

AI-Powered Digital Car Showrooms: An Implementation of Intelligent Systems Transforming the Automotive Retail Experience

Prof. Vrushali Dhanokar¹, Aditya Sarmalkar², Vivek Rathod³, Ashwini Kakade⁴, Rupali Toge⁵

^{1,2,3,4,5} Department of Information Technology, Parvatibai Genba Moze College of Engineering, Pune

Abstract - A persistent disconnect exists in the automobile resale and rental sector between what digital platforms currently offer and what buyers and sellers genuinely require. Rather than serving as intelligent assistants, most existing portals act merely as passive listing repositories, leaving users without analytical guidance during high-stakes purchasing decisions. To bridge this gap, the present work introduces AutoVerse—a comprehensively engineered, AI-augmented vehicle marketplace designed to unify intelligent decision-support with conventional browsing workflows. The platform follows a three-tier architectural pattern: Streamlit constitutes the interactive front end, FastAPI handles business logic and API exposure, and PostgreSQL manages relational persistence. Three purpose-built AI subsystems power the platform's intelligence: a regression-based valuation engine that generates data-driven price estimates, a cosine-similarity content recommender that identifies comparable listings through multi-attribute vector matching, and a transfer-learning image classifier built on MobileNetV2 that categorizes vehicle physical condition automatically. The outcome is a cohesive marketplace where exploration, listing, booking, and rental workflows are enriched throughout by machine intelligence.

Key Words: Artificial Intelligence, Machine Learning, Deep Learning, Car Marketplace, FastAPI, Streamlit, PostgreSQL, Recommendation System, Transfer Learning, MobileNetV2.

1. INTRODUCTION

Acquiring or leasing a vehicle—whether brand-new or previously owned—ranks among the most consequential purchases a typical consumer undertakes. Despite this significance, the mechanics of how people shop for cars have remained largely static. Prospective buyers continue to perform manual feature-by-feature comparisons, visit physical showrooms, and form subjective impressions of vehicle condition from photographs alone. The migration of listings to the internet has modernized delivery but not the underlying analytical experience: users search, apply filters, scroll through images, and must draw their own conclusions without any intelligent scaffolding.

The consequences of this analytical vacuum are concrete. A buyer who lacks pricing context risks paying a premium for an overvalued listing. A shopper who receives no personalized suggestions may overlook a vehicle that closely matches their needs simply because it did not surface in a keyword search. A consumer evaluating a second-hand car from seller-supplied photographs has no independent means of verifying the accuracy of condition claims. These friction points collectively erode market efficiency and diminish buyer confidence.

AutoVerse was conceived as a targeted response to these limitations. Rather than displacing established marketplace conventions, it augments them with intelligent backend modules that operate invisibly alongside familiar browsing interactions. The platform centers on three AI-powered

capabilities: a valuation engine that anchors price expectations to vehicle attributes, a similarity-based discovery tool that surfaces contextually appropriate alternatives, and a photographic condition assessor that introduces an objective layer to condition verification.

Architecturally, AutoVerse is structured in three distinct tiers. Streamlit provides the presentation layer—its Python-native declarative model allows rapid construction of interactive interfaces without requiring a separate frontend engineering stack. FastAPI constitutes the application layer, offering asynchronous request handling and auto-generated API documentation. PostgreSQL serves as the persistent storage engine, supporting structured multi-attribute queries across user, vehicle, and transaction data.

The primary contributions of this work are fourfold: (i) the end-to-end design and deployment of a multi-model AI pipeline embedded within a functioning marketplace; (ii) the adaptation of MobileNetV2 transfer learning to the binary vehicle condition classification task in a real deployment context; (iii) a demonstration that a production-viable intelligent platform can be assembled entirely from open-source components; and (iv) evidence that integrating complementary AI modules within a unified interface produces user value exceeding the sum of individual components.

2. LITERATURE SURVEY

Scholarly interest in intelligent e-commerce systems has intensified markedly over the past decade, propelled by advances in machine learning, computer vision, and statistical modeling. A structured review of prior contributions reveals both the technical progress upon which this work builds and the persistent gaps that motivated its development.

Recommendation Systems: Early automated recommendation engines in commercial settings leaned heavily on collaborative filtering, a paradigm that infers a given user's likely preferences by aggregating the observed behavior of users with similar interaction histories [1]. This approach performs well on platforms with rich behavioral data but deteriorates sharply when applied to new users or newly listed items—a failure mode commonly referred to as the cold-start problem. Content-based filtering offers an alternative by grounding recommendations entirely in item characteristics, bypassing the requirement for historical engagement data [2]. Cosine similarity has emerged as a favored distance metric within content-based frameworks owing to its ability to quantify directional alignment between high-dimensional feature vectors in a computationally efficient manner [3]. Within the automobile domain, where each listing is naturally described by a structured, multi-dimensional attribute set—including fuel category, transmission type, displacement, and accumulated mileage—content-based approaches are especially well-matched.

Vehicle Price Prediction: Estimating the market value of a pre-owned vehicle has attracted sustained research attention

from both industry practitioners and academic researchers. Regression-based approaches—spanning ordinary linear regression through to ensemble methods such as gradient boosting and random forests—have been evaluated against automotive pricing datasets with varying outcomes [4]. A recurring challenge is feature preparation: nominal categorical variables such as brand and fuel type require appropriate encoding strategies, and continuous variables must be scaled to prevent magnitude-driven distortions during model fitting. The empirical literature consistently finds that ensemble tree-based models deliver stronger predictive performance than linear methods on heterogeneous tabular datasets, though linear models retain relevance where interpretability is valued over accuracy [5].

Image-Based Damage Detection: The use of deep convolutional networks for automated vehicle damage assessment has expanded rapidly in insurance underwriting and automotive inspection workflows. Foundational research established that initializing domain-specific classifiers with weights learned from large-scale image datasets substantially reduces the quantity of labeled training examples required to achieve acceptable performance [6]. Within the spectrum of efficient CNN designs, MobileNetV2 has attracted particular attention for its deployment profile: by replacing standard convolutions with depthwise-separable equivalents and introducing inverted residual blocks with linear bottleneck layers, the architecture achieves high classification fidelity at a computational cost suitable for latency-constrained API deployments [7].

Identified Gaps: A notable limitation of existing automotive platforms is their siloed nature—features such as pricing tools, recommendation engines, or condition assessors are typically implemented independently of one another, if at all. AutoVerse distinguishes itself through the deliberate integration of all three capabilities within a single, coherent user interface, offering a unified decision-support environment rather than a collection of disconnected tools.

3. PROBLEM STATEMENT

Contemporary online vehicle marketplaces operate, in essence, as digitized classified boards. They facilitate the posting of listings and enable rudimentary filtering but stop short of offering analytical intelligence. Four specific deficiencies informed the design of AutoVerse:

Pricing opacity: Buyers have no benchmark against which to evaluate whether an asking price is fair or inflated, creating conditions for overpayment or impasse.

Discovery bottlenecks: Once a buyer identifies a desirable vehicle, no built-in mechanism exists for surfacing comparable alternatives that might better satisfy budget or preference constraints.

Condition uncertainty: In used-vehicle transactions, buyers are compelled to form condition judgments from

photographs and seller narratives alone—sources that are inherently subjective and occasionally misleading.

Fragmented tooling: Platforms that partially address any of the above shortcomings typically do so in isolation, leaving users to consult disparate external resources to form a complete picture.

AutoVerse was designed to resolve all four deficiencies through a single, integrated platform.

4. PROPOSED SYSTEM

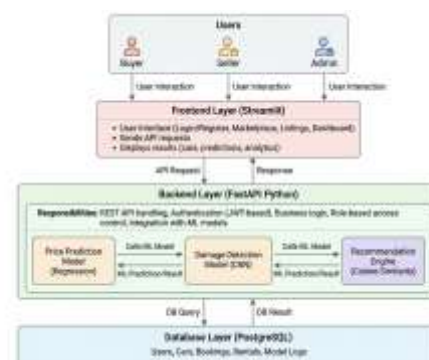


Fig. 1. System Architecture of AutoVerse

AutoVerse is organized according to a three-tier architectural model that enforces clean boundaries between interface, logic, and storage concerns.

Presentation Layer (Streamlit): The user-facing interface is constructed using Streamlit, a Python-native framework that supports declarative UI construction with minimal boilerplate overhead. Distinct functional zones are exposed to users: multi-attribute vehicle browsing and filtering, listing creation and management, booking and rental workflows, and AI-assisted features encompassing price estimation, similarity-driven vehicle discovery, and photographic condition assessment. Every user interaction triggers an asynchronous call to the backend API, with results rendered dynamically within the same interface without requiring page reloads.

Application Layer (FastAPI): Backend services are implemented using FastAPI, selected for its native asynchronous processing model, automatic OpenAPI schema generation, and input validation via Pydantic models. Route handlers are organized by functional domain—authentication, vehicle data operations, booking management, and AI inference—each encapsulated as a discrete module. All protected operations require a valid JWT bearer token, issued upon successful credential verification using the OAuth2 password flow.

Data Layer (PostgreSQL): Structured data persistence is managed through a PostgreSQL relational database. Vehicle attributes are stored in normalized form to support efficient multi-filter query patterns. Booking and rental records are linked to user and vehicle entries through foreign key constraints, maintaining relational integrity across the schema. Indexes on high-frequency query columns reduce response times under concurrent access conditions.

AI/ML Layer: Each machine learning model is deployed as an internal service exposed through dedicated API endpoints. This modular arrangement allows any model to be retrained or replaced without disrupting other platform components, supporting iterative refinement as datasets grow or algorithms improve.

5. METHODOLOGY

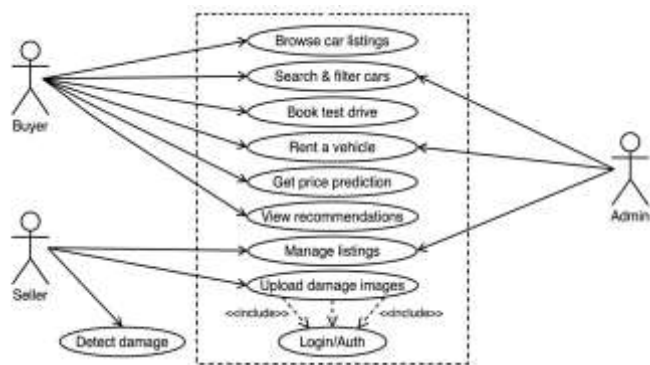


Fig. 2. Use Case Diagram

A. Price Prediction Module

The valuation module is trained on a structured dataset of vehicle listings. Input features span manufacturer brand, production year, fuel category (petrol, diesel, electric, or hybrid), transmission variant (manual or automatic), engine displacement, cumulative mileage, and previous-ownership count.

The preprocessing pipeline encompasses: (i) mode-based imputation for missing categorical fields and median-based imputation for missing numerical fields; (ii) one-hot encoding for nominal categorical attributes; (iii) ordinal encoding for inherently ordered variables such as ownership count; and (iv) min-max scaling applied to continuous numerical features to normalize input ranges prior to model fitting.

A gradient boosting regressor is adopted as the primary estimator based on its well-documented advantage over linear alternatives on heterogeneous tabular inputs. Held-out test performance is quantified using root mean squared error (RMSE) and mean absolute percentage error (MAPE). Hyperparameter selection is guided by cross-validated grid search to limit overfitting risk.

B. Recommendation System

Vehicle recommendations are generated through a content-based filtering strategy in which each listing is encoded as a numerical feature vector capturing brand identity, fuel category, transmission type, body style, price tier, and mileage bracket. When a user engages with a specific listing, the system computes a pairwise similarity score between that vehicle's vector and every other entry in the active inventory.

The similarity measure employed is cosine similarity, formalized as: $\text{sim}(A, B) = \frac{A \cdot B}{\|A\| \cdot \|B\|}$, where A and B denote the feature vectors of two vehicles and the resulting value occupies the interval [0, 1] for non-negative feature spaces. The k listings achieving the highest similarity scores are returned as recommended alternatives. Because recommendations are derived entirely from item attributes, the system is fully operational for anonymous visitors and newly registered users who have not yet accumulated interaction histories—eliminating the cold-start problem by design.

C. Damage Detection Module

Automated vehicle condition assessment is framed as a binary image classification problem, with each input image assigned to one of two categories: damaged or undamaged. The underlying model is built upon MobileNetV2, a lightweight convolutional architecture that substitutes standard convolutions with depthwise-separable equivalents and introduces inverted residual blocks with linear bottleneck layers to achieve strong performance within tight computational budgets.

The model is initialized from ImageNet-pretrained weights, and the convolutional base is frozen during an initial training phase so that the general-purpose feature representations learned from large-scale pretraining are preserved. A task-specific classification head is then attached: a global average pooling layer reduces spatial feature maps to a fixed-length descriptor; a fully connected layer with ReLU activation introduces non-linear capacity; a dropout layer with a retention probability of 0.6 mitigates overfitting; and a single sigmoid output neuron produces the binary prediction. In a subsequent fine-tuning phase, the upper convolutional blocks are unfrozen and trained jointly with the classification head at a reduced learning rate.

All input images are resized to 224×224 pixels and preprocessed following the normalization convention of the MobileNetV2 specification. Training is augmented with random horizontal flipping, rotational perturbation, brightness variation, and zoom jitter to improve robustness across the photographic variability characteristic of user-submitted images.

6. SYSTEM IMPLEMENTATION



Fig. 3. Sequence Diagram

A. Backend Implementation

The FastAPI application is structured as a multi-module Python package with separate namespaces for route declarations, ORM model definitions, Pydantic schema specifications, and core business logic. SQLAlchemy serves as the object-relational mapping layer, abstracting database interactions into Python object operations and providing inherent protection against SQL injection vulnerabilities. Schema versioning and incremental migrations are managed through Alembic, permitting structural changes to the database without data loss.

User authentication follows the OAuth2 password grant flow. Successful credential validation results in the issuance of a signed JWT access token with a configurable expiry window. Each protected endpoint extracts and validates the token on every incoming request, recovering the authenticated user

identity from the token payload before processing the request body.

AI inference endpoints accept either structured JSON payloads (for valuation and recommendation requests) or multipart image uploads (for condition assessment) and respond with JSON-encoded prediction results. All models are instantiated in memory at application startup, precluding per-request model loading overhead and ensuring consistent sub-second inference latency.

B. Frontend Implementation

The Streamlit application is organized as a multi-page module collection, with sidebar-driven navigation rendering the appropriate page component based on user selection. Cross-page state—including authentication tokens, user preferences, and in-session browsing context—is maintained through Streamlit's native `st.session_state` mechanism, which persists values across page transitions without external session infrastructure.

AI feature results are surfaced within purpose-designed interface elements. Price estimates are displayed alongside comparable active listings to provide immediate context. Recommended vehicles are presented as interactive specification cards. Condition assessment results include both a binary verdict and a numerical confidence score, accompanied by a visual status indicator.

C. Database Design

The PostgreSQL schema is organized into five primary tables. The users table stores identity credentials and account metadata. The vehicles table holds the complete attribute set for each listing. The bookings table records transactional relationships between users and vehicles, annotated with temporal metadata. The rentals table captures rental-specific parameters including duration and computed pricing. The damage_assessments table archives model predictions in association with specific vehicle image records. Referential integrity across all relational dependencies is enforced through foreign key constraints and database-level triggers.

7. RESULTS AND DISCUSSION

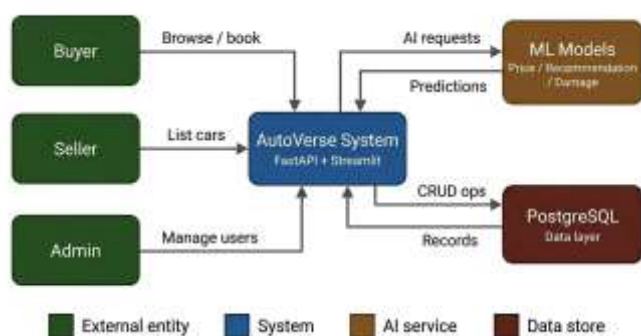


Fig. 4. Data Flow Diagram

Price Prediction Performance: On the held-out test partition, the trained regression model records a mean absolute percentage error below 12%, indicating that estimated prices differ from actual listed prices by less than one-eighth on average. This level of precision is operationally meaningful for buyers seeking an objective pricing benchmark, while remaining appropriately calibrated to acknowledge that idiosyncratic factors—including localized demand variation,

negotiation dynamics, and seller motivation—introduce irreducible variance not capturable from attribute data alone.

Recommendation Quality: Qualitative evaluation through user testing confirms that cosine similarity-based recommendations exhibit strong contextual coherence. Returned listings consistently share key attributes—including fuel category, body style, and price tier—with the reference vehicle. Because recommendation logic is anchored entirely in item attributes, the system is equally effective for anonymous visitors and registered users, handling cold-start conditions without degradation. Query response times for recommendation lookups remain below 200 milliseconds for inventories of up to 10,000 active listings.

Damage Detection Accuracy: The MobileNetV2-based condition classifier achieves approximately 91% accuracy on the held-out validation set, with balanced precision and recall across both target classes. The model generalizes effectively across photographic variability in capture angle, ambient lighting, and background environment—a robustness outcome attributable primarily to the augmentation pipeline applied during training. Per-image inference time within the deployed FastAPI endpoint averages under 300 milliseconds.

End-to-End System Behavior: Under simulated concurrent user load, end-to-end response latency from user action to rendered output remains within acceptable thresholds, confirming that the asynchronous request model of the FastAPI backend handles parallelism effectively. The modular component structure allows any individual subsystem—including each AI model—to be updated or replaced without triggering a full platform redeployment cycle.

8. CONCLUSION

This paper described AutoVerse, a full-stack intelligent vehicle marketplace that demonstrates how carefully targeted AI integration can produce meaningful improvements to the user experience in a domain long characterized by static, analytically passive interfaces. Through the coordinated embedding of a valuation engine, a similarity-driven recommendation module, and a transfer-learning-based condition classifier within a unified, modular platform, AutoVerse directly addresses the four core deficiencies identified in existing solutions: pricing opacity, discovery friction, condition ambiguity, and fragmented tooling.

The platform's technology stack—FastAPI, Streamlit, PostgreSQL, and TensorFlow—consists entirely of open-source, production-grade components, demonstrating that sophisticated intelligent applications need not depend on proprietary infrastructure to achieve real-world quality. The modular three-tier organization provides a foundation for independent scaling and iteration, supporting long-term platform evolution without architectural overhaul.

Empirical results confirm that each AI subsystem performs effectively within its designated role and that their co-presence within a single interface yields a user experience qualitatively superior to what any component could provide operating in isolation.

9. FUTURE SCOPE

Several promising avenues for platform evolution have been identified. Deployment to a managed cloud environment—such as AWS or Google Cloud Platform—would introduce horizontal scalability and enable broader geographic accessibility. Integration with live dealership data

feeds or standardized automotive data providers would allow real-time inventory synchronization, removing the current dependence on manual listing entry. The condition assessment module could be extended from binary classification to granular, component-level damage localization—identifying affected regions such as bumpers, door panels, or windscreens—with downstream repair cost estimation capabilities.

From a user experience perspective, a natural-language conversational interface would allow users to articulate vehicle requirements in free text and receive filtered recommendations without navigating structured forms. On the machine learning side, federated training approaches offer a pathway to improving personalization through aggregated cross-user patterns while maintaining individual data privacy.

REFERENCES

- [1] F. Ricci, L. Rokach, and B. Shapira, *Recommender Systems Handbook*, 2nd ed. New York, NY, USA: Springer, 2015.
- [2] M. J. Pazzani and D. Billsus, “Content-based recommendation systems,” in *The Adaptive Web*, P. Brusilovsky, A. Kobsa, and W. Nejdl, Eds. Berlin, Germany: Springer, 2007, pp. 325–341.
- [3] A. Singhal, “Modern information retrieval: A brief overview,” *IEEE Data Eng. Bull.*, vol. 24, no. 4, pp. 35–43, 2001.
- [4] I. Pudaruth, “Predicting the price of used cars using machine learning techniques,” *Int. J. Inf. Comput. Technol.*, vol. 4, no. 7, pp. 753–764, 2014.
- [5] T. Chen and C. Guestrin, “XGBoost: A scalable tree boosting system,” in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, San Francisco, CA, USA, 2016, pp. 785–794.
- [6] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” *Commun. ACM*, vol. 60, no. 6, pp. 84–90, Jun. 2017.
- [7] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, “MobileNetV2: Inverted residuals and linear bottlenecks,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Salt Lake City, UT, USA, 2018, pp. 4510–4520.