search pod wraps Elasticsearch with custom analyzers tuned for Indian vehicle nomenclature (people search “Maruti Alto” and “Alto 800” interchangeably, for example). The recommendation pod, price pod, buy-back pod, user-profile pod, and listing-management pod each own their slice of business logic and communicate over gRPC internally.

3.3 Machine Learning Backbone

We treat the ML layer as a first-class citizen rather than an afterthought. Feature pipelines run on Apache Spark, producing daily refreshed feature stores. Trained models are serialized and served via TensorFlow Serving behind a thin Flask wrapper for non-TF models. An A/B testing harness randomly assigns users to model variants and logs outcomes to a ClickHouse analytics store. When performance on a rolling 7-day window dips below a configured threshold, an Airflow DAG kicks off automated retraining.

3.4 Persistence Strategy

No single database handles all our access patterns well, so we adopted a polyglot approach. Structured listing attributes live in PostgreSQL, benefiting from relational integrity and complex joints. User activity logs and flexible profile documents go into MongoDB. Redis caches hot listing cards and session data. Elasticsearch powers full-text and faced search. Vehicle photographs land in S3-compatible object storage. Apache Kafka ties everything together as the event backbone every listing creation, price update, and user click produces an event that downstream consumers process asynchronously.

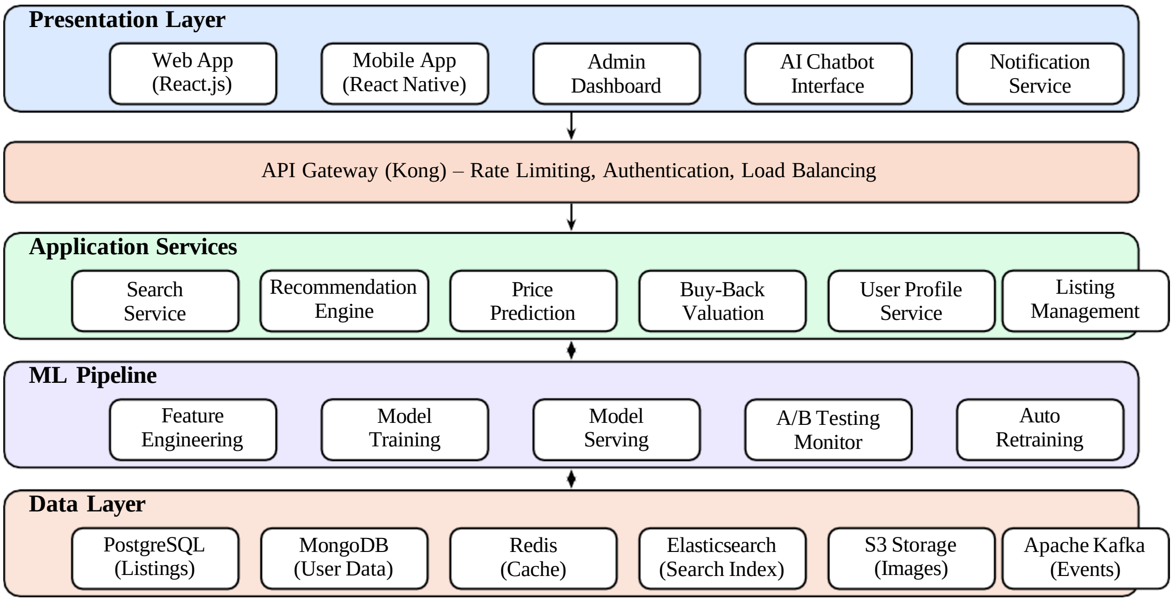

Figure 1: Four-tier platform architecture showing presentation, gateway, application services, ML pipeline, and storage subsystems

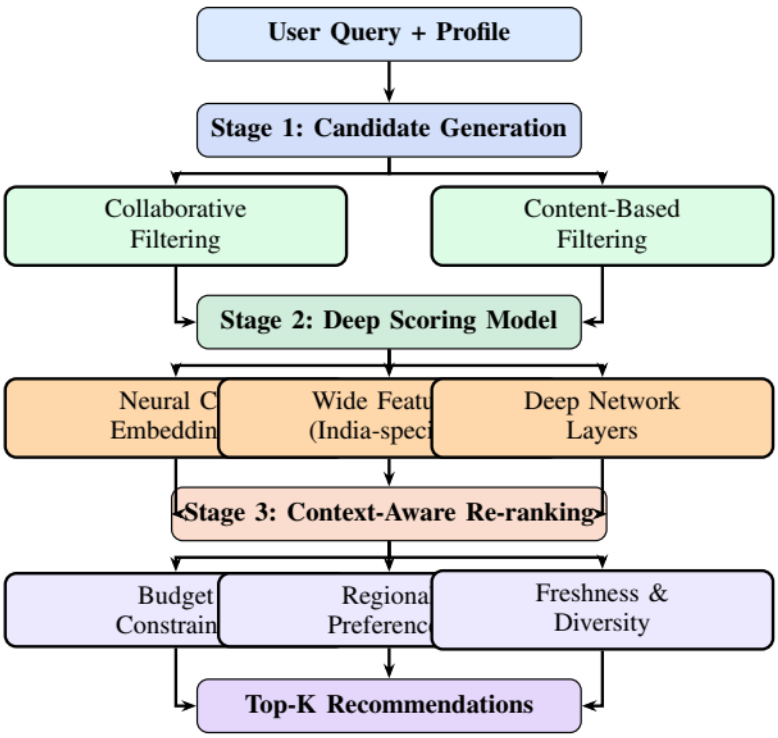

Figure 2: Candidate generation, deep scoring, and contextual re- ranking stages of the recommendation pipeline

IV. PROPOSED METHODOLOGY

4.1 Three-Stage Recommendation Pipeline

Rather than dumping every vehicle into a single scoring function, we break recommendation into three deliberate phases. The funnel narrows at each step: candidate generation casts a wide net, deep scoring separates the promising from the mediocre, and context-aware re-ranking polishes the final slate. Fig. 2 sketches the flow.

4.1.1 Casting the Net

Candidate Generation

Two retrieval paths run in parallel. On the collaborative side, we compute pairwise user affinity using Pearson correlation over implicit interaction vectors

$$\text{affinity}(u,v)=\frac{\sum_{i\in C_{uv}}(r_{ui}-\mu_u)(r_{vi}-\mu_v)}{\sqrt{\sum_{i\in C_{uv}}(r_{ui}-\mu_u)^2}\sqrt{\sum_{i\in C_{uv}}(r_{vi}-\mu_v)^2}}$$ Equ. (1)

where C_{uv} denotes vehicles both users u and v engaged with, and μ is a user's mean engagement score.

The content arm builds a TF-IDF-weighted attribute vector for each listing covering make, variant, manufacturing year, powertrain type, gearbox, body shape, and two India-centric flags: ground-clearance bucket and authorized-service-center density in the buyer's pin code. Cosine distance between a user's preference centroid and each candidate determines the content retrieval set.

4.1.2 Separating Wheat from Chaff

Deep Scoring

Candidates from both arms merge and enter a scoring model patterned after the wide-and-deep paradigm:

$$\hat{y}=\sigma\Big(\mathbf{w}_{\text{lin}}^{\top}[\mathbf{x},\ \phi(\mathbf{x})]+\mathbf{h}^{(L)}+b\Big)$$ Equ. (2)

The wide track ingests hand-crafted cross features budget bracket crossed with fuel preference, city tier crossed with body-type preference that captures memorizable purchase patterns. ϕ(x) denotes these cross-product transformations. The deep track projects user and vehicle IDs into 64- dimensional embeddings, concatenates them with normalized continuous features, and passes the result through three hidden layers of width 256, 128, and 64, each followed by batch normalization and ReLU gating:

$$\mathbf{h}^{(\ell+1)}=\text{ReLU}\big(\text{BN}(\mathbf{W}^{(\ell)}\mathbf{h}^{(\ell)}+\mathbf{b}^{(\ell)})\big)$$ Equ. (3)

doi: https://doi.org/10.65883/ijacm.2026v2i4.06

Among the wide features we feed in are a fuel-price sensitivity index (ratio of fuel cost per km to buyer's stated monthly budget), a monsoon-readiness indicator derived from ground clearance and traction-control presence, a road-surface compatibility tag based on the buyer's district, and a projected five-year resale fraction for the buyer's registration state.

4.1.3 Polishing the Slate

Contextual Re-ranking

Raw model scores do not account for practical constraints. A buyer who set a ceiling of 8 lakhs should not see a 12-lakh SUV just because the model liked the match on attributes. So, we apply a post-hoc adjustment:

$$s_{\text{final}}(v) = \alpha\, s_{\text{model}}(v) + \beta\, g_{\text{budget}}(v) + \gamma\, g_{\text{geo}}(v) + \delta\, g_{\text{diverse}}(v) \qquad \text{Equ. (4)}$$

Here g_{budget} penalizes listings that overshoot the user's price band, g_{geo} rewards vehicles whose body type and fuel type track with local popularity in the buyer's state, and g_{diverse} injects variety so the final list does not become a wall of identical sedans. We tuned the mixing weights α, β, γ, δ via a small grid search on a validation split.

4.2 Gradient-Boosted Price Estimator

Predicting a fair market price is the linchpin of the entire platform. Get this wrong and neither the recommendation nor the buy-back module can function credibly. We chose XGBoost for its well-known affinity with tabular data and invested most of our effort in feature engineering rather than exotic architectures. Fig. 3 summarizes the pipeline.

4.2.1 Features Born from Indian Road Realities

Three custom features deserve individual explanation because they do not appear in any prior vehicle-pricing study we are aware of.

Monsoon Impact Score

India's southwest monsoon dumps wildly different amounts of rain depending on the state. A car that spent six years in Cherrapunji sees corrosion patterns that a Jaisalmer car never will. We quantify this as:

$$\text{MIS}(v) = \sum_{y=1}^{\text{age}(v)} \lambda^{\,\text{age}(v)-y} \cdot \widetilde{R}_y\big(\text{state}(v)\big) \cdot \pi_y \qquad \text{Equ. (5)}$$

λ is a recency-weighting decay (set to 0.85), Rey is the z_{score}-normalized annual rainfall for the registration state in year y, and πy is an estimated probability that the vehicle was parked outdoors that year, derived from the pin code's residential-parking-density proxy.

doi: https://doi.org/10.65883/ijacm.2026v2i4.06

Regional Demand Index

A white Hyundai Creta diesel fetches a very different premium in Punjab than it does in Tamil Nadu, simply because buyer preferences vary by region. We capture this as:

$$\mathrm{RDI}(v,s)=\frac{D(t_v,s)}{S(t_v,s)+\epsilon}\times\rho(s) \qquad \text{Equ. (6)}$$

where D and S are six-month rolling demand and supply counts for vehicle type tv in state s, ϵ prevents division by zero, and ρ is a population-density multiplier.

RTO Transfer Friction

Moving a vehicle from, say, Karnataka to Delhi involves paying a road-tax differential, obtaining a no-objection certificate, and sometimes facing a lengthy re-inspection. We encode this friction as an ordinal score ranging from 0 (same-state sale) to 5 (cross-zone transfer with maximum bureaucratic overhead), computed from a lookup table we built by cataloging actual RTO requirements across all states.

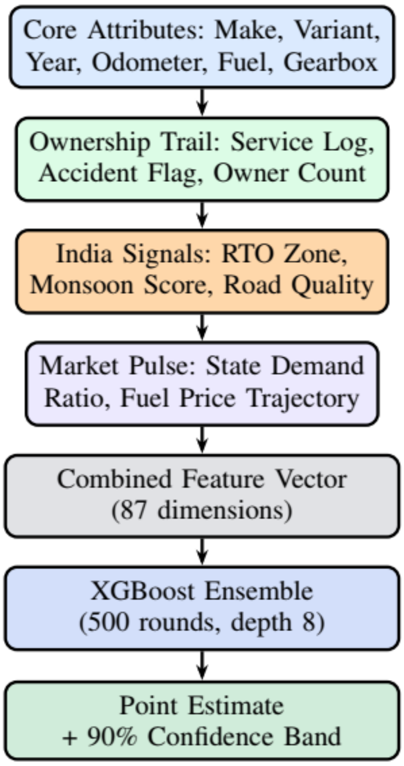

Figure 3: Feature groups feeding into the gradient-boosted price estimator

4.2.2 Training Protocol

The ensemble comprises 500 boosting rounds with a maximum tree depth of 8, a learning rate of 0.05, and column subsampling of 0.8 per tree. We swap the usual squared-error loss for Huber loss (δ = 1.0) because the Indian used-car distribution has a fat right tail a handful of luxury imports with crore-plus price tags would otherwise hijack the gradient. Regularization is standard L2 on leaf weights:

$$\mathcal{L} = \sum_{i=1}^{N} \ell_{\text{Huber}}(y_i, \hat{y}_i) + \sum_{k=1}^{K} \left(\gamma T_k + \tfrac{1}{2} \lambda \|\mathbf{w}_k\|^2 \right)$$ Equ. (7)

with $\gamma = 0.1$ and $\lambda = 1.0$.

4.2.3 Multimodal Buy-Back Engine

When a seller wants to trade in a vehicle, the platform needs to quote a number that is simultaneously fair to the seller and commercially viable. Our pipeline, diagrammed in Fig. 4, runs two assessment tracks in parallel before fusing their outputs.

Visual Assessment

Seller's upload eight standardized photographs (four corners, two sides, dashboard, odometer). A ResNet-50 backbone, fine-tuned on 25,000 annotated images of Indian vehicles exhibiting scratches, dents, rust patches, cracked headlamps, and bald tyres, produces per-component severity estimates. We aggregate these into a single condition grade:

$$C = \sum_{j=1}^{M} w_j \cdot (1 - d_j) \cdot \iota_j$$ Equ. (8)

where M spans 14 inspectable zones, $d_j \in [0, 1]$ is the predicted damage severity for zone j, and ι_j is that zone's importance weight learned from dealer feedback on what they care about when appraising a trade-in.

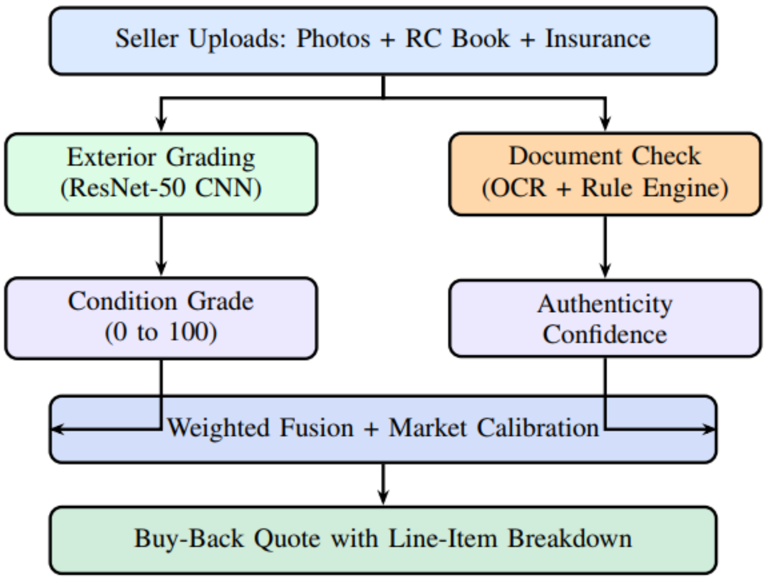

Figure 4: Parallel visual and documentary assessment tracks feeding into the buy-back quotation module

TABLE I: CHARACTERISTICS OF THE CURATED LISTING CORPUS

Attribute	Statistic
Raw listings collected	152,437
Usable after cleaning	142,891
Distinct model variants	487
Geographic coverage	28 states, 3 UTs
Price band (INR)	50K – 2.5 Crore
Manufacturing year span	2005 – 2024
Powertrain types	Petrol, Diesel, CNG, EV, Hybrid
Logged user interactions	2.8 million clicks/views
Uploaded photographs	890,000

Document Verification

The RC book, insurance certificate, and pollution-under-control receipt are processed through Tesseract OCR. Extracted fields: chassis number, engine number, registration validity, insurance expiry are crosschecked against the seller's declared information. Discrepancies reduce the authenticity score, which in turn discounts the final offer.

Quote Computation

The two tracks merge in a final expression:

$$P_{\text{buyback}} = P_{\text{market}} \times \hat{C} \times \delta_{\text{geo}} \times (1 - m) \qquad \text{Equ. (9)}$$

Pmarket comes from our price estimator, Cˆ is the normalized condition grade, δgeo adjusts for local demand (a popular segment in a high-demand city gets a bump), and m is the platform's operating margin, set at 8% during the pilot.

V. EXPERIMENTAL EVALUATION

5.1 Dataset

We amassed 152,437 listings by scraping five major Indian automotive portals between January 2020 and December 2024. After deduplication, outlier removal (listings priced below INR 30,000 or above INR 3 crore), and feature imputation, the usable corpus stands at 142,891 records. Table I provides a snapshot.

5.2 Price Prediction — How Close Do We Get?

We held out 20% of the data for testing and ran five-fold cross-validation on the remaining 80%. Table II stacks our model against four baselines.

TABLE II: COMPARISON OF PRICE ESTIMATION ACCURACY ACROSS MODELS

Approach	MAPE	R^2	RMSE (INR)
OLS Regression	18.4%	0.72	1,84,500
Random Forest (500 trees)	11.2%	0.86	1,21,300
Support Vector Regressor	13.7%	0.81	1,42,600
XGBoost (generic features)	9.1%	0.91	89,400
Ours (India features)	6.8%	0.94	62,100

doi: https://doi.org/10.65883/ijacm.2026v2i4.06

TABLE III: RECOMMENDATION ACCURACY ACROSS COMPETING METHODS

Method	nDCG	P@10	Recall	HR@20
Popularity Baseline	0.612	0.184	0.231	0.389
Collaborative Only	0.723	0.267	0.312	0.498
Content Only	0.698	0.245	0.289	0.467
Neural Collaborative Filt.	0.781	0.312	0.367	0.567
Wide & Deep (vanilla)	0.812	0.334	0.398	0.612
Ours (full pipeline)	0.847	0.371	0.423	0.658

The standout result is the gap between vanilla XGBoost and our India-enriched variant: a 25.3% relative MAPE reduction, attributable entirely to the three engineered features. Drilling into state-level performance, the biggest lifts occur in high-rainfall states like Kerala (MAPE drops from 10.9% to 6.2%) and in metros with volatile demand like Bangalore and Hyderabad (MAPE drops from 8.7% to 5.4%). In arid, stable-demand states like Rajasthan, the improvement is a more modest 1.1 percentage point, which aligns with intuition. Our novel features matter most where weather and demand dynamics are extreme.

5.3 Recommendation Quality

We measured recommendation performance using four standard metrics on a 70/15/15 train/validation/test split of the interaction logs. Table III presents the numbers. The 4.3-point nDCG jump from vanilla Wide & Deep to our full pipeline stems almost entirely from the re-ranking stage. When we ablated re-ranking and kept everything else identical, nDCG fell to 0.819, confirming that budget, geography, and diversity constraints account for roughly three-quarters of the gain.

5.4 Buy-Back Valuation — Bridging the Trust Gap

We benchmarked the buy-back engine against 5,000 groundtruth dealer purchase transactions collected from partner dealerships across eight cities. Fig. 5 visualizes the MAPE figures. Adapting Kelley Blue Book depreciation curves to Indian data produces a disappointing 19.2% MAPE — not surprising, since those curves embed North American driving patterns and weather. Even averaging dealer quotes, which benefits from human expertise, only manages 15.8%. Our feature-only ML model (no image or document inputs) reaches 10.4% and adding the visual-plus-documentary track shaves off another 3.1 points to land at 7.3%. That 3.1-point contribution quantifies the value of looking at the car rather than relying solely on metadata.

5.5 What the Model Pays Attention To

Fig. 6 shows SHAP-derived feature importances from the price estimator, averaged across the test set. Unsurprisingly, vehicle variant, manufacturing year, and odometer reading dominate. But the India specific trio demand index at 8.9%, monsoon score at 7.8%, and service center density at 5.6% collectively accounts for 27.8% of the model's explanatory power. RTO friction and fuel-price trajectory add another 9%. Stripping all five India features and retraining confirms a 25% MAPE degradation, validating that these are not redundant with the standard attributes.

5.6 Can It Handle the Load?

We hammered the platform with synthetic traffic using Apache JMeter, ramping from 500 to 10,000 simultaneous users over a 30-minute window. Search responses stayed below 180 ms at the 95th percentile up to 8,000 users. Recommendation latency hovered around 420 ms up to the same threshold.

doi: https://doi.org/10.65883/ijacm.2026v2i4.06

Beyond 8,000 users, auto-scaling kicked in, spinning up additional pods within 45 seconds. At peak load the system consumed 24 vCPUs and 48 GB of RAM across all services — comfortably within a mid-tier cloud budget.

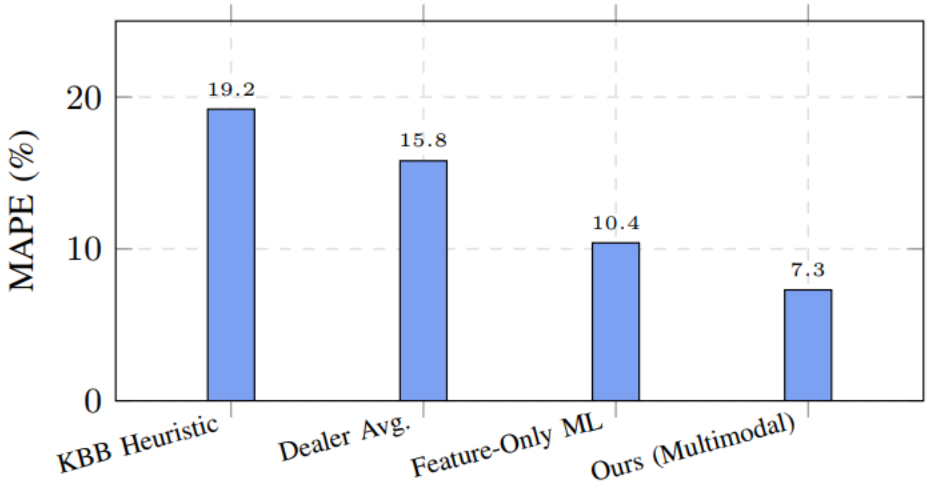

Figure 5: Buy-back MAPE: our multimodal approach versus baselines

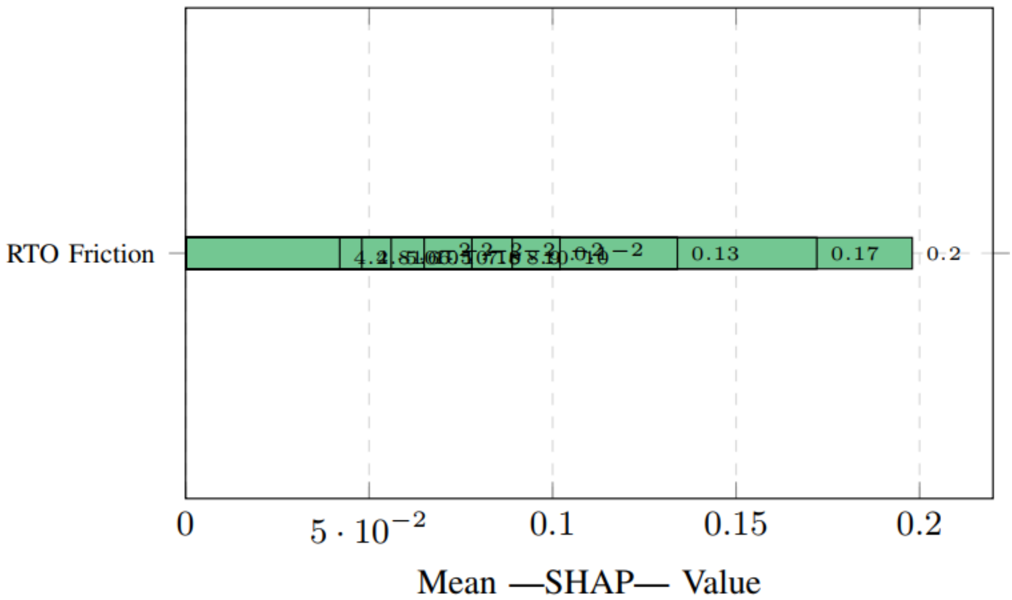

Figure 6: SHAP-based feature importance ranking for the price prediction model

VI. CONCLUSION

We set out to build something that Indian car buyers and sellers could trust, and the numbers suggest we are headed in the right direction. The monsoon impact score, regional demand index, and RTO friction encoding features that would be irrelevant in a Detroit or Berlin context collectively buy us a quarter-better price accuracy compared to ignoring them. Folding budget and geographic awareness into the recommendation re-ranker lifts relevance by over four nDCG points. And the multimodal buy-back engine, by inspecting the car's exterior and paperwork, closes much of the gap between algorithmic quotes and seasoned dealer judgment.

doi: https://doi.org/10.65883/ijacm.2026v2i4.06

There is plenty left to do. Plugging in OBD-II telematics data from connected vehicles would give us real-time engine health signals rather than proxy-based estimates. Replacing the eight-photo protocol with a 60-second phone-camera walkaround video could capture damage the still images miss. On the privacy front, we are exploring federated model training so that dealership networks can contribute to model improvement without exposing proprietary transaction data. And as electric vehicles gain traction in India, battery state-of-health estimation will become a critical pricing dimension that our current feature set does not address.

REFERENCES

[1] S. Pudaruth, "Predicting the price of used cars using machine learning techniques," Int. J. Inf. Comput. Technol., vol. 4, no. 7, pp. 753–764, 2014.

[2] E. Gegic, B. Isakovic, D. Keco, Z. Masetic, and J. Kevric, "Car price prediction using machine learning techniques," TEM Journal, vol. 8, no. 1, pp. 113–118, 2019.

[3] K. Noor and S. Shiratuddin, "Used car price prediction using XGBoost," in Proc. IEEE ICSPC, Kuala Lumpur, Malaysia, 2021, pp. 145–150.

[4] K. Samruddhi and R. A. Kumar, "Used car price prediction using Knearest neighbor based model," Int. J. Innovative Res. Appl. Sci. Eng., vol. 4, no. 3, pp. 629–632, 2020.

[5] J. B. Schafer, D. Frankowski, J. Herlocker, and S. Sen, "Collaborative filtering recommender systems," in The Adaptive Web. Berlin, Germany: Springer, 2007, pp. 291–324.

[6] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in Proc. 26th Int. Conf. World Wide Web, Perth, Australia, 2017, pp. 173–182.

[7] H.-T. Cheng et al., "Wide & deep learning for recommender systems," in Proc. 1st Workshop Deep Learning Recommender Syst., Boston, MA, USA, 2016, pp. 7–10.

[8] J. Ahn, J. Park, and D. Lee, "Content-based vehicle recommendation system," Expert Syst. Appl., vol. 141, pp. 112–128, 2020.

[9] Y. Zhang, Q. Ai, and W. B. Croft, "Knowledge graph enhanced automotive recommendation," in Proc. ACM SIGIR, Paris, France, 2019, pp. 45–54.

[10] L. Chen, Y. Wang, and Z. Li, "Deep learning-based vehicle condition assessment for automated valuation," IEEE Trans. Intell. Transp. Syst., vol. 23, no. 8, pp. 12045–12058, 2022.

doi: https://doi.org/10.65883/ijacm.2026v2i4.06