Engineering Systems and Intelligent Technologies
ESIT
ISSN: 3071-253X/© 2026 ESIT (CC BY 4.0) License.
Journal Homepage
https://pub.scientificirg.com/index.php/ESIT

An Intelligent Multi-Task Framework for Used Vehicle Valuation Marketability Classification and Budget-Aware Recommendation

Rasha Y. Youssef[a], **Donia Alaa Eldeen**[a], **Shaimaa Mohamed Elsayed**[a], **and Ahmed Emadeldein**[a,1]

[a] *Department of Information Systems, Faculty of Computers and Information, Suez University, Suez 43221, Egypt;*
E-mails: Rasha-yousef@fci.suezuni.edu.eg, doniaalaaeldeen2002@gmail.com, shaimaamohamed68931@gmail.com, and ahmedemadeldein.dev@gmail.com

ABSTRACT

An intelligent decision support framework for used car evaluation, pricing, and recommendation is presented, integrating acceptability assessment, price estimation, marketability classification, and budget aware recommendation. The Car Evaluation benchmark is used for vehicle evaluation. After removing nineteen invalid marker rows from the 1,729 row source file, 1,710 valid, label consistent instances remain, with no duplicates, and preprocessing including SMOTE balancing is performed within training folds only. Ten classifiers are evaluated using stratified ten fold cross validation with grid search optimization and a soft voting ensemble, reaching a cross validated accuracy of 0.9985, with the leading tree based and boosting models statistically indistinguishable on the test set. A synthetic Egyptian used car dataset of 6,000 listings, generated from documented depreciation and pricing rules, supports the market oriented tasks. CatBoost gives the strongest price estimation performance, with an MAE of 44,768 EGP, a MAPE of 6.0 percent, and an R^2 of 0.986, significantly outperforming Random Forest. The marketability classifier reaches 0.605 accuracy across three demand classes, matching a logistic regression baseline given the deliberately noisy demand label. The recommendation engine, evaluated on 200 simulated buyer queries against a transparent ground truth, achieves a Precision at 5 of 0.88 and an NDCG at 10 of 0.82, measuring recovery of the generator's hidden signals rather than validated real world quality, since the ground truth shares the engine's utility formula. The results demonstrate feasibility for integrating predictive analytics, market assessment, and recommendation into a single framework, with real listing validation left for future work.

PAPER INFORMATION

HISTORY
Received: 11 May 2026
Revised: 30 June 2026
Accepted: 18 July 2026
Online: 31 August 2026

MSC
68T07; 68R10; 94A60; 68M15

KEYWORDS
Multi-Task;
Vehicle Valuation;
Marketability Classification;
Budget-Aware Recommendation;
SMOTE.

1. INTRODUCTION

Selecting a used vehicle is a multi criteria decision problem that cannot be resolved from a single number. A buyer must judge whether a specific car is mechanically and functionally acceptable, whether its asking price is fair given its age, mileage, and condition, whether it fits a fixed budget, and whether a small shortlist genuinely represents the best available options among many candidate listings. A seller faces the mirror problem of judging whether a vehicle is likely to attract demand at a given price. In mature markets this uncertainty is partly absorbed by standardized inspection reports, transaction

[1]Corresponding author at Department of Information Systems, Faculty of Computers and Information, Suez University, Suez 43221, Egypt; E-mail: ahmedemadeldein.dev@gmail.com

price databases, and established resale platforms. In emerging and rapidly growing used vehicle markets such as Egypt's, pricing remains opaque, listings are inconsistent, and buyers and sellers have little independent evidence with which to evaluate a transaction. This is precisely the kind of uncertain, multi attribute, high stakes decision problem that intelligent decision support systems are designed to address. The term intelligent mobility denotes the broader application context motivating this work rather than a claim that the present study evaluates mobility behavior, transportation networks, or connected vehicle systems. The system itself is scoped specifically to used vehicle evaluation, pricing, and recommendation.

Artificial intelligence and machine learning are natural building blocks for such a system because they can learn patterns from heterogeneous, largely categorical, and often imbalanced automotive data and translate them into the specific judgments a used vehicle decision requires, namely an acceptability assessment, a price estimate, a marketability signal, and a ranked shortlist of recommended vehicles. Used in isolation, however, each of these judgments only partially supports the buyer's or seller's decision. A vehicle acceptability classifier alone cannot indicate whether an acceptable car is fairly priced. A price estimation model alone cannot indicate which of many fairly priced cars best matches a stated budget and preferences. A recommendation list is trustworthy only if the price and demand signals feeding it are themselves validated. The decision support value therefore lies not in any single predictive model but in the way vehicle evaluation, price estimation, marketability assessment, and personalized recommendation are connected into one coherent workflow.

One widely studied component of this problem is car acceptability evaluation, in which a vehicle is judged as unacceptable, acceptable, good, or very good on the basis of a small set of descriptive attributes. The canonical Car Evaluation dataset, derived from a hierarchical multi attribute decision model [1, 2], relates acceptability to six categorical features, namely buying price, maintenance cost, number of doors, passenger capacity, luggage boot size, and safety level. Because its underlying concept is well defined yet non trivial for many learners, the dataset serves as a standard benchmark for comparing classification algorithms [3], and it is used in this study as the vehicle evaluation component of the proposed decision support framework.

Acceptability classification, however, answers only one of the questions a buyer or seller asks in practice. Buyers also need to know how much a given car is worth and which vehicles best fit a fixed budget, while sellers need to know whether a car is likely to sell and at what price. These needs are acute in emerging used car markets such as Egypt, where pricing is opaque and listings are inconsistent. Addressing them requires regression for price estimation, classification for marketability assessment, and ranking for budget aware recommendation, in addition to acceptability classification. Because no public, well structured Egyptian dataset with local prices was available, this study constructs a synthetic but realistic market dataset from explicit, documented generation rules. The market track results are accordingly read as a demonstration of decision support pipeline feasibility rather than as validated real market performance.

Despite the maturity of the individual sub fields, acceptability classification, price estimation, marketability assessment, and recommendation are typically studied and reported in isolation, each optimized and evaluated against its own metric with no explicit account of how its output would feed a buyer's or seller's actual decision. This fragmentation is the research gap addressed here, namely the absence of an integrated intelligent decision support framework connecting vehicle evaluation, price estimation, marketability assessment, and personalized recommendation into a single, evaluated, system level workflow for used vehicle selection within intelligent mobility applications.

This paper proposes such a framework, in which vehicle evaluation, price estimation, marketability assessment, and budget aware recommendation are treated as interconnected components of one decision pipeline rather than as independent machine learning experiments. This integration is currently defined at the architectural and mathematical level rather than demonstrated as a single empirical pipeline evaluated record by record across all four components. The vehicle evaluation component performs acceptability classification with ten supervised algorithms, including the gradient boosting systems XGBoost [4], LightGBM [5], and CatBoost [6], enhanced through domain aware ordinal encoding, engineered features, SMOTE balancing applied strictly to training folds, grid search tuning, and a soft voting ensemble. The market oriented components operate on the synthetic Egyptian dataset and deliver a price estimation regressor, a marketability classifier, and a budget aware recommendation engine that consumes the price and marketability outputs together with the buyer's stated budget and preferences.

The contributions of this work are summarized as follows. An intelligent decision support framework is proposed that integrates machine learning based vehicle evaluation, price estimation, marketability assessment, and personalized recommendation for used vehicle selection, together with a formal mathematical formulation of the underlying recommendation problem. A rigorous, leakage controlled comparison of ten classifiers is conducted on the cleaned Car Evaluation benchmark, with ten fold cross validation and formal statistical testing using McNemar and Wilcoxon signed rank tests, with Bonferroni and Benjamini Hochberg multiple comparison correction, serving as the framework's vehicle evaluation component. A comparative evaluation of five regression models for used car price estimation is presented, in which CatBoost significantly outperforms a Random Forest baseline, feeding the framework's price estimation and value for money assessment. A quantitative, system level evaluation of the budget aware recommendation engine is provided using standard ranking metrics, namely Precision at K, Recall at K, MAP, and NDCG, against a transparently defined ground truth that is explicitly acknowledged as partially circular. A fully documented, reproducible methodology is released as runnable code with a fixed random seed, in which the synthetic nature of the market data and all of its generation rules are made explicit, together

with an explicit discussion of the engineering steps required before the framework could be deployed operationally.

The remainder of the paper is organized as follows. Section 2 reviews the related literature on categorical classification benchmarks, gradient boosting systems, class imbalance handling, explainability, and used car price and recommendation research. Section 3 presents the proposed decision support framework, its mathematical formulation, the dataset provenance, the preprocessing and leakage prevention procedures, the classification and market models, the recommendation engine, and the evaluation protocol. Section 4 reports results for vehicle acceptability classification, price estimation, marketability assessment, and recommendation quality, accompanied by statistical testing and a system level summary. Section 5 sets out the priorities for future work, and Section 6 concludes the paper.

2. LITERATURE REVIEW

Supervised learning has been applied extensively to categorical benchmarks, and the Car Evaluation dataset is among the most frequently used. Prior studies report that tree based learners capture its hierarchical concept structure far more effectively than linear models, with decision trees and random forests typically achieving very high accuracy because the target is a deterministic function of the six attributes [1, 2]. Classical algorithm families recur across this literature. The Random Forest [7] aggregates decorrelated decision trees. The Support Vector Machine [8] seeks a maximum margin separating hyperplane. The k Nearest Neighbours algorithm classifies by proximity in feature space [9]. Gradient Boosting builds an additive ensemble of weak learners in a stagewise manner [10]. These methods are conveniently available in the scikit learn library [11].

Beyond these classical methods, modern gradient boosting systems have become the reference approach for tabular prediction. XGBoost introduced a scalable, regularized boosting framework [4]. LightGBM accelerated training through histogram based splitting and leaf wise growth [5]. CatBoost added ordered boosting and native handling of categorical variables [6]. Large scale empirical comparisons find that such tree based ensembles remain competitive with, and often superior to, deep neural networks on medium sized tabular datasets [12, 13], a conclusion also reached by a survey of deep learning applied to tabular data [14]. Comparative evaluations of tuned gradient boosting frameworks generally report that, once each system is properly tuned, the three families perform comparably with no single implementation uniformly dominant, a pattern consistent with the statistically indistinguishable results obtained for these models in the present study. These observations motivate the inclusion of all three boosting systems.

Class imbalance is a well known obstacle in benchmark and real world datasets alike. The Synthetic Minority Over sampling Technique, known as SMOTE [15], interpolates new synthetic minority instances rather than duplicating existing ones and has repeatedly been shown to improve minority class recall and macro averaged F1. Correct usage requires that oversampling be fitted only on training data. Applying it before the train and test split contaminates the evaluation and inflates reported performance. Beyond resampling, class imbalance has also been addressed through synthetic data generation combined with adaptive, class aware loss weighting, an approach recently demonstrated for imbalanced object detection [16], indicating that the combination of synthetic data and imbalance correction used for the Car Evaluation benchmark reflects a broader methodological pattern rather than a domain specific workaround.

Interpretability has likewise become a first class requirement. SHAP unified additive feature attribution methods on solid game theoretic foundations [17], and its tree specific variant enables exact, fast explanations for tree ensembles [18]. Recent surveys consolidate the concepts, evaluation methods, and open challenges of explainable artificial intelligence [19, 20], and a subsequent manifesto charts its interdisciplinary research agenda [21]. Permutation feature importance, a model agnostic global explanation technique, is used in this study to interpret all final models.

For used car price estimation, comparative studies consistently demonstrate that ensemble regressors outperform linear baselines and that vehicle age and mileage are among the strongest price determinants [22, 23, 24]. This literature continues to grow, with recent studies benchmarking gradient boosting regressors against classical baselines on newly collected listings [25] and applying comparable methodology to Middle Eastern used car markets [26], a regional context closely related to the Egyptian market studied here. Tree based models remain the dominant choice, and recent work increasingly pairs XGBoost with SHAP to attribute price predictions to individual features such as mileage and engine size, directly addressing the interpretability need highlighted above [27]. Recommender systems form a mature research field with established evaluation protocols based on Precision at K, recall, mean average precision, and normalized discounted cumulative gain [28], and deep learning approaches have broadened the design space of such systems [29]. Recent car specific recommenders combine content based and collaborative filtering with Random Forest classifiers over natural language buyer preferences to personalize vehicle suggestions [30]. A comprehensive review tracing the field's transition from classical filtering to deep, graph based, and language model driven systems similarly emphasizes rigorous, metric based evaluation as a prerequisite for practical deployment [31], reinforcing the evaluation protocol adopted for the recommendation engine.

Table 1 positions the present study against the most closely related prior work on the component tasks it combines. No prior study in this comparison set evaluates acceptability classification, price estimation, and recommendation together with

formal statistical testing and an explicit accounting of leakage controls. This is the gap the present framework addresses, while making explicit the limitations of its own synthetic market data and its architectural, rather than fully empirical, cross track integration. Acceptability classification, price estimation, marketability assessment, and recommendation are usually studied in isolation, each reported against its own metric with little discussion of how its output would feed a buyer's or seller's decision. The present work differs by integrating all four components within a single, statistically validated intelligent decision support framework, by formally defining how their outputs combine into a final recommendation, and by making explicit, rather than implicit, the limitations of its synthetic market data and its architectural cross track integration. This positioning also extends naturally to the broader body of work on decision support for high value, infrequent consumer purchases, where the central challenge, as in the used vehicle domain, is combining a valuation signal with a personalization layer under budget constraints rather than optimizing either signal in isolation.

Table 1: Positioning relative to related work

Study	Data	Tasks	Leakage control	Statistical testing
[22]	Real	Price only	No	No
[23]	Real	Price only	No	No
[24]	Real	Price only	No	No
[25]	Real	Price only	No	No
[26]	Real (Saudi Arabia)	Price only	Partial	No
[30]	Real	Recommendation only	No	No

3. PROPOSED METHODOLOGY

3.1 Proposed Intelligent Decision Support Framework

The proposed system is an intelligent decision support framework for used vehicle selection rather than a set of independent machine learning experiments. **Figure 1** presents the framework architecture as an explicit decision support workflow proceeding from vehicle information through vehicle evaluation, price estimation, marketability assessment, and user budget and preferences, to the recommendation engine and the final ranked vehicle recommendations.

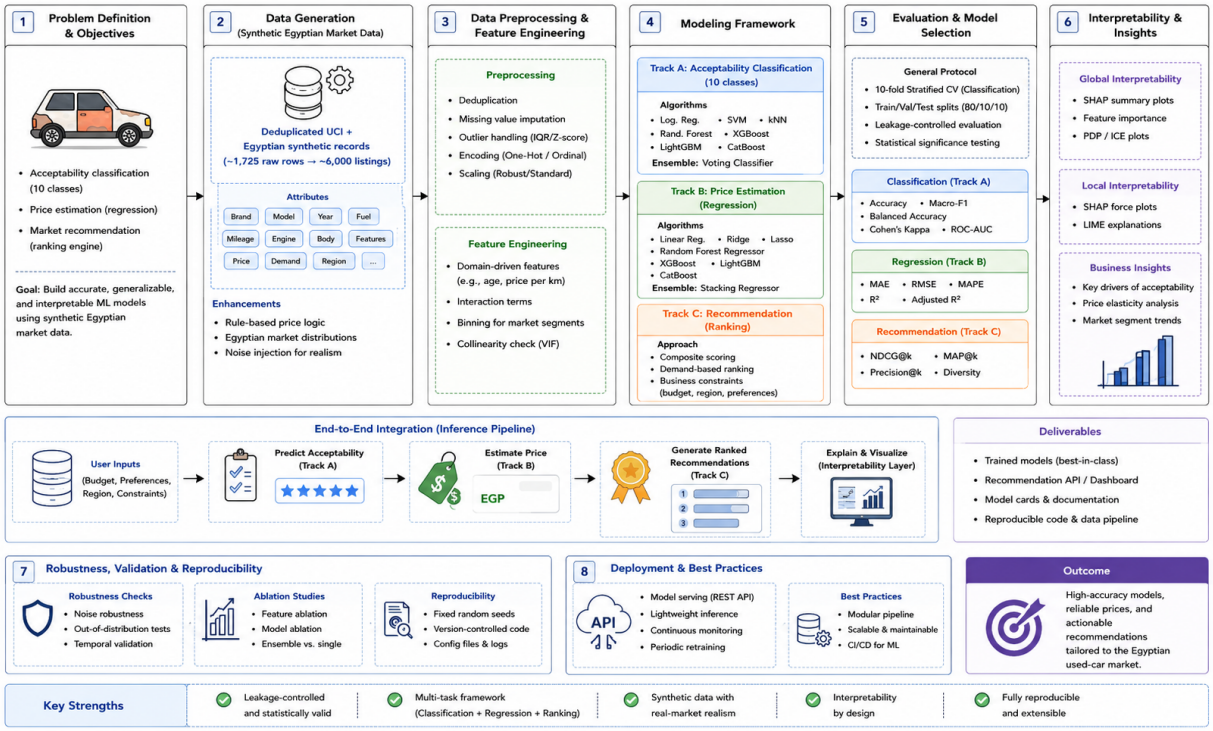

Figure 1: Proposed decision support framework

Raw vehicle information enters a shared preparation stage of data analysis, cleaning, encoding, and scaling, after which the framework branches into two decision support components that are architecturally and logically connected through the recommendation engine's inputs but are trained and evaluated on two separate datasets with non overlapping feature spaces. No single record passes through both tracks in the present study. The vehicle evaluation component, referred to as Track A, assigns each car to one of four acceptability classes using the Car Evaluation benchmark, providing the initial qualitative judgment a buyer or seller would form about a vehicle. The price estimation component, part of Track B, regresses the fair market value of a listing from its attributes. This fair value estimate is the quantity against which a listing's asking price is judged to be a bargain or overpriced, and it is consumed directly by the recommendation engine as the value for money signal described below. The marketability assessment component, also part of Track B, classifies a listing's relative demand as Low, Medium, or High, providing a market liquidity signal that complements price. A fairly priced but low demand vehicle and an identically priced high demand vehicle are not equally attractive to a buyer who may need to resell later. Finally, the recommendation engine takes the buyer's stated budget and preferences together with the price and marketability outputs of the upstream components, filters the inventory to feasible listings, scores each candidate with a weighted utility function, and returns a ranked list of vehicle recommendations. All components are evaluated individually with standard metrics and formal statistical tests, and the framework as a whole is summarized at the system level, showing how each component's output propagates into the final decision.

To make the intended data flow concrete, **Algorithm 1** summarizes the end to end inference workflow. The pseudocode restates the same steps as the narrative description and introduces no additional processing, so that the implementation and the methodology remain aligned.

Algorithm 1 Intended end to end decision support workflow

Require: Vehicle attribute vector $\mathbf{x}_i$, buyer budget B, hard constraints C, preference weights $\mathbf{w}$
Ensure: Ranked top k vehicle recommendations
Clean and validate raw listing attributes
Encode categorical attributes and scale numerical attributes within the appropriate data split
Compute $\hat{e}_i = f_{\text{eval}}(\mathbf{x}_i)$, the acceptability class, where applicable (Track A)
Compute $\hat{p}_i = f_{\text{price}}(\mathbf{x}_i)$, the estimated fair price (Track B)
Compute $\hat{m}_i = f_{\text{market}}(\mathbf{x}_i)$, the marketability class (Track B)
Form the feasible candidate set $\mathcal{F}(B, C)$ by applying the budget ceiling and hard constraints
for each candidate $i \in \mathcal{F}(B, C)$ **do**
    Compute per term utilities $u_k(\mathbf{x}_i, \hat{p}_i, \hat{m}_i)$
    Compute $\text{Score}(i \mid B, C, \mathbf{w}) = \sum_{k=1}^{K} w_k \, u_k(\mathbf{x}_i, \hat{p}_i, \hat{m}_i)$
end for
Rank candidates in $\mathcal{F}(B, C)$ by score and return the top k list

3.1.1 Interaction Between Framework Components and Scope of Integration

The four components function as interconnected decision support elements rather than independent machine learning experiments. Vehicle evaluation contributes an initial acceptability filter. A vehicle assessed as unacceptable is of limited interest to a buyer regardless of price, so its output narrows the candidate space handed downstream. Estimated price affects recommendation directly through the value for money term of the utility function defined below, where the recommendation engine compares each listing's asking price with the price model's fair value estimate, so under priced listings are ranked more favorably and over priced listings are penalized. Any improvement in price estimation accuracy therefore propagates directly into recommendation quality. Marketability contributes a resale liquidity term to the same utility function, so that, all else equal, vehicles predicted to be in higher demand are preferred, reflecting a buyer's practical interest in a vehicle that will be easy to resell later. Buyer budget and preferences are incorporated as hard constraints, namely the budget ceiling and any required transmission or maximum age constraints, that define the feasible candidate set, and as preference weights that determine the relative importance of value, reliability, safety, comfort, and other utility terms for a given buyer profile. The final ranking is generated by scoring every feasible candidate with the resulting weighted utility function and returning the top k vehicles, so the acceptability, price, and marketability judgments of the upstream components, together with the buyer's own constraints, are jointly reflected in a single ranked output rather than reported as separate, disconnected model outputs.

This interaction is defined at the level of the mathematical formulation and the intended production data flow, not as an empirically executed pipeline in which the same 6,000 synthetic listings are first scored by the acceptability classifier and then passed to price estimation and recommendation. The Car Evaluation benchmark's six attributes, namely buying price category, maintenance cost, doors, persons, luggage boot, and safety, do not correspond to fields in the synthetic Egyptian listings, so f_{eval} is not currently applied to the 6,000 listing inventory used in this study. Track A and Track B are therefore integrated architecturally, as interconnected components of one proposed decision support workflow with a shared

mathematical formulation, rather than empirically demonstrated as a single pipeline evaluated record by record across all four components. A concrete feature mapping that would close this gap, deriving ordinal buying price, maintenance, safety, passenger capacity, and door count proxies from the synthetic listing's existing fields such as segment, price tier, maintenance class, safety rating, seats, and doors, is identified as a priority for future work.

3.2 Mathematical Formulation of the Decision Support Problem

This section formalizes the used vehicle recommendation problem solved by the framework. A vehicle listing is represented by an attribute vector $\mathbf{x}_i \in \mathcal{X}$, for $i = 1, \ldots, N$, where $\mathcal{X}$ is the space of vehicle attributes, including model, origin, segment, fuel, transmission, age, mileage, condition, accident history, previous owners, and related fields. Three learned functions are defined on this attribute space in **Equations 1** through **3**.

$$\hat{p}_i = f_{\text{price}}(\mathbf{x}_i) \quad \text{(estimated fair price)}, \tag{1}$$

$$\hat{e}_i = f_{\text{eval}}(\mathbf{x}_i) \in \{\text{unacc}, \text{acc}, \text{good}, \text{vgood}\} \quad \text{(vehicle evaluation score)}, \tag{2}$$

$$\hat{m}_i = f_{\text{market}}(\mathbf{x}_i) \in \{\text{Low}, \text{Medium}, \text{High}\} \quad \text{(marketability score)}, \tag{3}$$

Where f_{price} is the Random Forest price regressor, used specifically for its tree ensemble uncertainty estimate. CatBoost is the most accurate standalone price regressor but is not the model embedded in the recommendation engine in the present implementation. f_{eval} is the acceptability classifier, defined here for generality but exercised only on the UCI benchmark in this study, and f_{market} is the demand classifier. Each listing additionally carries a known asking price p_i^{ask}.

A buyer query is represented by a budget B, a set of hard constraints C, such as a required transmission or a maximum vehicle age, and a preference priority profile $\mathbf{w} = (w_1, \ldots, w_K)$ over K utility terms, including value for money, reliability, safety, age, mileage, running cost, comfort, and resale demand, with $\sum_{k=1}^{K} w_k = 1$ and $w_k \geq 0$. The feasible candidate set for a query is defined in **Equation 4**.

$$\mathcal{F}(B, C) = \{\, i : p_i^{\text{ask}} \leq B \ \text{ and } \ \mathbf{x}_i \text{ satisfies every constraint in } C \,\}. \tag{4}$$

For each feasible listing $i \in \mathcal{F}(B, C)$, a final recommendation score is computed as the weighted combination of normalized utility terms $u_k(\mathbf{x}_i, \hat{p}_i, \hat{m}_i) \in [0, 1]$ given in **Equation 5**.

$$\text{Score}(i \mid B, C, \mathbf{w}) = \sum_{k=1}^{K} w_k \, u_k(\mathbf{x}_i, \hat{p}_i, \hat{m}_i), \tag{5}$$

The value for money term is driven by the price estimation output, as expressed in **Equation 6**.

$$u_{\text{value}} = g\left(\frac{\hat{p}_i - p_i^{\text{ask}}}{\hat{p}_i}\right), \tag{6}$$

For a monotonically increasing bounded function g, so that listings priced below their estimated fair value score higher, and the resale demand term u_{demand} is a monotonic mapping of the marketability score $\hat{m}_i$. The recommendation objective, given in **Equation 7**, is to return the ranked top k list

$$\text{Rec}_k(B, C, \mathbf{w}) = \underset{i \in \mathcal{F}(B,C)}{\arg\,\text{top-k}} \ \text{Score}(i \mid B, C, \mathbf{w}), \tag{7}$$

Subject to the budget and hard constraints already encoded in $\mathcal{F}(B, C)$. This formulation, adapted from the standard top k utility ranking approach used in constrained recommendation [28], makes explicit that the recommendation engine is not a standalone ranking heuristic. It is a decision function of the vehicle evaluation, price estimation, and marketability outputs of the upstream components, parameterized by the buyer's own budget and preferences, and it is against this formal objective that the evaluation protocol and the ground truth construction, including the circularity that construction implies, are discussed directly.

3.3 Dataset Description and Provenance

Two datasets are used in this study, and their roles are kept explicitly separate throughout the paper. The Car Evaluation dataset is a real, publicly available benchmark and is used exclusively for the vehicle evaluation component. No price,

marketability, or recommendation experiment uses this dataset. The synthetic Egyptian Car Market dataset is generated for this study under documented, seeded rules and is used exclusively for the price estimation, marketability assessment, and recommendation experiments. It is never used for the acceptability classification experiments. This separation is maintained so that the very high accuracy obtainable on the real, deterministic Car Evaluation benchmark is never conflated with the market oriented results, which are generated from, and therefore bounded by, the documented synthetic rules described below.

3.3.1 Car Acceptability Dataset

The first dataset is the Car Evaluation database donated to the UCI Machine Learning Repository [3], derived from the hierarchical multi attribute decision model of Bohanec and Rajkovic [1, 2]. The canonical benchmark enumerates all 1,728 combinations of six categorical attributes. The provided source file contains 1,729 rows. Inspection reveals scattered invalid markers across six of the seven columns, namely three in `buying`, four in `maint`, three in `doors`, three in `persons`, two in `lug_boot`, two in `safety`, and two in `class`, rather than duplicated rows, and the valid subset contains zero duplicates. **Table 2** summarizes the provenance audit. After removing the 19 rows containing at least one invalid marker, 1,710 unique, valid attribute combinations remain, and every combination maps to exactly one class label, confirming internal consistency. All experiments therefore use this cleaned set of 1,710 instances. Removing invalid entries is essential, since an invalid categorical value cannot be meaningfully encoded or compared against the canonical attribute domains, and leaving such rows in the data would silently corrupt every downstream encoding step.

Table 2: Provenance audit of the Car Acceptability dataset

Property	Value
Rows in source file	1,729
Rows with at least one invalid marker	19
Rows after validity filtering	1,710
Duplicate rows found in valid subset	0
Unique valid instances used	1,710
Canonical benchmark size (full factorial) [1, 3]	1,728

Table 3 lists the six attributes and the target.

Table 3: Attributes and permitted values

Attribute	Description	Values
buying	Purchase price of the car	vhigh, high, med, low
maint	Cost of maintenance	vhigh, high, med, low
doors	Number of doors	2, 3, 4, 5more
persons	Passenger capacity	2, 4, more
lug_boot	Size of the luggage boot	small, med, big
safety	Estimated safety level	low, med, high
class (target)	Overall car acceptability	unacc, acc, good, vgood

Table 4 reports the class distribution of the cleaned data, which closely matches the canonical benchmark and is strongly imbalanced toward the unacceptable class, motivating the balancing step.

Table 4: Class distribution of the cleaned dataset (n = 1,710)

Class	Meaning	Count	Percentage
unacc	Unacceptable	1,195	69.9%
acc	Acceptable	381	22.3%
good	Good	69	4.0%
vgood	Very good	65	3.8%
Total		1,710	100%

3.3.2 Synthetic Egyptian Car Market Dataset

The second dataset is synthetically generated, because no public, well structured dataset of Egyptian used car listings with local prices was available. Unlike the Car Evaluation benchmark, this dataset is not real world data. Every field is produced by the documented generative rules below so that the assumptions underlying the market track are fully transparent and reproducible. It comprises 6,000 listings sampled from a catalog of 42 real models commonly sold in Egypt, for example Toyota Corolla, Hyundai Elantra, Kia Sportage, MG, and Chery, each with an approximate showroom price. Every listing receives an age, mileage, condition grade, transmission, accident flag, and owner count drawn from plausible, age correlated distributions, and its price is computed from the explicit depreciation model given in **Equation 8** rather than assigned at random.

$$\begin{aligned}\text{price} = \text{showroom\_price} \times D(\text{age}, \text{segment}) \times M(\text{mileage}, \text{age}) \times C(\text{condition}) \times \\ A(\text{accident}) \times O(\text{owners}) \times T(\text{transmission}) \times \varepsilon,\end{aligned} \tag{8}$$

Where D applies an annual value retention factor of 0.85 to 0.92 depending on body segment, with a 3 percent initial drive off loss and a floor of 22 percent of the showroom price. M adjusts price by approximately ±7 percent per 100,000 km of deviation from an expected 18,000 km per year. C equals 1.04, 1.00, 0.88, or 0.74 for excellent, good, fair, or poor condition, respectively. A equals 0.85 if the car has an accident history. O subtracts 3 percent per previous owner beyond the first. T equals 1.03 for automatic and 0.98 for manual transmission. The term ε is multiplicative noise drawn from $N(1, 0.06)$, truncated to $[0.85, 1.15]$. Prices are capped at 98 percent of the showroom price and rounded to the nearest 5,000 EGP. A relative demand label of Low, Medium, or High is produced by scoring each listing on value, reliability, safety, age, mileage, brand origin, liquidity, fuel, transmission, and accident history, adding substantial random noise to that score so the label is not a near deterministic function of the visible attributes, and splitting the noised score at its 30th and 70th percentiles, yielding a deliberately noisy, balanced three class target. **Table 5** lists the principal attributes.

Because the target values obey these documented rules, results obtained on this dataset demonstrate the feasibility and internal consistency of the modeling pipeline, without constituting evidence about real Egyptian market behavior. The pipeline is, however, directly transferable, since the models consume only standard listing attributes and retraining on real scraped listings requires no structural changes.

Table 5: Principal attributes of the synthetic market dataset

Attribute	Description
model / origin / segment	Car model, brand origin, and body segment
fuel / transmission	Fuel type (petrol, hybrid, electric, diesel) and gearbox
age_years / mileage_km	Vehicle age in years and odometer reading
condition / accident	Condition grade and accident history
previous_owners	Number of previous owners
price_egp (regression target)	Used market price in Egyptian pounds
demand_level (classification target)	Relative market demand, Low, Medium, High (30/40/30 split)

Table 6 reports descriptive statistics.

Table 6: Descriptive statistics of key numerical attributes

Attribute	Mean	SD	Minimum	Maximum
Price (EGP)	814,231	631,737	120,000	4,410,000
Age (years)	5.1	4.0	0	15
Mileage (km)	92,569	81,170	0	466,362
Annual maintenance (EGP)	25,534	21,098	6,300	144,899

Table 7 summarizes the generation rules and value ranges used to construct the synthetic dataset, consolidating the depreciation model above with the categorical sampling assumptions used for the remaining fields. All ranges and rules are fixed and seeded, so the dataset can be regenerated exactly.

Table 7: Generation rules and ranges

Variable	Generation rule / range
Vehicle catalog	42 real models sold in Egypt, each with an approximate showroom price
Age (years)	Sampled 0 to 15 years, correlated with mileage and condition
Mileage (km)	Age correlated around an expected 18,000 km per year, with individual variation
Condition grade	Excellent, Good, Fair, or Poor, with probability shifting toward lower grades as age and mileage increase
Accident history	Binary flag, probability increasing with age and mileage
Previous owners	Count increasing stochastically with age
Transmission	Automatic or Manual, sampled per model and segment
Depreciation factor D	Annual retention 0.85 to 0.92 by segment, 3 percent initial drive off loss, floor of 22 percent of showroom price
Mileage adjustment M	±7 percent per 100,000 km deviation from the expected 18,000 km per year
Condition multiplier C	1.04, 1.00, 0.88, or 0.74 for Excellent, Good, Fair, or Poor
Accident multiplier A	0.85 if accident history present, otherwise 1.00
Owner multiplier O	Minus 3 percent per previous owner beyond the first
Transmission multiplier T	1.03 automatic, 0.98 manual
Noise ε	$N(1, 0.06)$, truncated to $[0.85, 1.15]$
Price rounding and cap	Capped at 98 percent of showroom price, rounded to the nearest 5,000 EGP
Demand label	Weighted score over value, reliability, safety, age, mileage, brand origin, liquidity, fuel, transmission, and accident history, plus substantial injected noise with standard deviation 0.9 times the signal standard deviation, split at the 30th and 70th percentiles into Low, Medium, High

3.4 Data Preprocessing and Leakage Prevention

Table 8 summarizes the preprocessing steps together with the specific measures taken to prevent information leakage. Records with invalid or missing entries are removed before any split. For the baseline classifiers, categorical attributes are transformed with label encoding. For the enhanced pipeline, domain aware ordinal encoding preserves each attribute's natural ordering, for example low is less than med, which is less than high, which is less than v high. Four domain driven features are engineered from the ordinal codes, namely a cost index, a capacity score, a safety cost ratio, and a door person interaction term. Each is a row wise function of that row's own attributes, so no cross row information is used. Feature scaling and SMOTE are implemented inside scikit learn and imbalanced learn pipeline objects so that, in every train and test split and in every cross validation fold, they are fitted exclusively on the training portion and merely applied to the evaluation portion. SMOTE is never applied to test data.

Table 8: Preprocessing and leakage prevention measures

Step	Description and leakage control
Data cleaning	Remove invalid or missing entries, restrict every attribute to its valid categories
Encoding	Label encoding, domain aware ordinal encoding
Feature engineering	Four row wise engineered features computed from each instance's own attributes only
Feature scaling	StandardScaler inside a pipeline fitted on training folds only
Class balancing	SMOTE ($k = 5$) inside an imbalanced learn pipeline fitted and applied on training folds only
Splitting	Stratified 80/20 train and test split, stratified 10 fold cross validation, fixed random seed 42

3.5 Classification Models (Track A)

Ten supervised algorithms are compared. The baseline set comprises Logistic Regression in unweighted and class weighted forms, k Nearest Neighbours, a linear kernel Support Vector Machine, a Decision Tree, a Random Forest, and Gaussian Naive Bayes, spanning linear, instance based, margin based, tree based, ensemble, and probabilistic families. The enhanced pipeline adds Gradient Boosting and the three modern boosting systems, XGBoost [4], LightGBM [5], and CatBoost [6], trained on the ordinally encoded data augmented with the four engineered features. The Random Forest is tuned by grid search over the number of trees, maximum depth, and minimum samples per split, using five fold cross validation within the training data, and a soft voting ensemble combines the tuned Random Forest, Gradient Boosting, and k Nearest Neighbours. All enhanced models are assessed with ten fold stratified cross validation on the training partition and finally on the held out test set.

3.6 Market Models (Track B)

Five regression models, spanning linear and tree based or boosting families, are compared for price estimation, namely Linear Regression, Random Forest, XGBoost, LightGBM, and CatBoost, all trained on identical one hot encoded and standardized features covering the car's model, origin, segment, fuel, transmission, condition, accident history, safety, reliability, maintenance class, luggage capacity, age, mileage, previous owners, seats, and doors. A Random Forest classifier predicts the relative demand level based on the listing's attributes. The price derived feature used in earlier development, price relative to the segment's median, is deliberately excluded from the classifier's inputs. Because the demand score generator itself includes a price based value term, that feature was found to leak the label almost directly, producing an artificially easy task with accuracy near 0.89, inconsistent with the label's intended difficulty. The corrected result is reported here. The Random Forest price model additionally returns an uncertainty range derived from the spread of its individual trees' predictions, which is why it, rather than CatBoost, is the model embedded in the recommendation engine.

3.7 Recommendation Engine and Evaluation Protocol

3.7.1 Buyer Profile and Query Simulation

Using the buyer query notation $(B, C, \mathbf{w})$ defined in Section 3.2, 200 buyer queries are simulated to evaluate the engine quantitatively. Each query's budget is drawn uniformly from 350,000 to 2,600,000 EGP, spanning the entry level to premium segments of the synthetic catalog. Each query is assigned one of five pre defined priority profiles at random, namely balanced, value focused, safety focused, comfort focused, and reliability focused, whose exact weight vectors over the eight utility terms determine $\mathbf{w}$. These profiles are author constructed archetypes intended to exercise the engine's sensitivity to preference structure, not derived from a user survey or observed buyer behavior, and should not be read as validated buyer segments. Hard constraints are added stochastically to mimic realistic buyer requirements, with 20 percent of queries requiring automatic transmission and 15 percent imposing a maximum vehicle age of eight years. A query may combine both, neither, or one of the two constraints. Any query whose feasible set $\mathcal{F}(B, C)$ contains fewer than 40 listings is discarded and redrawn, ensuring that every evaluated query has a genuine candidate pool to rank rather than a nearly empty one. **Table 9** presents the utility term weight vectors for the five profiles.

Table 9: Utility term weight vectors for the five priority profiles

Profile	Value for money	Reliability	Safety	Age	Mileage	Running cost	Comfort	Resale demand
Balanced	0.20	0.15	0.15	0.10	0.10	0.10	0.10	0.10
Value focused	0.40	0.15	0.05	0.10	0.10	0.10	0.05	0.05
Safety focused	0.10	0.15	0.35	0.05	0.05	0.05	0.10	0.15
Comfort focused	0.10	0.10	0.10	0.05	0.05	0.10	0.40	0.10
Reliability focused	0.15	0.35	0.10	0.10	0.10	0.10	0.05	0.05

3.7.2 Candidate Selection, Scoring, and Ranking

For each simulated query, the engine first applies the budget and hard constraints to the full 6,000 listing inventory to obtain the feasible candidate set $\mathcal{F}(B, C)$. Every feasible candidate is then scored with the weighted utility function, using the learned price estimate $\hat{p}_i$ from the Random Forest price model to compute the value for money term and the learned marketability score $\hat{m}_i$ to compute the resale demand term. The remaining terms, namely reliability, safety, age, mileage, running cost, and comfort, are computed directly from each listing's attributes. Candidates are ranked by their resulting score, and the top 10 form the engine's recommendation list for that query.

3.7.3 Ground Truth Definition and Its Circularity

The term transparent ground truth denotes a relevance judgment generated by an explicit, disclosed formula rather than by human annotation or a black box process. Ground truth relevance for a query is defined by the same utility function, **Equation 5**, and the same feasible set $\mathcal{F}(B, C)$ as the engine itself, but computed with the data generator's noise free fair value in place of the learned price estimate $\hat{p}_i$ and the generator's underlying, pre noise demand score in place of the learned marketability score $\hat{m}_i$. These noise free generator quantities are withheld from every trained model and used only to construct the evaluation labels. For each query, the 20 feasible listings with the highest true utility form the relevant set, and their graded true utility values are used as gains for NDCG.

3.7.4 Baseline Comparison, Scope and Future Work

A complete evaluation of the recommendation engine would ideally benchmark it against simpler baseline strategies, such as a random recommender, a popularity based recommender that always returns the highest demand feasible listings regardless of price fit, and a content based recommender that ranks purely by attribute similarity to the buyer's stated preferences without a learned price signal. These baselines are not implemented in the present study.

3.7.5 Evaluation Metrics

Performance is reported as Precision at 5, Precision at 10, Recall at 10, MAP at 10, and NDCG at 10, averaged over the 200 queries. Because exactly 10 items are recommended against 20 relevant ones, Recall at 10 has a ceiling of 0.50.

3.8 Evaluation Metrics and Statistical Testing

Classification models are evaluated with accuracy and macro averaged precision, recall, and F1 score, complemented by confusion matrices and one versus rest ROC curves with their areas under the curve. Cross validated results are reported as mean plus or minus standard deviation with a 95 percent confidence interval for the mean, based on Student's t distribution with nine degrees of freedom. Because the ten cross validation folds are constructed from overlapping training data drawn from the same underlying instances, they are not independent observations, and this interval is interpreted as a descriptive, exploratory indicator of stability rather than a formally valid inferential statistic under an independence assumption.

Every enhanced pipeline model, namely Random Forest, Gradient Boosting, Decision Tree, XGBoost, LightGBM, CatBoost, the tuned Random Forest, and the voting ensemble, is compared against a reference model, XGBoost, chosen because it, LightGBM, CatBoost, and the voting ensemble are tied on both cross validated and held out accuracy. Any of the four could equivalently serve as the reference, and the choice carries no claim of superiority. Two complementary tests are applied at a significance level of $\alpha = 0.05$. McNemar's exact test is applied to the held out test set predictions because it directly compares two classifiers on the same paired instances through a binomial test on the discordant pairs and does not require the independence across instances assumption that a simple accuracy comparison would implicitly make. Because it is computed once on genuinely unseen instances, this test serves as the primary basis for the significance claims made here. The Wilcoxon signed rank test is applied across the ten paired cross validation folds as a secondary, corroborating check. It is a non parametric test on paired, non independent measurements of the same models on the same folds, appropriate when the normality of the per fold accuracy differences cannot be assumed for only ten folds, but subject to the fold dependence caveat that limits both cross validation based tests to a descriptive rather than a strictly inferential role. Because every competitor is compared against the same reference model, this constitutes a family of $m - 1$ pairwise comparisons rather than a single test. The exact p value is reported for every comparison, together with Bonferroni and Benjamini Hochberg adjusted values, so that the multiple comparison corrected conclusion can be assessed directly.

Statistical significance and practical significance are kept conceptually distinct throughout the paper. A statistically significant result, such as the price estimation comparison, indicates that an observed difference is unlikely to be due to chance, but a numerically small MAE improvement can still be statistically significant with a large paired sample, just as a numerically large but noisy classification accuracy gap can fail to reach significance with only 342 test instances. Effect sizes, namely accuracy and MAE differences, are therefore reported alongside every p value, and significance and practical relevance are interpreted separately.

Regressors are evaluated with mean absolute error, mean absolute percentage error, and the coefficient of determination R^2. The per instance absolute errors of the best regressor are compared with each competitor's using the Wilcoxon signed rank test at the same $\alpha = 0.05$ significance level.

3.9 Hyperparameter Settings and Reproducibility

Table 10 reports the final hyperparameter settings of every model. Parameters not listed take their library defaults. All experiments use a fixed random seed of 42 for data generation, splitting, resampling, and model training. The software environment comprises Python 3.12.10, scikit learn 1.6.1, XGBoost 3.4.1, LightGBM 4.7.0, CatBoost 1.2.10, NumPy, and pandas. Every script used to produce the results in this paper, covering dataset generation, model training, statistical testing, and recommendation evaluation, is provided as supplementary material and can be executed end to end with this fixed seed to reproduce every reported number exactly.

Table 10: Final hyperparameter settings

Model	Settings
Logistic Regression	max_iter = 1000, C = 1.0 (unweighted and class_weight = balanced variants)
k Nearest Neighbours	$k = 5$, Euclidean distance
SVM	linear kernel, C = 1.0
Decision Tree	Gini impurity, unrestricted depth
Random Forest (default)	100 trees
Tuned Random Forest	grid searched, 300 trees, max_depth = None, min_samples_split = 2
Gradient Boosting	100 estimators, learning rate 0.1, max depth 3
XGBoost (classifier)	300 estimators, learning rate 0.1, max depth 6
LightGBM (classifier)	300 estimators, learning rate 0.1, 31 leaves
CatBoost (classifier)	300 iterations, learning rate 0.1, depth 6
Voting Ensemble	soft voting over tuned Random Forest, Gradient Boosting, and k NN
SMOTE	k_neighbors = 5, applied to training folds only
Price regressors	RF, 200 trees, depth 18, min leaf 3. XGBoost, 400 estimators, lr 0.1, depth 8. LightGBM, 400 estimators, lr 0.1, 31 leaves. CatBoost, 500 iterations, lr 0.1, depth 8

3.10 Proposed System Level Architecture for Engineering Deployment

Beyond model training and offline evaluation, this section outlines, as a design proposal rather than implemented infrastructure, how the three tracks could be assembled into a deployable engineering decision support system, following established machine learning operations practice for taking predictive models from experimentation into operational use [32, 33]. At a proposed ingestion layer, live dealership or marketplace listings would replace the static CSV files used in this study, entering the pipeline through the same encoding, scaling, and feature engineering steps already validated. At a proposed model serving layer, the classification, price regression, and recommendation components would be exposed as independent, versioned services behind a common application programming interface, consistent with containerized microservice patterns recommended for industrial machine learning deployments [32].

At a proposed decision support layer, the three services would be consumed by role specific interfaces, namely a classification and pricing view for sellers listing a vehicle, and a budget aware recommendation view for buyers. Because model behavior can degrade once deployed on live, non stationary market data, the proposed architecture would incorporate continuous performance monitoring and scheduled retraining triggers, following documented practice for sustaining machine learning systems in real world industrial settings [34]. No component of this architecture, including the API layer, monitoring, or retraining triggers, has been implemented or evaluated in this study.

4. RESULTS AND DISCUSSION

4.1 Dataset Characteristics

Figure 2 shows the class distribution of the cleaned Car Acceptability dataset. The unacceptable class accounts for 69.9 percent of instances, while the good and very good classes are strongly under represented at 4.0 percent and 3.8 percent, respectively. A model can therefore attain deceptively high accuracy by favoring the majority class, which is why macro averaged metrics and training fold only SMOTE balancing are used throughout.

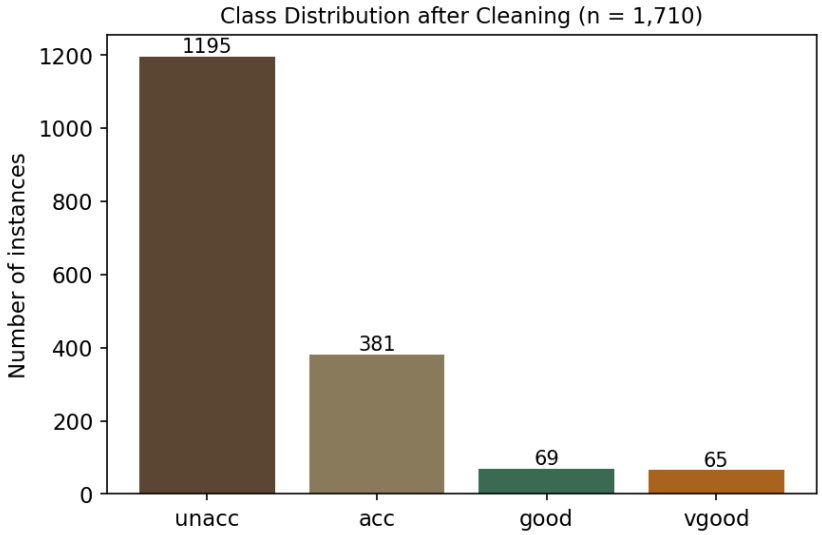

Figure 2: Class distribution after cleaning

4.2 Baseline Classification Results

Table 11 reports baseline results for label encoding on the cleaned data, with a test set of $n = 342$. The best results are obtained with tree based models. Decision Tree yields 0.991 accuracy and 0.985 macro F1, followed by Random Forest at 0.985 and k Nearest Neighbours at 0.965. The linear and probabilistic models achieve much lower accuracy and markedly lower macro F1, reflecting poor performance on minority classes. This ordering is compatible with the rule like hierarchical structure of the dataset, which tree based models may represent directly [1, 12].

Table 11: Baseline classification performance

Model	Accuracy	Macro precision	Macro recall	Macro F1
Logistic Regression (unweighted)	0.678	0.320	0.297	0.292
Logistic Regression (weighted)	0.553	0.373	0.547	0.381
k Nearest Neighbours	0.965	0.971	0.873	0.915
SVM (linear)	0.725	0.322	0.284	0.269
Decision Tree	0.991	0.981	0.990	0.985
Random Forest	0.985	0.963	0.986	0.973
Gaussian Naive Bayes	0.602	0.342	0.465	0.291

4.3 Effect of SMOTE Balancing

Table 12 and **Figure 3** present the change in macro F1 for each model with and without SMOTE, applied only on the training partition within a pipeline. The benefit of balancing is most pronounced for the weaker learners. The macro F1 of Gaussian Naive Bayes improves from 0.291 to 0.380, and that of the linear SVM from 0.269 to 0.387, because the synthetic minority cases strengthen the learning signal for the acceptable and very good classes. The tree based models, already strong performers, change little. As is characteristic with oversampling, k Nearest Neighbours' macro F1 declines under SMOTE, from 0.915 to 0.737, reflecting the precision recall trade off that balancing imposes on an instance based learner sensitive to local class density.

Table 12: Macro F1 score without and with SMOTE

Model	Macro F1 without SMOTE	Macro F1 with SMOTE
Logistic Regression (unweighted)	0.292	0.382
Logistic Regression (weighted)	0.381	0.382
k Nearest Neighbours	0.915	0.737
SVM (linear)	0.269	0.387
Decision Tree	0.985	0.998
Random Forest	0.973	0.964
Gaussian Naive Bayes	0.291	0.380

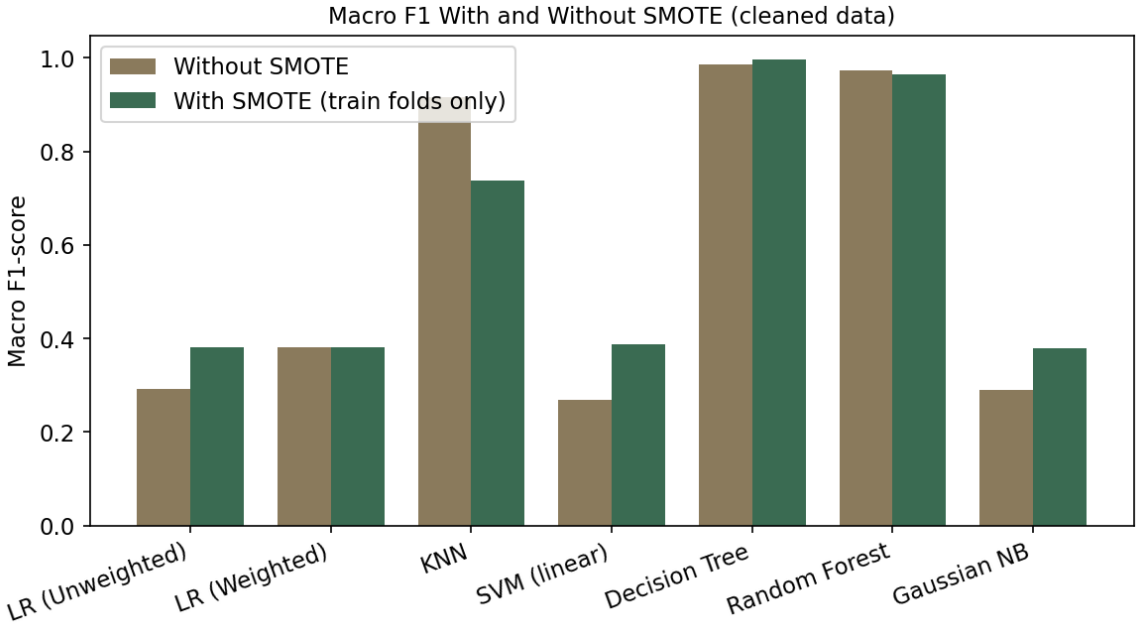

Figure 3: Macro F1 with and without SMOTE

4.4 Enhanced Pipeline Results

Table 13 shows 10-fold cross-validated and held-out test performance for the enhanced pipeline, which combines ordinal encoding with engineered features. XGBoost and LightGBM achieve the highest mean cross-validated accuracy (0.9985 ± 0.0029; 95% CI half-width: 0.0021). XGBoost, LightGBM, Gradient Boosting, Decision Tree, CatBoost, and the voting ensemble achieve perfect accuracy on the held-out test set, while tuned Random Forest reaches 0.9971. These results reflect exceptionally strong predictive performance, but the Car Evaluation benchmark's construction matters for interpretation. The class label is a deterministic function of six input attributes, generated from a fully specified hierarchical decision model rather than derived from noisy real-world observations [1, 2]. The near-perfect performance shows that the enhanced pipeline can recover the underlying rule-based decision function with high fidelity. It does not estimate the accuracy a similarly configured classifier would achieve on noisy, human- or market-derived vehicle acceptability judgments. That assessment would require a dataset with empirically observed labels reflecting actual consumer preferences, expert assessments, or market outcomes.

Table 13: Enhanced pipeline performance

Model	CV accuracy (mean ± SD)	Test accuracy	Test macro F1
Random Forest	0.9868 ± 0.0103	0.9971	0.9978
Gradient Boosting	0.9927 ± 0.0098	1.0000	1.0000
Decision Tree	0.9934 ± 0.0069	1.0000	1.0000
XGBoost	0.9985 ± 0.0029	1.0000	1.0000
LightGBM	0.9985 ± 0.0029	1.0000	1.0000
CatBoost	0.9971 ± 0.0048	1.0000	1.0000
Tuned Random Forest	0.9876 ± 0.0093	0.9971	0.9978
Voting Ensemble	0.9890 ± 0.0128	1.0000	1.0000

The confusion matrix of the reference model, XGBoost, is presented in **Figure 4**.

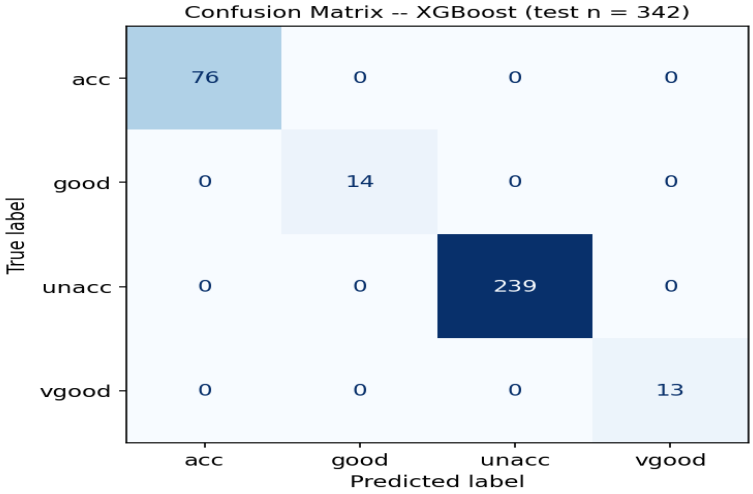

Figure 4: Confusion matrix of the reference model (XGBoost)

One-versus-rest ROC curves for the same model are presented in **Figure 5**. Permutation feature importance for the tuned Random Forest is shown in **Figure 6**. The safety cost ratio is the most influential predictor, followed by `persons` and the original `safety` attribute. These results suggest that the engineered safety cost ratio is highly informative for the model's predictions.

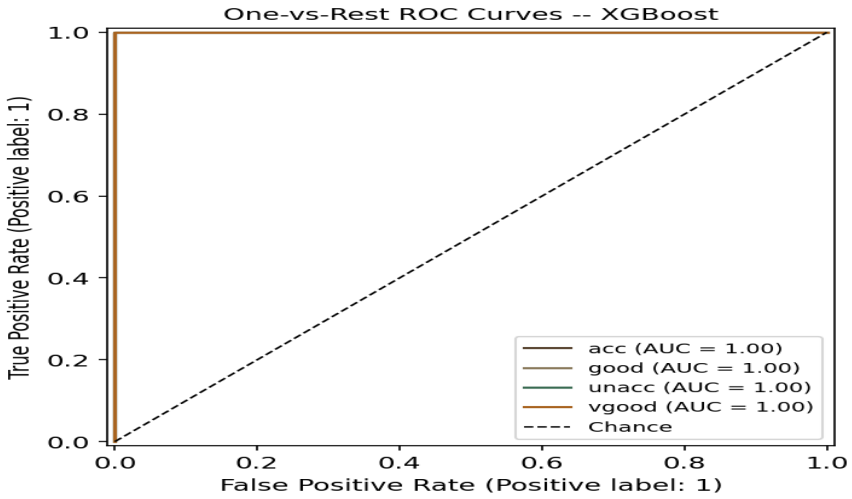

Figure 5: One versus rest ROC curves (XGBoost)

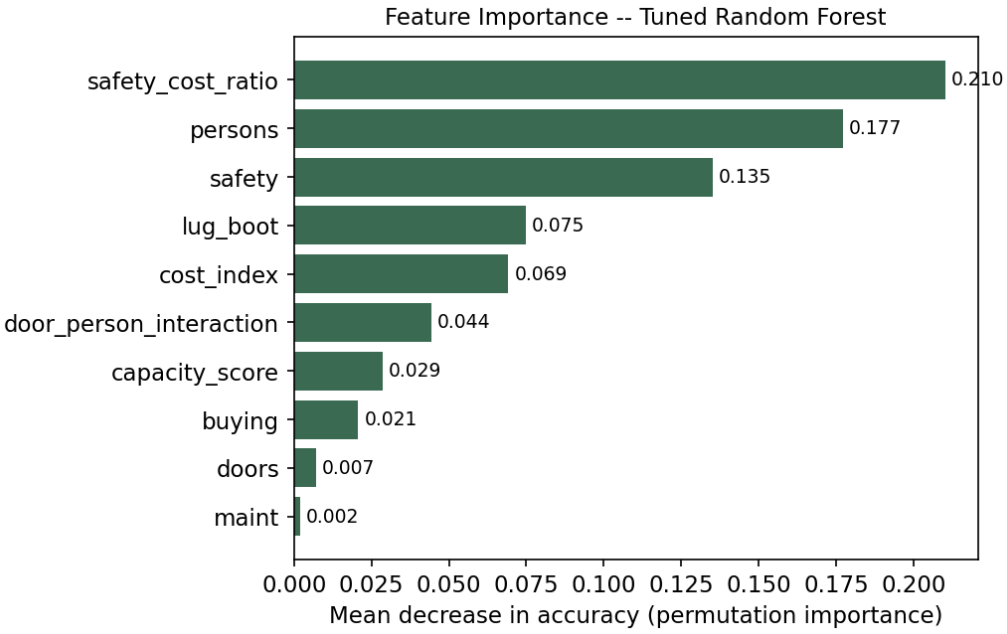

Figure 6: Permutation feature importance (tuned Random Forest)

4.5 Statistical Comparison of Classifiers

Table 14 reports McNemar's exact test between the reference model, XGBoost, and each competitor on the held out test set, together with the Wilcoxon signed rank test across the ten paired cross validation folds and Bonferroni and Benjamini Hochberg adjusted McNemar p values. No comparison reaches statistical significance on the primary, held out set McNemar test, with all p equal to 1.00 both before and after correction. Several models produce predictions identical to XGBoost's on the test set, and the remaining differences, for Random Forest and the tuned Random Forest, amount to a single discordant instance each. The appropriate conclusion is therefore that the leading tree based and boosting models are statistically indistinguishable on this benchmark by the primary test, rather than that any single one of them is superior. XGBoost is designated the reference model for this comparison protocol only, and the choice carries no claim of superiority.

One nuance in the secondary, fold level test is worth noting. The Wilcoxon signed rank test across the ten cross validation folds gives $p = 0.0156$ for both Random Forest and the tuned Random Forest against XGBoost, below the $\alpha = 0.05$ threshold, even though the primary held out McNemar test shows no significant difference for either. Given the fold independence caveat, this is treated as a secondary, exploratory signal rather than evidence overturning the primary conclusion, and is reported explicitly rather than omitted.

Table 14: Statistical comparison against the reference model

Model vs. XGBoost	n_{01}/n_{10}	McNemar p	Bonferroni	Benjamini-Hochberg	Wilcoxon p
Random Forest	1 / 0	1.00	1.00	1.00	0.0156
Gradient Boosting	0 / 0	1.00	1.00	1.00	0.1875
Decision Tree	0 / 0	1.00	1.00	1.00	0.0938
LightGBM	0 / 0	1.00	1.00	1.00	1.0000
CatBoost	0 / 0	1.00	1.00	1.00	0.7500
Tuned Random Forest	1 / 0	1.00	1.00	1.00	0.0156
Voting Ensemble	0 / 0	1.00	1.00	1.00	0.0938

4.6 Price Estimation Results

Table 15 compares the five regression models on the synthetic market dataset. CatBoost is the most accurate, with an MAE of 44,768 EGP, a mean error of 6.0 percent, and $R^2 = 0.986$, followed by LightGBM and XGBoost. Random Forest trails the boosting systems, and Linear Regression is far behind at $R^2 = 0.888$, confirming that the depreciation structure is strongly non linear. This R^2, like every price estimation result in this section, reflects recovery of the documented generative pricing rule rather than validated accuracy against real transaction prices, which are not used anywhere in this

study. Wilcoxon signed rank tests on the per instance absolute errors show that CatBoost's advantage over every competitor is statistically significant, with $p < 0.001$ in all cases. The inclusion of the modern boosting systems thus improves price estimation materially, with MAE falling by roughly 37 percent relative to Random Forest.

Table 15: Price estimation performance

Model	MAE (EGP)	MAPE	R^2	Wilcoxon p vs. CatBoost
Linear Regression	122,965	26.2%	0.888	1.8×10^{-102}
Random Forest	71,200	9.9%	0.966	2.3×10^{-49}
XGBoost	48,530	6.3%	0.983	3.8×10^{-4}
LightGBM	47,968	6.4%	0.984	1.6×10^{-3}
CatBoost	44,768	6.0%	0.986	n/a

The error analysis for the primary Random Forest model, whose fair value estimates drive the recommendation engine, with MAE equal to 71,200 EGP and $R^2 = 0.966$, is presented in **Figures 7** and **8**.

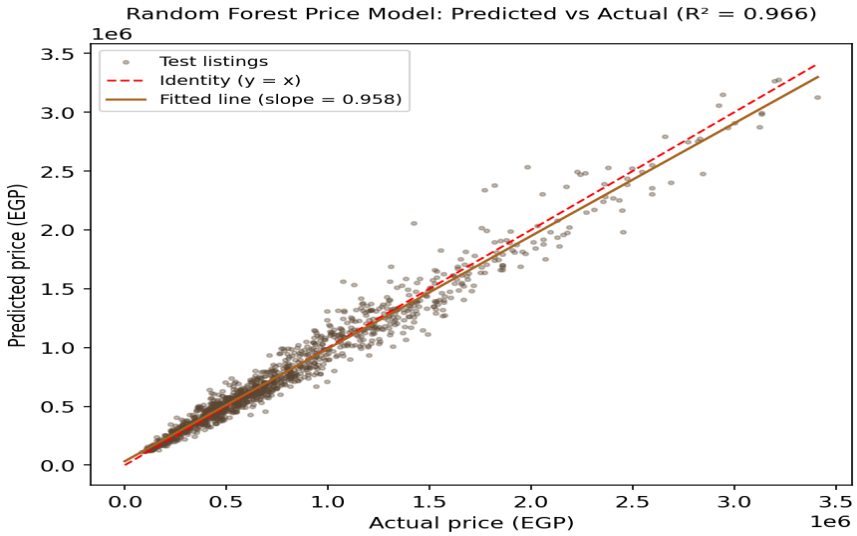

Figure 7: Predicted versus actual price (Random Forest)

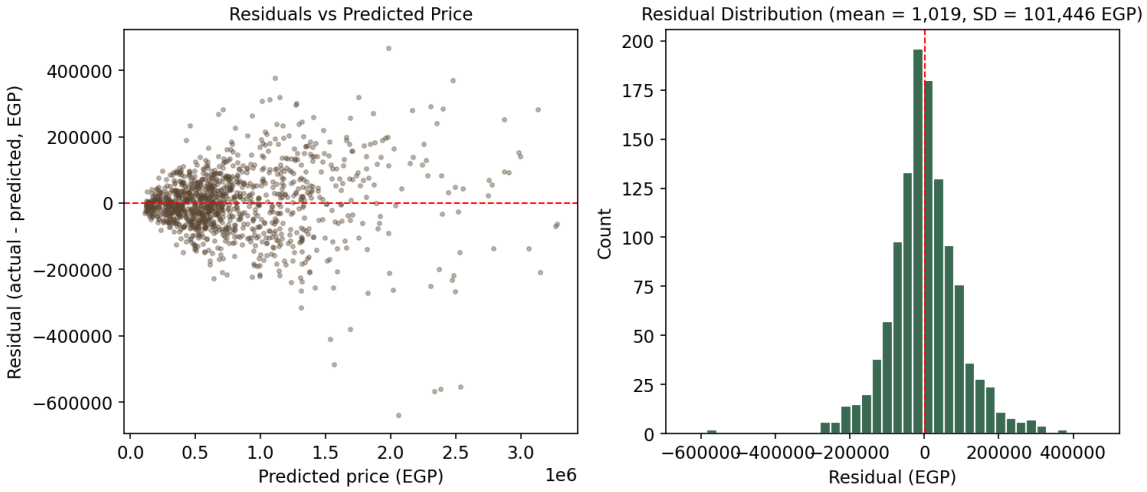

Figure 8: Residual analysis (Random Forest)

Permutation feature importance for the price model is reported in **Figure 9**. The attributes `segment` and `age_years` are essentially tied as the dominant determinants, followed by `model` and `origin`, consistent with the price tier and vehicle age effects built into the generative rule.

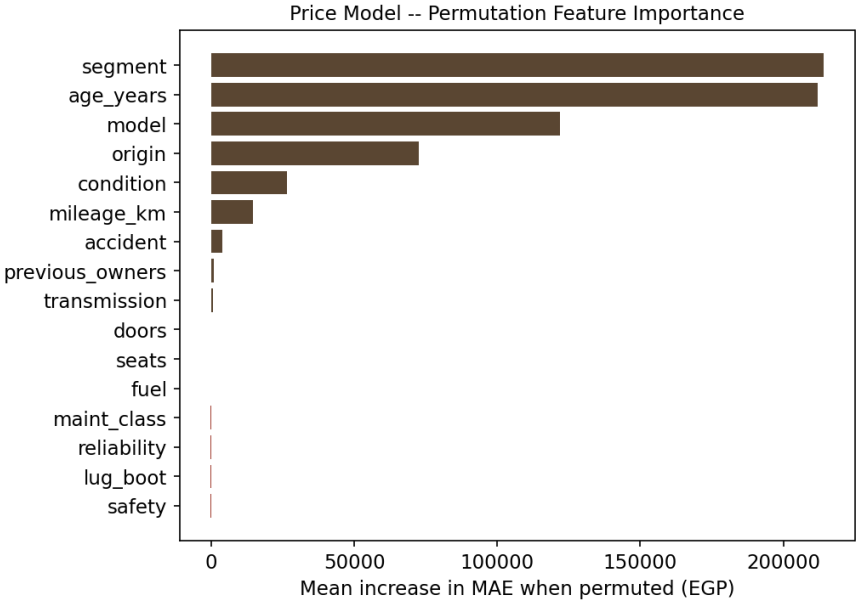

Figure 9: Permutation feature importance (price model)

4.7 Marketability Prediction Results

The marketability classifier attains 0.605 accuracy and 0.613 macro F1 in reconstructing the noised, percentile discretized synthetic demand score from listing attributes. Because the label is itself a function of a documented weighted score with substantial injected noise, this accuracy characterizes synthetic demand score reconstruction difficulty, not real world market demand prediction accuracy. The Random Forest classifier essentially ties a logistic regression baseline at 0.613 accuracy. This result is reported directly rather than implying that the ensemble model provides a clear advantage, since under this demand label construction it does not. **Table 16** reports per class precision, recall, and F1.

Table 16: Per class performance of the marketability classifier

Class	Precision	Recall	F1	Support
Low	0.644	0.647	0.645	360
Medium	0.527	0.562	0.544	480
High	0.684	0.619	0.650	360

The corresponding confusion matrix is shown in **Figure 10**.

Figure 11 shows that reliability is by far the strongest demand signal, followed by safety, with all remaining features contributing comparatively little.

4.8 Recommendation System Evaluation

Table 17 reports the quantitative evaluation of the recommendation engine over 200 simulated buyer queries, following the query simulation, candidate scoring, and ground truth construction protocol previously defined. The engine attains Precision at 5 equal to 0.875 plus or minus 0.264, Precision at 10 equal to 0.798 plus or minus 0.254, Recall at 10 equal to 0.399 plus or minus 0.127 against a ceiling of 0.50, MAP at 10 equal to 0.760 plus or minus 0.304, and NDCG at 10 equal to 0.817 plus or minus 0.267. This NDCG at 10 value is read specifically as evidence that the learned price and marketability models recover a substantial share of the generator's hidden ordering, not as evidence that the ranked list would satisfy real buyers. The observed variance is substantial. The interquartile range is [1.00, 1.00] for Precision at 5, which is median saturated because many queries achieve perfect precision within the top 5, [0.70, 1.00] for Precision at 10 and MAP at 10, and [0.803, 0.998] for NDCG at 10, indicating that recommendation quality is markedly less stable across the query population than a mean alone would suggest. This variance is driven primarily by queries with small feasible sets near the 40 listing minimum and by queries where the learned price and marketability models are locally less accurate within that subset.

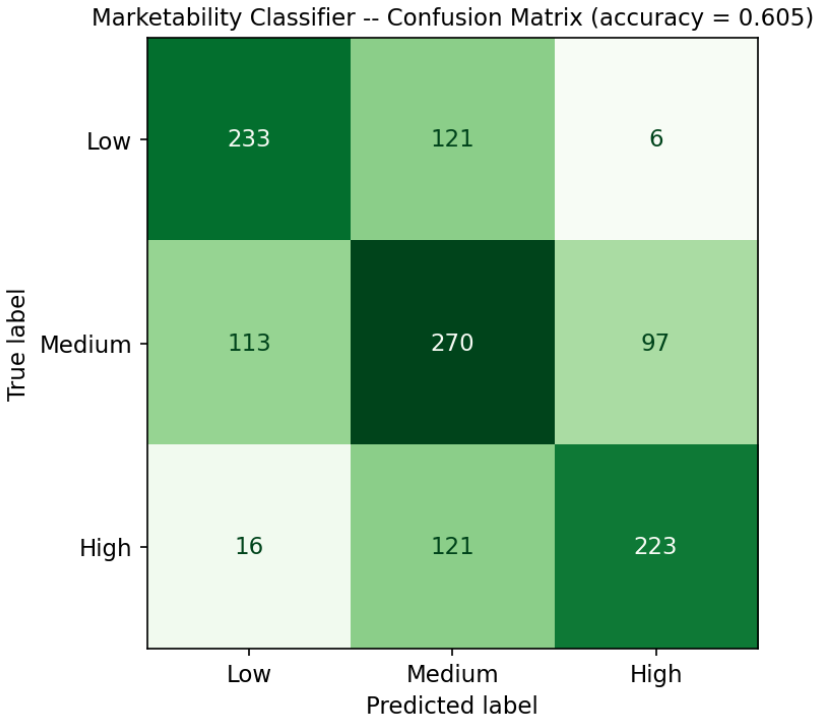

Figure 10: Confusion matrix of the marketability classifier

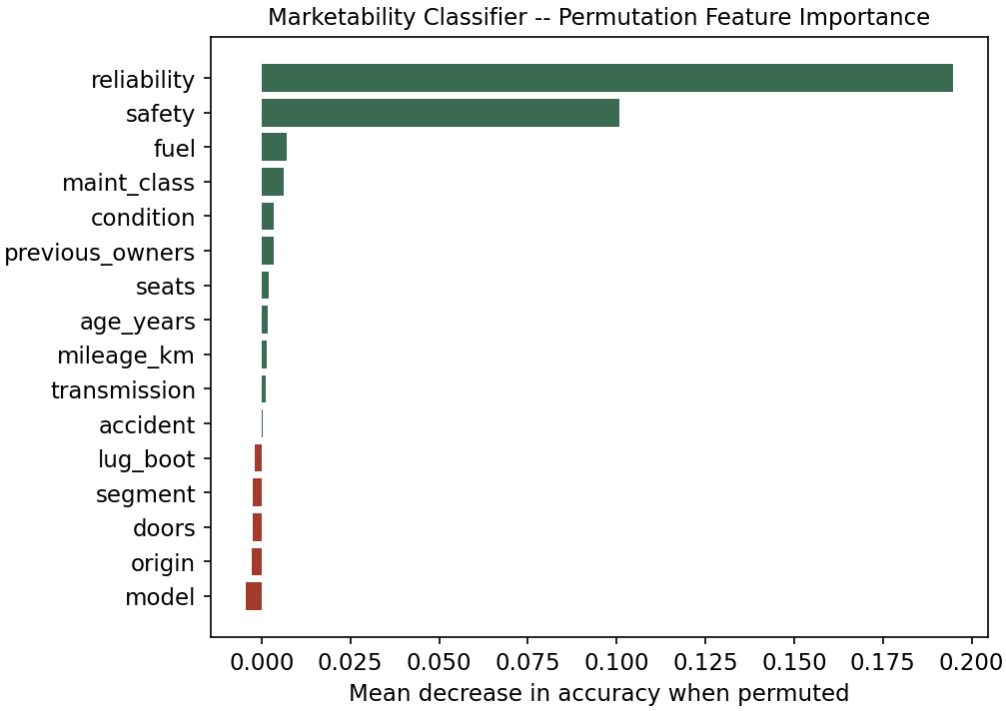

Figure 11: Permutation feature importance (marketability classifier)

Table 17: Recommendation quality over 200 simulated queries

Metric	Mean ± SD
Precision at 5	0.875 ± 0.264
Precision at 10	0.798 ± 0.254
Recall at 10 (ceiling 0.50)	0.399 ± 0.127
MAP at 10	0.760 ± 0.304
NDCG at 10	0.817 ± 0.267

Table 18 illustrates the engine's output for a budget of 500,000 EGP under balanced priorities.

Table 18: Example recommendation output

Model	Age	Mileage (km)	Fuel	Asking (EGP)	Fair est. (EGP)	Score
Peugeot 301	3	37,500	hybrid	425,000	469,019	0.895
Kia Rio	3	84,200	petrol	430,000	467,347	0.876
Renault Logan	1	15,800	petrol	435,000	608,135	0.874
Suzuki Swift	2	31,500	petrol	465,000	503,688	0.866
Chery Tiggo 4	3	62,500	hybrid	500,000	573,937	0.863

4.9 System Level Summary of the Framework

Table 19 summarizes the same results at the system level. For each component, the table states its input, its output, and the metric used for evaluation. This table summarizes independently evaluated component performance and each component's intended role in the framework. It is not an empirical end to end evaluation on a shared record set, since Track A and Track B currently operate on disjoint datasets.

Table 19: System level summary of the framework

Component	Input	Output	Evaluation metric
Vehicle Evaluation	Vehicle attributes (UCI benchmark)	Acceptability class	Accuracy 1.000, macro F1 1.000 (XGBoost, one of four statistically tied top models)
Price Estimation	Vehicle attributes (synthetic listings)	Estimated price $\hat{p}_i$	MAE 44,768 EGP, R^2 0.986 (CatBoost, standalone benchmark). The engine uses Random Forest, with MAE 71,200 EGP, R^2 0.966
Marketability	Vehicle attributes (synthetic listings)	Demand class $\hat{m}_i$	Accuracy 0.605, macro F1 0.613 (reconstruction of noised synthetic label, not real market accuracy)
Recommendation	Vehicle and buyer profile $(B, C, \mathbf{w})$	Ranked top 10 vehicles	Precision at 5, 0.875, MAP at 10, 0.760, NDCG at 10, 0.817 (measures recovery of ground truth ordering)

Read as a system, the framework's overall decision support quality is bounded by the quality of its upstream price and marketability components. Given accurate price and marketability inputs, the ranking mechanism itself would recover the hidden fair value ordering closely, so the framework's practical usefulness on real data depends primarily on how well the upstream price estimation and marketability components transfer to real listings, rather than on the ranking logic itself. This system level framing underlies the limitations and future work priorities discussed below.

4.10 Discussion

Two aspects of the results merit explicit discussion at the modeling level. The very high classification accuracy on the Car Evaluation benchmark is an expected property of the dataset rather than evidence of methodological novelty. The target is a deterministic function of the six attributes, derived from a documented hierarchical decision model, so sufficiently expressive learners can reconstruct the rule [1, 2]. The contribution of the vehicle evaluation component lies in the controlled comparison itself, quantifying the gap between model families, the effect of encoding, balancing, and feature engineering, and demonstrating with McNemar and Wilcoxon tests, including multiple comparison correction, that the leading models are statistically indistinguishable by the primary test.

Every safeguard against information leakage is stated explicitly. Invalid value filtering precedes splitting. Encoders, scalers, and SMOTE are fitted inside pipelines on training folds only. Feature engineering is strictly row wise, and grid search is nested within the training partition. The stability of the ten fold results, with standard deviation at or below 0.013, is consistent with, though does not on its own prove, the absence of leakage, given the fold dependence caveat.

4.11 Positioning Within Intelligent Mobility Applications

The proposed framework is positioned as a component that could support several classes of intelligent mobility applications, without implying that any of them has been operationally deployed or validated here. In an intelligent vehicle selection setting, the framework could act as a buyer facing decision aid embedded in a digital automotive marketplace, surfacing an acceptability judgment, an estimated fair price, and a ranked shortlist for a stated budget in place of unstructured listings. In a mobility decision support platform, the same three signals could feed fleet acquisition or vehicle replacement decisions, where a marketability estimate is directly relevant to resale planning. In a smart transportation or automotive analytics service, the price and marketability estimation components could support market monitoring dashboards that track how estimated fair values and demand shift over time. Personalized vehicle recommendation, as formalized above, is the natural buyer facing entry point for all of these use cases, since it is the component that directly consumes a user's budget and preferences.

These are potential application directions motivated by the framework's design, not claims of deployment. The present study reports offline, cross validated results on a real benchmark and a synthetic market dataset, and no live marketplace integration, real user, or production system was built or evaluated. The section below outlines, at the level of engineering requirements rather than achieved results, what would be needed to move from this offline evaluation to an operational system.

4.12 Deployment Validation and Engineering Implementation Considerations

The results establish predictive feasibility under offline, cross validated conditions. Engineering deployment of the framework would additionally require validation under operational conditions before it could be transferred from research into practice. This includes benchmarking end to end response latency of the recommendation engine as the vehicle catalog grows well beyond the 6,000 synthetic listings used here, and load testing the price estimation and classification services under concurrent user requests, in line with reported industrial experience that large scale deployment introduces engineering constraints such as throughput, monitoring overhead, and infrastructure cost that are not visible during offline model development [34].

The market track can also be framed within the broader engineering practice of treating a live data feed as a continuously updated digital representation of an operating system, an approach used to support real time operational decisions in manufacturing and logistics through digital twin based decision support [35, 36]. Under this framing, the marketability classifier and price estimation regressor play a role analogous to condition monitoring models in predictive maintenance systems, which likewise convert streamed operational data into an early estimate of an asset's remaining value or reliability rather than waiting for a failure, or in this case a sale, to be observed directly [37]. This analogy motivates two concrete engineering extensions beyond the present, synthetic data feasibility study, namely integrating the pipeline with a live listings feed so model outputs can be benchmarked against verified real transactions rather than documented generation rules, and instrumenting the deployed services with monitoring and drift detection mechanisms, so that degradation in classification, pricing, or ranking quality is detected operationally rather than discovered only at the next offline retraining cycle. These steps, rather than further offline tuning, are regarded as the primary remaining work needed to move the framework from a validated research pipeline toward a deployed engineering decision support system.

4.13 Limitations

Synthetic market data. The Egyptian Car Market dataset is entirely synthetic. It is generated from explicit, documented depreciation, condition, and demand rules rather than scraped or observed from real transactions. Consequently, the price estimation, marketability, and recommendation results demonstrate that the modeling pipeline can recover a known generative structure, without constituting empirical evidence about real Egyptian used car prices, demand, or buyer behavior.

Absence of real transaction prices. No real sale price, dealership quote, or verified listing price was used anywhere in the market track. The 6.0 percent mean price error and $R^2 = 0.986$ reported for CatBoost describe how closely the model recovers the generator's own pricing formula, not how closely it would match a real transaction price.

Marketability validated only against a synthetic label. The 0.605 accuracy is validated only against the synthetic, noised demand score. The classifier has not been evaluated against real world demand signals such as listing duration, time to sale, or resale velocity, none of which exist in this synthetic dataset.

Simulated buyer queries and the absence of real user feedback. The 200 evaluation queries are simulated from parametric distributions over budget, constraints, and priority profiles whose weights are author constructed rather than survey derived. No real buyer was surveyed, and no click, purchase, or satisfaction feedback informs the evaluation. The recommendation ground truth is transparent in the sense that its scoring formula is fully disclosed, but it is a formula based construct rather

than a human relevance judgment, and its use of the same utility formula for both engine and ground truth means the reported ranking metrics are partially circular.

No baseline recommendation comparison. The recommendation engine has not been benchmarked against simpler baseline strategies such as random, popularity based, or content based recommendation. No superiority claim over such alternatives is made.

No ablation study. The manuscript does not include a controlled ablation isolating the individual contributions of ordinal encoding, engineered features, and SMOTE balancing to classification performance, nor an ablation of the individual utility terms, namely price, marketability, acceptability filtering, and preference weights, in the recommendation engine. Feature engineering value is supported only by permutation importance, which indicates relevance but not incremental contribution.

Architectural rather than empirical integration of Track A and Track B. The acceptability classifier and the market oriented components are trained and evaluated on datasets with disjoint feature spaces. The framework's end to end data flow is defined mathematically but not empirically demonstrated record by record.

Possible differences between simulated and real market behavior. Real buyers may weight price, brand reputation, and non quantified factors such as styling, dealer trust, and financing terms differently, and real Egyptian market dynamics, including currency fluctuation, import policy, and seasonal demand, are not represented in the synthetic generation rules.

Scope of the acceptability benchmark. The Car Evaluation dataset uses only six categorical attributes and a rule based label, which is why the classification results reach near perfect accuracy. This benchmark does not include price, mileage, or condition information and is therefore evaluated only as the vehicle evaluation component, not as a stand in for a real acceptability judgment on the synthetic listings.

No external validation. No external, independently collected dataset was used to validate any component.

Together, these limitations indicate that the framework should be read as a validated, reproducible, and internally consistent decision support pipeline, whose real world effectiveness for Egyptian used vehicle selection remains to be established on real listings, real transactions, and real users.

5. FUTURE WORK

Several directions are prioritized to move the framework from a validated research pipeline toward one demonstrated on genuine market data. Real listings and verified transaction prices scraped from Egyptian online marketplaces would replace the synthetic dataset, enabling external validation of the price, marketability, and recommendation components against genuine consumer behavior. Real buyer preferences and user feedback, including clicks, purchases, and satisfaction ratings, would be collected to replace the simulated buyer queries and formula based ground truth used in this study.

The recommendation engine would be benchmarked against random, popularity based, and content based baselines to establish whether its learned scoring provides a measurable advantage over simpler strategies. A controlled ablation study would compare original features only, ordinal encoding without engineered features, engineered features without SMOTE, and the complete enhanced pipeline under an identical cross validation protocol, together with an analogous ablation of the recommendation engine's utility terms to quantify each component's marginal contribution to ranking quality. A factorial evaluation of the recommendation engine would cross learned versus generator true price and learned versus generator true marketability across four conditions, two of which, learned paired with learned and true paired with true, are already computed as the primary engine and ground truth results, to directly quantify how price estimation error and marketability error each independently propagate into ranking quality degradation.

A sensitivity analysis would retrain the price and marketability models under perturbed generative rules, for example three percentage point changes to the depreciation retention factors or alternative demand score weightings, to assess whether the performance conclusions are artifacts of the specific rule set chosen. An explicit feature mapping from the Car Evaluation benchmark's six attributes to proxies derivable from the synthetic Egyptian listings would close the architectural gap identified in Track A and Track B integration, allowing the acceptability classifier to be applied to the same inventory used by the market oriented components.

Temporal market changes and geographic variation within the Egyptian market, which the current cross sectional synthetic dataset does not capture, warrant further study, and the synthetic demand label would ideally be replaced with real world marketability signals, such as listing duration, time to sale, or resale velocity, once real listings data become available. Finally, integrating the framework with a live online automotive platform, together with the field validation and engineering benchmarking priorities identified for deployment, represents the natural progression from a validated research pipeline toward operational intelligent mobility decision support.

6. CONCLUSION

An intelligent decision support framework was developed that integrates machine learning based vehicle evaluation, price estimation, marketability assessment, and personalized recommendation into a single workflow, rather than treating them as independent machine learning experiments. The underlying recommendation problem was formalized mathematically, and the resulting system was evaluated both component by component and at the system level, with explicit acknowledgment of where that integration remains architectural rather than empirically demonstrated end to end. Each component was validated with the rigor its role in the framework requires. The Car Evaluation benchmark used for the vehicle evaluation component was audited and cleaned from 1,729 rows, 19 of which contained invalid markers, to 1,710 valid, label consistent instances, with zero duplicate rows found. Every preprocessing step, including SMOTE, was verified to be fitted on training folds only. Ten classifiers were compared under ten fold cross validation with 95 percent confidence intervals, and McNemar and Wilcoxon tests, together with Bonferroni and Benjamini Hochberg correction, showed that the leading tree based and boosting models are statistically indistinguishable by the primary, held out set test, whose near perfect achievable accuracy reflects the benchmark's deterministic, rule based construction rather than real world classification difficulty. On the synthetic Egyptian market data used for the price estimation, marketability assessment, and recommendation components, CatBoost provided significantly more accurate price estimates than a Random Forest baseline, with Wilcoxon $p < 0.001$. A marketability classifier reached 60.5 percent accuracy across three deliberately noisy demand classes, essentially tying a logistic regression baseline, and the budget aware recommendation engine was evaluated with standard ranking metrics against an explicitly defined, and explicitly acknowledged as partially circular, transparent ground truth. The key limitations of this study, stated explicitly rather than implicitly, are that the market oriented components were validated on synthetic data whose generation rules are fully documented but which is not itself real market data, that the recommendation evaluation is partially circular by construction, that no ablation study isolates the contribution of the enhanced pipeline's individual components, and that the acceptability classifier and market oriented components are integrated architecturally rather than empirically demonstrated on a shared record set. Taken together, this framework demonstrates that vehicle evaluation, price estimation, marketability assessment, and recommendation can be combined into one coherent, statistically validated pipeline, and it identifies the specific data and evaluation gaps that must be closed before such a framework could be deployed with confidence on real used vehicle markets.

AUTHOR CONTRIBUTION STATEMENT

All authors contributed equally to the study conception and design. Material preparation, data collection, and analysis were performed by the authors. The first draft of the manuscript was written by the authors, and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

ETHICS APPROVAL AND CONSENT TO PARTICIPATE

Ethics declaration: not applicable. This study did not involve human participants or animals. Therefore, ethical approval and consent to participate are not applicable.

CONSENT FOR PUBLICATION

Consent to Publish declaration: not applicable.

DATA AVAILABILITY

The Car Evaluation dataset used for the vehicle acceptability experiments is publicly available through the UCI Machine Learning Repository. The synthetic Egyptian used car market dataset used for price estimation, marketability assessment, and recommendation experiments was generated by the authors using the documented rules, parameter ranges, and fixed random seed described in this article.

ACKNOWLEDGMENTS

The authors thank the Faculty of Computers and Information, Suez University, for institutional support. The authors are also grateful to the anonymous reviewers, whose detailed comments substantially improved the rigor of this work. In particular, the reviewers identified the dataset provenance inconsistency, the inconsistency between the price model used in the recommendation engine and the standalone regression benchmark, the circularity of the recommendation ground truth, and the need for a controlled ablation study. The authors acknowledge the use of an AI-based assistant (Anthropic's Claude) for assistance with code development, statistical re-analysis, and improving the clarity of the English language. All scientific content, experimental design, interpretations, and conclusions remain the sole responsibility of the authors.

FUNDING

No funding.

DISCLOSURE STATEMENT

The authors declare no conflict of interest.

REFERENCES

[1] M. Bohanec and V. Rajkovic, "Knowledge acquisition and explanation for multi-attribute decision making," in *8th intl workshop on expert systems and their applications*, pp. 59–78, Avignon France, 1988.

[2] M. Bohanec and V. Rajkovic, "Expert system for decision making," *Sistemica*, vol. 1, no. 1, pp. 145–157, 1990.

[3] M. Bohanec, "Car evaluation," *UCI Machine Learning Repository*, vol. 10, p. C5JP48, 1997.

[4] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pp. 785–794, 2016.

[5] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "Lightgbm: A highly efficient gradient boosting decision tree," *Advances in neural information processing systems*, vol. 30, 2017.

[6] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, "Catboost: unbiased boosting with categorical features," *Advances in neural information processing systems*, vol. 31, 2018.

[7] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[8] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[9] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE transactions on information theory*, vol. 13, no. 1, pp. 21–27, 1967.

[10] J. H. Friedman, "Greedy function approximation: a gradient boosting machine," *The Annals of Statistics*, vol. 29, no. 5, pp. 1189–1232, 2001.

[11] J. Hao and T. K. Ho, "Machine learning made easy: a review of scikit-learn package in python programming language," *Journal of educational and behavioral statistics*, vol. 44, no. 3, pp. 348–361, 2019.

[12] L. Grinsztajn, E. Oyallon, and G. Varoquaux, "Why do tree-based models still outperform deep learning on typical tabular data?," *Advances in neural information processing systems*, vol. 35, pp. 507–520, 2022.

[13] R. Shwartz-Ziv and A. Armon, "Tabular data: Deep learning is not all you need," *Information fusion*, vol. 81, pp. 84–90, 2022.

[14] V. Borisov, T. Leemann, K. Seßler, J. Haug, M. Pawelczyk, and G. Kasneci, "Deep neural networks and tabular data: A survey," *IEEE transactions on neural networks and learning systems*, vol. 35, no. 6, pp. 7499–7519, 2022.

[15] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[16] O. Farid, M. Melhi, and A. Somaie, "Optimizing yolov12-l for imbalanced satellite vehicle detection via physics-informed data synthesis and adaptive class-aware loss weighting," *Journal of Smart Algorithms and Applications (JSAA)*, vol. 4, no. 1, pp. 1–21, 2026.

[17] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," *Advances in neural information processing systems*, vol. 30, 2017.

[18] S. M. Lundberg, G. Erion, H. Chen, A. DeGrave, J. M. Prutkin, B. Nair, R. Katz, J. Himmelfarb, N. Bansal, and S.-I. Lee, "From local explanations to global understanding with explainable ai for trees," *Nature machine intelligence*, vol. 2, no. 1, pp. 56–67, 2020.

[19] A. B. Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. García, S. Gil-López, D. Molina, R. Benjamins, *et al.*, "Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai," *Information fusion*, vol. 58, pp. 82–115, 2020.

[20] W. Saeed and C. Omlin, "Explainable ai (xai): A systematic meta-survey of current challenges and future opportunities," *Knowledge-based systems*, vol. 263, p. 110273, 2023.

[21] L. Longo, M. Brcic, F. Cabitza, J. Choi, R. Confalonieri, J. D. Ser, R. Guidotti, Y. Hayashi, F. Herrera, A. Holzinger, R. Jiang, H. Khosravi, F. Lecue, G. Malgieri, A. Páez, W. Samek, J. Schneider, T. Speith, and S. Stumpf, "Explainable artificial intelligence (xai) 2.0: A manifesto of open challenges and interdisciplinary research directions," *Information Fusion*, vol. 106, p. 102301, 2024.

[22] N. Pal, P. Arora, P. Kohli, D. Sundararaman, and S. S. Palakurthy, "How much is my car worth? a methodology for predicting used cars' prices using random forest," in *Future of Information and Communication Conference*, pp. 413–422, Springer, 2018.

[23] C. Selvarathi, G. Bhava Dharani, and R. Pavithra, "Survey on pre-owned car price prediction using random forest algorithm," in *International Conference on Information and Communication Technology for Intelligent Systems*, pp. 177–189, Springer, 2023.

[24] D. A. Budiono, K. S. Utomo, K. J. Wibowo, and M. J. Wiradinata, "Used car price prediction model: a machine learning approach," *International Journal of Computer and Information System (IJCIS)*, vol. 5, no. 1, pp. 59–66, 2024.

[25] P. Bhatnagar, G. H. Lokesh, J. Shreyas, F. Flammini, and S. Gautam, "An analysis of car price prediction using machine learning," in *Proceedings of the 2024 9th International Conference on Machine Learning Technologies*, pp. 11–15, 2024.

[26] M. Gollapalli, T. A. Alqahtani, D. H. Alhamed, M. R. Alnassar, A. M. Alajmi, Y. H. Alali, M. M. Abdulqader, and A. Saadeldeen, "Intelligent modelling techniques for predicting used cars prices in saudi arabia.," *Mathematical Modelling of Engineering Problems*, vol. 10, no. 1, p. 139, 2023.

[27] T. Li, "Used car price prediction and feature importance analysis using xgboost and shap," *Highlights in Business, Economics and Management*, vol. 65, pp. 460–465, 2025.

[28] M. D. Ekstrand, A. Das, R. Burke, and F. Diaz, "Fairness in recommender systems," in *Recommender Systems Handbook* (F. Ricci, L. Rokach, and B. Shapira, eds.), pp. 679–707, New York, NY, USA: Springer, 2022.

[29] S. Zhang, L. Yao, A. Sun, and Y. Tay, "Deep learning based recommender system: A survey and new perspectives," *ACM computing surveys (CSUR)*, vol. 52, no. 1, pp. 1–38, 2019.

[30] K. U. K. Reddy, K. L. Sai, and R. N. Rao, "Machine learning-powered car recommendation system: A content-based and collaborative approach," in *International Conference on Smart Computing and Communication*, pp. 339–349, Springer, 2024.

[31] S. Raza, M. Rahman, S. Kamawal, A. Toroghi, A. Raval, F. Navah, and A. Kazemeini, "A comprehensive review of recommender systems: Transitioning from theory to practice," *Computer science review*, vol. 59, p. 100849, 2026.

[32] D. Kreuzberger, N. Kühl, and S. Hirschl, "Machine learning operations (mlops): Overview, definition, and architecture," *IEEE access*, vol. 11, pp. 31866–31879, 2023.

[33] A. Bodor, M. Hnida, and D. Najima, "From development to deployment: An approach to mlops monitoring for machine learning model operationalization," in *2023 14th International Conference on Intelligent Systems: Theories and Applications (SITA)*, pp. 1–7, IEEE, 2023.

[34] L. E. Lwakatare, A. Raj, I. Crnkovic, J. Bosch, and H. H. Olsson, "Large-scale machine learning systems in real-world industrial settings: A review of challenges and solutions," *Information and software technology*, vol. 127, p. 106368, 2020.

[35] M. Kunath and H. Winkler, "Integrating the digital twin of the manufacturing system into a decision support system for improving the order management process," *Procedia Cirp*, vol. 72, pp. 225–231, 2018.

[36] B. Korth, C. Schwede, and M. Zajac, "Simulation-ready digital twin for realtime management of logistics systems," in *2018 IEEE international conference on big data (big data)*, pp. 4194–4201, IEEE, 2018.

[37] S. Gautam, R. Noureddine, and W. D. Solvang, "Machine learning and iiot application for predictive maintenance," in *International Workshop of Advanced Manufacturing and Automation*, pp. 257–265, Springer, 2022.