IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940



International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 10, October 2026

VALUDRIVE - VEHICLE PRICE PREDICTION AND RECOMMENDATION SYSTEM USING MACHINE LEARNING

K Suraj[1], Shruthilayaa M[2], Vidya Sree[3], Dr. Beenarani Manoj[4]

Student, Department of Computer Science and Engineering, SRM Institute of Science and Technology, Vadapalani, Chennai, India[1-3]

Mentor, Department of Computer Science and Engineering, SRM Institute of Science and Technology, Vadapalani, Chennai, India[4]

Abstract: Used-car resale prices are veering according to vehicle attributes such as year, kilometres driven, fuel type, transmission, engine capacity and ownership history. This paper introduces ValuDrive, a machine-learning web application that assessments used-car selling prices and recommends comparable vehicles in one interface, to a important degree, in many respects. Linear Regression, Decision Tree Regressor and Random Forest Regressor were inspected employing an 80:20 training, testing split and the metrics Mean Absolute Error (MAE), Root Mean Squared Error (RMSE) and R^2. This model is using the Car details v3.csv dataset and applies data cleaning, feature preparation, numerical imputation, categorical imputation and One-Hot Encoding, in real-world scenarios. Random Forest hit the strongest findings among the found models, with an MAE of ₹80,195, an RMSE of ₹180,544 and an R^2 of 0.851. The application blends the picked prediction pipeline with a content-based recommendation engine that's combining weighted credit matching and cosine similarity. These results reflect the available dataset and test split. The anticipated value is an estimate rather than a live-market quote or professional valuation, at least in practice.

1. INTRODUCTION

1.1 Background and Motivation

The second-hand car market is currently buying with many choices, and even though, verifying separate vehicles isn't always easy, in the main. Cars from the same brand can, to some extent, differently price resting on their age, mileage, fuel type, transmission, ownership history, condition and market demand, in many respects. Buyers often are, after all, checking listings manually. So, sellers may find it difficult to decide on a competitive asking price. a data-driven estimate can probably support an initial decision. In turn, then again, it can't replace a physical inspection, at least for now. In one web application, these capabilities are being combined by ValuDrive. Patterns in historical listings can, in practice, be spotted by machine learning, and all at once, recommendation systems can help users discover vehicles that are normally matching their needs.

1.2 Problem Statement and Objectives

The central problem tackled by this project is the obstacle of estimating a employed car's resale price and spotting apt alternatives without manually checking many listings, in many respects. The aims are to prepare the Indian used-car dataset, compare Linear Regression, Decision Tree Regressor and Random Forest Regressor, appraise their performance using MAE, RMSE, and R^2, and formulate a recommendation component that ranks comparable listings and alternative vehicles according to applicable attributes, in actual application.

Price prediction and vehicle discovery are being combined by the applied prototype in a Flask web application; it projections a vehicle's selling price from historical listing data, then introduces similar listings and alternative models employing vehicle attributes, in real-world scenarios; and thus, its advancement is the integration of a measured regression-model comparison with an attribute-based recommendation workflow and readable explanations for matches, in the main. The output is intended for initial comparison, not as a guarantee of a vehicle's actual sale price or mechanical condition.

The suggested framework is emphasizing on used-car listings available in the picked dataset, and is intended to support initial price comparisons rather than officially value or find out the mechanical condition of a car. And yet, the actual usefulness and performance of the system can only be corroborated after code and testing. The principal insight of the project is the combination of price prediction and preference-based vehicle recommendations in one application.

 This work is licensed under a Creative Commons Attribution 4.0 International License

2. RELATED WORK

2.1 Used-Car Price Prediction

Used-car price prediction is routinely treated as a supervised machine-learning issue, where vehicle details are used to estimate resale prices, at least empirically. Lessmann and Voß analyzed car resale price forecasting and investigated how regression methods, private information and differences between vehicles alter prediction accuracy [1], at least for now; and thus, their work is relevant to this project, because used cars can (at least in theory) differ considerably in age and model, alongside condition, especially for price. In a bit, a model that's running well for one category of vehicles may (in actual operation) not yield equally exact predictions for another, and a study published in Information checked out distinct machine-learning schemes for anticipating pre-owned car prices in Bangladesh. The researchers found the models using R^2, RMSE and MAE [2]; in fact, even as the study concentrates on a diverse market from the Indian market considered in this project, it's showing why checking many models and evaluation metrics is essential when formulating a price prediction system.

2.2 Regression and Recommendation Methods

Linear Regression is offering a baseline by estimating price from relationships between vehicle features and the target value, all things considered; indeed, non-linear relationships can be approximated by a Decision Tree Regressor but may overfit if its architecture is too complicated, on the whole. A Random Forest Regressor combines several decision trees and can raise generalisation; in turn, in this job, these three models were checked employing held-out test data, and ValuDrive is supporting similar listings for the same brand and model, as well as alternatives from different models. Using their credits, content-based recommendation systems propose items; indeed, the recommendation process is using weighted content matching and cosine similarity to rank candidates according to vehicle attributes and user requirements.

2.3 Research Gap

The work tackles a concrete integration gap: many price-prediction demonstrations are emphasizing on a individual estimated value, broadly speaking, and ValuDrive brings price estimation, similar-listing discovery and alternative-vehicle recommendations into one application; in turn, on a held-out test split, its check is comparing three regression models, in the main. Besides, the recommendation score combines weighted content matching with cosine similarity; the score is a relative match indicator, not a probability of suitability, at least in principle.

3. METHODOLOGY

3.1 System Workflow and Dataset

The used process consists of dataset loading, data preprocessing, feature preparation, regression-model training and check, and vehicle recommendation, in many respects; the target variable is selling_price. The project uses the Car details v3.csv Indian used-car dataset, so the data is divided into 80% training data and 20% testing data, on the whole; in fact, model performance is reported on the test split employing MAE, RMSE and R^2.

3.2 Data Preprocessing and Features

The preprocessing pipeline removes duplicate entries, handles missing values and prepares the vehicle features for modelling, at least in principle; categorical inputs are reshaped with One-Hot Encoding. Numerical attributes are performed employing median imputation, so categorical traits use most-frequent imputation, and brand and model information are pulled from the vehicle name, broadly speaking. As, well, quantitative fields such as mileage and engine capacity are cleaned; indeed, the preprocessing pipeline is fitted as part of model training so that the same transformations can (at least in most settings) be applied consistently (specifically in this setting) when the application receives new inputs.

3.3 Prediction Models

Three regression algorithms were checked. Linear Regression was a baseline model, Decision Tree Regressor was employed to represent non-linear relationships, and Random Forest Regressor combined multiple decision trees to raise generalisation, all things considered; and thus, the test-set comparison showed Random Forest ran best, across the reported metrics, so it's picked as the prediction model used by the application, at least empirically.

3.4 Training and Evaluation

The dataset was, after all, divided into 80% training and 20% testing data. The models were verified using Mean Absolute Error (MAE), Root Mean Squared Error (RMSE) and R^2, in real-world scenarios. Together, a higher R^2 is showing that the model explains more change in the target values, in actual application. And thus, lower MAE and

 This work is licensed under a Creative Commons Attribution 4.0 International License

RMSE propose smaller prediction errors. In turn, the selected Random Forest pipeline is saved and blended with the Flask application, in many respects.

Mean Absolute Error(MAE): MAE = (1/n) Σ $|y_i-\hat{y}_i|$

Root Mean Squared Error(RMSE): = $\sqrt{[(1/n)\Sigma (y_i-\hat{y}_i)^2]}$

R^2 Score, specifically R2 = 1 - $[\Sigma (y_i-\hat{y}_i)^2 / \Sigma (y_i-\hat{y}_i)^2]$

Here, y_i represents the actual recorded price, $\hat{y}_i$ represents the projected price, $\bar{y}$ is the average recorded price and n is the number of entries in the test set, on the whole, and r^2 shouldn't be treated as percentage accuracy. MAE and RMSE are said clearly in the same currency unit as the dataset, all things considered.

3.5 Content-Based Recommendation

Two monitors, like similar listings that are emphasizing on the same brand and model, and alternative vehicles that are considering different models are being offered by the recommendation component, broadly speaking. The engine combines weighted content matching (85%) with cosine similarity (15%) to generate a hybrid match score and rank the candidates; this score is a relative matching score out of 100, not a probability that the vehicle is fitting, in real-world scenarios; indeed, candidate alternatives can, to some extent, be filtered by credits such as year, fuel type, transmission, price range and seating capacity.

Cosine similarity, namely $\cos(\theta) = (A \cdot B) / (\|A\| \|B\|)$

The hybrid score is employed to rank candidate vehicles and relevantly match; in turn, readable attribute-based reasons behind recommendations can be shown by the application, on the whole; in fact, the score weights are design choices and should be further found with ranking metrics and user feedback, in real-world scenarios. A high match score suggests similarity, under the framework's picked attributes; it doesn't guarantee vehicle quality or suitability.

3.6 System Implementation

Python, Pandas and NumPy for data handling, and scikit-learn for preprocessing and regression are being used by the code. The trained model pipeline and model metadata are saved as Pickle-based `.pkl` files. Flask is connecting the saved prediction pipeline and recommendation engine to the web interface, which is built using HTML, the standard web markup language, the standard web markup language, CSS, the web styling language, the web styling language, and JavaScript. Vehicle details can be entered by users, receive an estimated selling price, and check out comparable listings or alternative vehicles.

4. EXPERIMENTAL SETUP AND RESULTS

4.1 Dataset and Experimental Setup

The project has been used as an academic prototype named ValuDrive; the available README reports a model comparison on an 80:20 train, test split employing the Car details v3.csv dataset, broadly speaking. An MAE of ₹80,195, an RMSE of ₹180,544 and an R^2 of 0.851 was hit by Random Forest Regressor, in actual application; Decision Tree Regressor and Linear Regression results are also reported below. The check represents performance on the available dataset and test split; it shouldn't be took as a guarantee of live-market accuracy.

4.2 Evaluation Metrics and Model Selection

The check employed the same held-out test split for the comparison of the three regression algorithms, broadly speaking. The README is identifying Random Forest as the picked model, so what's more validation could encompass cross-validation, hyperparameter tuning and testing on more recent listings, at least in principle. RMSE is giving greater weight to larger deviations. MAE measures the average absolute difference between projected and actual selling prices. R^2 measures the proportion of change in selling prices explained by the model relative to a mean-prediction baseline.

4.3 Model Performance Results

Table 1 introduces the reported test-set findings. Random Forest Regressor created the lowest MAE and RMSE and the highest R^2 among the three evaluated models, which is making it the chosen model for the current prototype, at least in principle; the R^2 value of 0.851 suggests the model explains about 85.1% of the fluctuation in selling prices in this test data; it doesn't mean that the model has 85.1% prediction accuracy, to a significant degree.

The reported metrics are particular to the dataset and train, test split employed; so, findings should be re-evaluated when newer data or extra attributes become available, at least in practice, and large deviations may (to a meaningful extent) still emerge for vehicles with uncommon credits, unusual conditions or prices that differ from historical listing patterns, to a significant degree.

Table 1. Model evaluation results reported for the held-out test split.

Regression model	MAE	RMSE	R^2
Random Forest Regressor	₹80,195	₹180,544	0.851
Decision Tree Regressor	₹102,677	₹249,537	0.716
Linear Regression	₹177,094	₹329,462	0.505

5. DISCUSSION

The findings show that Random Forest Regressor is the strongest of the three evaluated models for the current dataset, with lower prediction errors and a higher R^2 than Decision Tree Regressor and Linear Regression, all things considered. The combined recommendation engine adds a discovery function beyond price estimation; it first filters candidates and then ranks vehicle matches employing weighted content matching and cosine similarity, at least for now; in fact, the prototype's model selection is being supported by the reported results. At the same time, further testing is being needed by broader generalisation and recommendation quality.

5.1 Shortcomings

The model importantly is limiting. Asking prices may (in practical terms) differ from the actual prices at which vehicles are sold; information about vehicle condition, maintenance history and other significant details may, at least in theory, also be missing from the dataset, in many respects. Used-car prices can change across the regions and over time; a model trained on older data may therefore become less solid as market conditions change, in real-world scenarios.

5.2 Future Work

Another shortcoming is that vehicles with analogous attributes aren't necessarily in comparable mechanical condition, to a significant degree; indeed, future enhancements could, after all, encompass solid information about vehicle condition and location, testing extra machine-learning models, raising the explanation of model predictions and collecting user feedback to gauge the quality of recommendations, at least empirically.

6. CONCLUSION

This paper offered ValuDrive, an academic machine-learning web application that's combining used-car price prediction and vehicle recommendations, all things considered. A hybrid recommendation engine combines weighted content matching (85%) with cosine similarity (15%) to rank similar listings and alternative vehicles, at least for now. The best reported performance, with an MAE of ₹80,195, an RMSE of ₹180,544 and an R^2 of 0.851 was hit by Random Forest Regressor, and was picked for integration into the Flask application. Employing the Car details v3.csv dataset and an 80:20 train, test split, three regression models were checked using MAE, RMSE and R^2. Preparatory comparisons can be supported by the framework; so, still, its predictions are based on historical dataset patterns and aren't live-market valuations. Future work should encompass cross-validation, hyperparameter tuning, check of recommendation rankings, and the addition of vehicle-condition, location and more recent market data.

REFERENCES

[1]. S. Lessmann and S. Voß, “Car resale price forecasting: The impact of regression method, private information and heterogeneity on forecast accuracy,” 2017. And full publication details and DOI should be checked before submission.

[2]. “Application of Machine Learning Techniques to Predict the Price of Pre-Owned Cars in Bangladesh,” Information, vol. 12, no. 12, article 514, 2021, broadly speaking. Available at: https://www.mdpi.com/2078-2489/12/12/514

[3]. Available at: https://www.kaggle.com/code/mennahitham/us ed-car-price-prediction/notebook. “Used Car Price Prediction,” Kaggle notebook mentioned in the project presentation; so, the source and dataset licence should be checked before use, all things considered.

 This work is licensed under a Creative Commons Attribution 4.0 International License

[4]. scikit-learn developers, "Metrics and scoring: quantifying the quality of predictions." Available at: https://scikit-learn.org/stable/modules/model_e valuation.html

[5]. G. Milunovich, L. Wu and Y. Zhao, "Forecasting the Prices of Used Cars: A Comparative Analysis of Supervised Learning Algorithms," SSRN, 2023. Available at: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4334750