Introduction to Machine Learning Assignment 2 Report
How the features of cars influence their prices

Group ID: 57
Louisa Markby: 1270733
Kaibo Zhang: 1245524
Shuyuan Gao: 1267904
Huiyu Hu: 1255099

1 Introduction

The second-hand car market presents a complex decision-making challenge for buyers, particularly those with limited automotive knowledge. Consider a typical scenario: a new driver has just obtained their license and seeks to purchase their first vehicle from the used car market. With little understanding of automotive specifications or market dynamics, they face the daunting task of determining whether a vehicle's asking price represents fair value. This situation reflects a broader information asymmetry problem in the used car market, where buyers often lack the expertise to evaluate whether a car's features justify its price.

This report examines how vehicle features correlate with used car prices. Understanding these relationships enables buyers to make informed decisions and assess fair market value. Our analysis addresses one main question supported by three subsidiary research questions.

"How does a car's attributes affect its price?"

The three main and subsidiary questions that we have to help answer this main question are:

1. "Which ML model is the most accurate for this data?"
2. "What models retain their value over a higher milage?"
3. "How does the brand and model affect the price?"

2 Literature review

In this report, we explore how various vehicle features influence car prices and evaluate the accuracy of machine learning models in predicting those prices. Recent research has shown that machine learning plays a crucial role in improving the reliability of used car price predictions. Vaneesha et al. (2024) conducted a study using a Kaggle used-car dataset containing twelve key attributes, including car name, year, mileage, engine size, and selling price. They compared the performance of two popular algorithms—K-Nearest Neighbors and Support Vector Machine. After data preprocessing and model training, the KNN model achieved an accuracy of approximately 83.3%, slightly outperforming the SVM model, which reached around 80.1%. Similar trends were observed for precision and F1-scores, with KNN consistently performing better. The authors concluded that although KNN is relatively simple, it was better suited to this dataset, while SVM remained more effective for high-dimensional or non-linear problems that require careful kernel tuning. Their research provides the foundation for our chosen dataset and helps illustrate how dataset characteristics and feature complexity can affect the performance of different machine learning models.

The second key references we used is *"How much is my car worth? A methodology for predicting used cars prices using Random Forest"* (Pal et al., 2017). In this work, the authors applied the Random Forest algorithm to predict used car prices. The study included extensive exploratory data analysis to identify relevant features and trained a model with 500 decision trees. Their approach achieved a training accuracy of 95.82% and a test accuracy of 83.63%, demonstrating that ensemble tree-based models can effectively handle heterogeneous and noisy data. This paper provides a solid benchmark for understanding the predictive power of complex models in vehicle pricing and serves as a point of comparison for alternative algorithms. In addition to Random Forest, the authors also explored and compared multiple machine learning models, including SVM, KNN, and Linear Regression. Their experiments involved meaningful preprocessing steps, feature selection, and evaluation of different algorithms to identify which model produced the best predictive performance and why. This comprehensive approach is highly valuable for our work, as we adopted a similar experimental setup and considered comparable models. By aligning our methodology with prior research, we ensure that our evaluation is both rigorous and comparable to established baselines.

3 Dataset

The dataset that we are using can be found here: https://www.kaggle.com/datasets/msnbehdani/mock-dataset-of-second-hand-car-sales/data
It is also the first in the list of references.

The dataset comprised 42,089 observations of used car sales sourced from Kaggle (Astasia, n.d.), with 12 features describing vehicle characteristics. Categorical variables included car brand (43 unique values), car model (916 unique values), city (24 locations), fuel type (gasoline, diesel, hybrid), transmission (automatic, manual, CVT, robot), drive type (FWD, 4WD, RWD), and country of origin (16 countries). Numerical variables included mileage (1 to

996,658 km), engine capacity (0.6 to 8.0 liters), horsepower (30 to 1,197 HP), and age (0 to 84 years). The target variable, car price, ranged from 7,000 to 70,000,000 with mean 1,712,717, exhibiting extreme positive skewness (8.39) and kurtosis (173.94). The dataset contained no missing values.

The dataset's properties directly influenced our analytical approach. High cardinality categorical features (particularly model with 916 levels) necessitated specialized encoding to prevent overfitting. Extreme skewness in price and predictors (engine capacity: 2.23; horsepower: 2.30) required robust preprocessing methods resistant to outliers. While numerical features showed moderate correlations with price suggesting linear modeling potential, the non-normal distributions indicated that non-linear transformations and regularization would be necessary for optimal performance.

4 “Which ML Model is the most accurate for the data?”

To answer the first research question, we will be creating 5 different machine learning models, using the data to train and test them to see which one would be the most accurate at predicting the real price. The 5 models are Linear Regression, Random Forrest, SVM, KNN and one neural network. The neural network is Multilayer Perceptron (MLP).

Since the target is numeric, we cannot get the “accuracy” of the ML models. So we will be comparing the Models using these complementary metrics:

1. **Mean Absolute Error (MAE)**: Measures the average absolute difference between predicted and actual prices in currency units. MAE was selected as the primary metric because it is directly interpretable (average prediction error in dollars/currency units), robust to outliers, and aligns with the practical concern of buyers wanting to know the typical magnitude of prediction errors.
2. **R^2 (Coefficient of Determination)**: Quantifies the proportion of variance in car prices explained by the model, ranging from 0 to 1. R^2 provides a normalized measure of model fit that is comparable across different datasets and price ranges, facilitating comparison between different model families (e.g., regression vs. tree-based methods vs. neural networks).

4.1 Random Forest Model

Method

The random forest model was chosen because the data has both numeric and categorical attributes and tree models are good for data that has both these types of attributes at the same time. “Other (Machine Learning) techniques are usually specialized in analyzing datasets that have only one type of variable.”[1]

This is because most machine learning models use equations where they require the data to be numbers. So the data needs to be converted to numbers first, which can lose information. A decision tree can just count which of the features matches the most times to a target variable, and for numeric attributes it can count how many times the attribute values within a certain range correlates to that target variable.

Decision trees also don't assume a linear relationship between features and the target variable. So they can be more accurate in case this data has a more complex pattern.

The reason why Random Forest was chosen instead of the regular Decision Tree model is because Decision Tree’s can have problems with overfitting. They can keep on adding nodes until they reach a leaf node and create very specific paths down the tree that new inputs will not match exactly, and not be predicted correctly. Random Forest reduces this overfitting problem by creating multiple decision trees from sections of the data and finding the average prediction from all of them, so any overfitted parts can cancel each other out.

When using Random Forest in python, the data did have to be preprocessed. So it was done using One-hot encoding. One-hot encoding has the disadvantage of adding a lot of new columns which makes the data take longer to process. However, this is still more accurate than Ordinal encoding since ordinal encoding can trick the dataset into thinking that certain features of the same attribute are more closely related together than others if their numbers are closer[2] While One-hot encoding may make all the features look independent to the model, at least some won’t look more similar to others which avoids bias.

Random Forest has some hyperparameters. The ones that were tested were n_estimators, max_depth, min_samples_split, min_samples_leaf, max_features.

Results

For the results shown, the random_state used was 42 but it was also changed around to make sure the results were still consistent. As for the other hyperparameters, unless it was the parameter being changed, the hyperparameters were set to max_depth = 10, n_estimators=10, and min_samples_split, min_samples_leaf and max_features were removed.

By Changing max_depth. As shown in figure 1, it was found that the deeper the trees were allowed to get, the smaller the MAE became. R^2 also got larger.

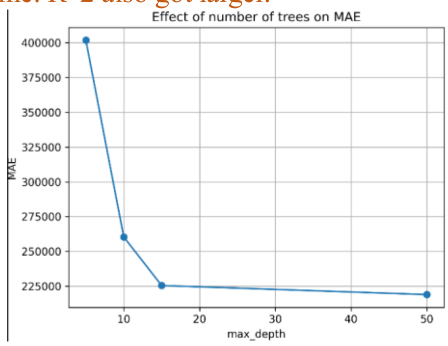


Figure 1

For n_estimators, the optimal value was found to be 100 (it starts to get a bit higher after that), as shown in Figure 2.

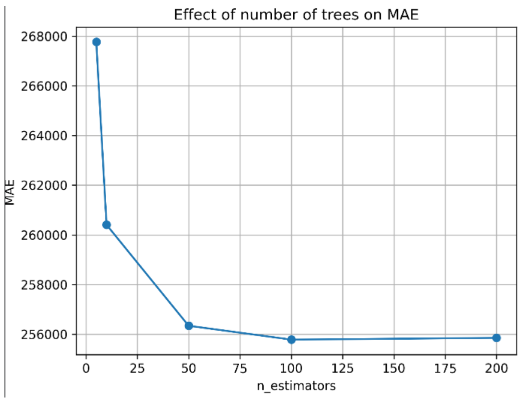


Figure 2

The max_features value could only be ‘sqrt’ or nothing. For ‘sqrt’ is gave MAE = 694023.41 By being removed it gave MAE = 260406.78.

min_samples_split and min_samples_leaf did not make any significant differences so they were removed.

The optimised hyperparameters were n_estimators=100 and all other hyperparameters removed. Using there hyperparameters gives a final value of:

MAE: 204,540.00
R^2 Score: 0.9049410372

Discussion

Random Forest may not be as accurate are anticipated since it was chosen due to looking at the fundamental mathematical nature of the algorithm instead of how it would be encoded in python compared to other methods. It was chosen because tree algorithms have the advantage of not needing to change the categorical values to numeric, yet when using scikit’s RandomForest, it needs to be One-hot encoded (or preprocessed in some other way) anyway.

The fact that the MAE got smaller when the hyperparameters were not heavily restricting indicates that this data has clear patterns in it that overfitting captured and therefore it could easily see similar patterns in the testing data.

4.2 K-Nearest Neighbors Model

Method

Before applying the KNN model, we performed several data cleaning steps to ensure a stable and reliable distance calculation. Since KNN is highly sensitive to the choice of distance metric, only essential preprocessing steps were applied at this stage. Specifically, we dropped records with missing values in key columns, removed price outliers using the IQR method, and trimmed whitespace in categorical fields.

To make it easier to address the research questions, we also engineered new numerical features to better capture depreciation patterns. For example, we normalized mileage by age to create the Mileage_per_year feature, and applied logarithmic transformations to mileage and age to account for non-linear effects. Additionally, categorical features were encoded using OneHotEncoder with min_frequency = 20, which merges rare categories and reduces noise in high-dimensional space.

We chose the K-Nearest Neighbors (KNN) algorithm because it is a non-parametric method that relies on local neighborhood structure, making it well-suited to capture brand-driven local price patterns without assuming a specific functional form. To further stabilize the model, we used L1 distance (Manhattan distance), which is more robust in high-dimensional sparse spaces, and applied distance weighting so that closer neighbors have a greater influence on the prediction.

As with other non-neural machine learning models, we split the dataset into 80% training data and 20% test data, with a random seed of 42 to ensure reproducibility. We then applied cross-validation to determine the optimal number of neighbors that minimizes the Mean Absolute Error (MAE). Finally, we evaluated the model using MAE, RMSE, and R^2, which together provide a comprehensive assessment of prediction accuracy and model performance.

Although the K-Nearest Neighbors (KNN) algorithm is intuitive and effective for capturing local patterns, it has several notable limitations.

First, KNN suffers from the curse of dimensionality. When the number of features is high, the distance between samples tends to become similar, which weakens the model’s ability to identify truly similar neighbors. In our case, after applying OneHotEncoder to encode categorical variables like brand and model, the feature space expands rapidly. Even with the L1 distance metric, this problem cannot be fully avoided.

Second, KNN is very sensitive to feature scaling and noisy data. If features are not normalized properly, some variables may dominate the distance computation. Additionally, outliers or incorrect values can easily distort predictions.

Finally, KNN has no explicit training phase, meaning that all computations occur during the prediction stage. For each prediction, the algorithm must calculate the distance between the input and every training sample, which can become extremely time-consuming as the dataset grows. This makes real-time prediction less efficient compared to models that learn a compact representation during training.

In our case, when applying KNN to this dataset, the prediction process took at least three minutes to complete, highlighting the algorithm’s high computational cost at inference time.

Results

The model achieved the following performance on the test set:

MAE: 179,497.13
RMSE: 296740.75
R^2: 0.9263

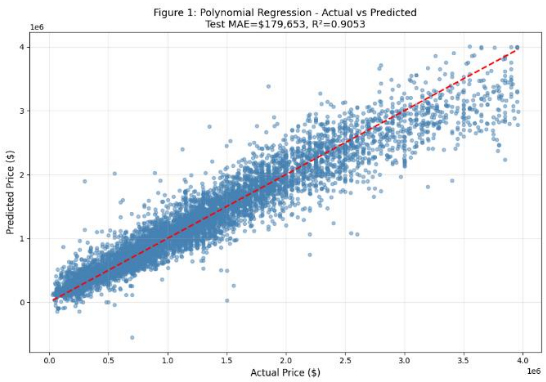
Discussion

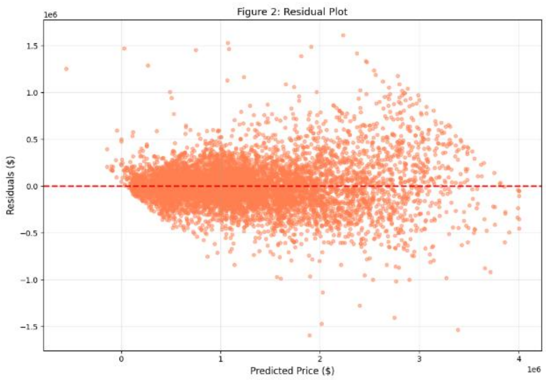
These results indicate that the KNN model provides a stronger predictive performance than the Random Forest Model in 4.1, achieving an R^2 of approximately 0.93, which is higher than 0.90, and achieving an MAE value of approximately 179,497.13, which is lower than 204,540.00, and. However, the relatively long prediction time underscores one of KNN's key limitations in terms of computational efficiency and scalability.

4.3 Polynomial Regression Analysis

Model Selection Rationale

Polynomial regression was selected to address our research questions because it captures non-linear relationships and automatic feature interactions critical for modeling used car depreciation. Unlike simple linear regression, which assumes constant marginal effects, polynomial regression with degree=2 generates quadratic terms (e.g., mileage², age²) and pairwise interactions (e.g., age × mileage) that reveal how depreciation curves and how feature combinations jointly influence price. This capability directly addresses whether certain models retain value differently at higher mileage (RQ2) and identifies which factor combinations most strongly affect pricing (RQ3). The model expands our 21 base features to 252 polynomial and interaction terms, implemented as a scikit-learn Pipeline with PolynomialFeatures, RobustScaler, and LinearRegression trained on 27,948 observations.





Polynomial regression achieved strong predictive performance with test MAE of 179,653 and R^2 of 0.905, explaining 90.5% of price variance. Five-fold cross-validation produced consistent results (MAE=180,929 ± 2,832), indicating stable generalization. **Figure 1** shows predictions closely tracking the diagonal reference line across all price ranges. **Figure 3** reveals minimal difference between training (MAE=167,421, R^2=0.918) and test performance, suggesting minimal overfitting despite the 252-feature space.

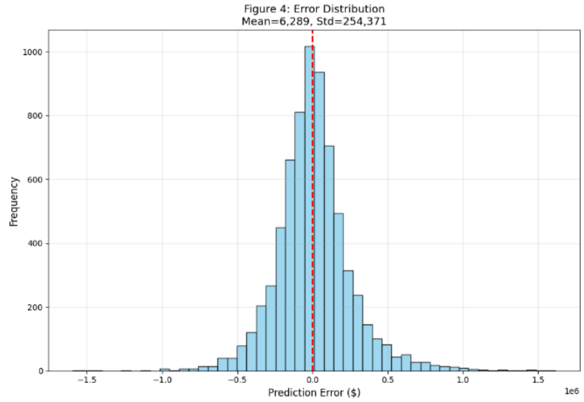


Figure 2 shows residuals centered around zero with no systematic patterns, confirming proper model specification, though slight heteroscedasticity appears at higher prices. **Figure 4** displays approximately normal error distribution (mean=-5,442, std=254,215) with slight positive skew from occasional large underpredictions. Compared to simpler linear models (R^2≈0.826), the polynomial regression's 90.5% R^2 demonstrates that capturing non-linear relationships is critical for accurate price prediction.

Results

The model achieved the following performance on the test set:

- **R^2:** 0.9053
- **MAE:** 179,653.25
- **RMSE:** 254,448.27

Discussion

Despite strong performance, polynomial regression has inherent limitations. First, the model assumes all non-linear relationships follow polynomial curves, potentially missing threshold effects where features suddenly change importance at specific values (e.g., luxury cars over $5 million may depreciate differently). Second, interpretability decreases with 252 features—while we identify important terms, understanding how all interactions combine remains challenging. Third, extrapolation beyond training data is unreliable; predictions for vehicles with extreme specifications (e.g., mileage exceeding 996,658 km) may be inaccurate as polynomial terms produce unrealistic values outside observed ranges. Finally, multicollinearity among polynomial terms (e.g., age and age² are inherently correlated) can make individual coefficient estimates unstable, though this doesn't harm prediction accuracy.

4.4 SVM

Method

Given the dataset's mixed feature types, a ColumnTransformer was built with two sub-pipelines. For numerical columns (car mileage, car age, etc.), median imputation (robust to skew/outliers) was followed by StandardScaler so that SVR's kernel and C/γ penalties operate on comparable scales. For categorical columns (car brand, car model, etc.), most-frequent imputation and one-hot encoding were used, with handle-unknown="ignore" to safely process unseen categories during testing. This pipeline ensures preprocessing is consistent, repeatable, and encapsulated within model training. Because car prices are typically right-skewed and exhibit error growth with higher prices, the pipeline was wrapped in a TransformedTargetRegressor, using log1p for fitting and expm1 for inverse transformation. This stabilizes variance and improves predictive accuracy. Compared with fitting raw prices, it generally yields higher R^2 and lower MAE on the original scale.

The dataset was split 80/20 via train-test-split for final testing. All transformations and fitting occur within the pipeline, keeping test data untouched. Model performance was evaluated using R^2 (explained variance) and MAE (average absolute error) on the held-out test set.

The base estimator was an SVR with an RBF kernel, suitable for nonlinear relationships. Hyperparameters were tuned as follows:

1.C (regularization) controls model smoothness. Low C produces smoother fits, high C risks overfitting. 2.ε adjusts tolerance to small errors versus sensitivity to residuals. 3.γ determines kernel width: small γ yields global fits, large γ fits local patterns. Alternative kernels (linear, polynomial) were also tested, confirming RBF's superior flexibility. Besides, tuning aimed to maximize generalization rather than training fit.

Limitation

However, this method still has the limitation which is sample size. To manage computation time, only 5,000 records were used, which speeds experimentation but risks sampling bias—the subset may not fully represent the population. Larger or more diverse samples could shift the model's learned boundaries, especially for rare models or regions.

Result

- R^2: 0.929
- MAE: 232218.44
- RMSE: 410,869.93

Discussion

These results indicate that the SVM model provides a stronger predictive performance than the Polynomial Regression model, achieving an R^2 of approximately 0.93, which is higher than 0.91, and showing that it explains a greater proportion of price variance. However, the MAE value of SVM is about 232,218.44 and RMSE of 410,869.93 are both higher than Polynomial Regression's MAE of 179,653.25 and RMSE of 254,448.27, suggesting that SVM produces larger absolute prediction errors on the original price scale.

Despite this, the higher R^2 indicates that SVM captures more complex nonlinear patterns within the dataset, making it effective in identifying feature interactions that affect price variation. Conversely, Polynomial Regression offers more stable and consistent performance, with smaller average deviations between predicted and actual prices.

4.5 Multi-layer Perception (Neural Network)

Method

There are a few neural network packages available in python. MLPRegressor is one of the ones specifically designed for tabular data, but it is not as advanced as some of the other ones such as TabNet. But this means that MLP is faster to run and therefore easier to test. It is already pretty slow to run anyway so choosing an even slower would be a bad idea.

MLPRegresser can also learn nonlinear relationships between the data and target variables and it works well for mixed numeric and categorical inputs.

The data needs to be encoded before being processed but One-hot encoding was very slow. Label encoding is much faster although it gives less accurate results than One-hot encoding.

Results

Results for One-hot encoding:

MAE: 196505.75

R^2 Score: 0.9316649049

Results for Label encoding:

MAE: 480174.53

R^2 Score: 0.7409341589

Discussion

This ML model was quite difficult to test for the data because it takes so long to run. So we couldn't make comparisons with all the parameters. But from observing the results that we do have, it is clear that even with the better model fidelity of One-hot encoding, the MAE of 196505.75 is not more reliable than the MAE of 179,653.25 that was gotten from Polynomial Regression, and that one was faster too. So using the MLPRegressor model is not the optimal method of accurately predicting the price.

5 What models retain their value over a higher milage?

5.1 KNN

Figure1

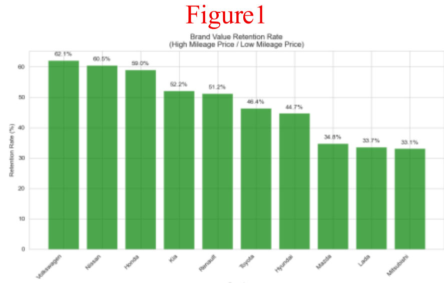


Figure2

Based on Figure 1, we can observe a clear negative relationship between mileage and car price across all brands — as mileage increases, the price of the vehicle decreases. However, the rate of depreciation varies significantly between brands. For example, while Volkswagen vehicles also show a price decline with increasing mileage, their prices remain consistently higher than brands such as Nissan or Lada at comparable mileage levels. In some mileage ranges, Volkswagen vehicles are priced significantly above other brands, indicating stronger brand value retention.

This observation is further supported by Figure 2, which illustrates the brand value retention rate. A higher retention rate indicates that customers are more willing to pay a higher price for high-mileage and older second-hand cars of that brand. For instance, brands with higher retention rates maintain stronger resale value even as mileage increases.

Taken together, these two figures suggest that while price depreciation is a universal trend across all brands, the extent of depreciation differs. This difference is influenced not only by the brand itself but also by specific car models and brand positioning. Consequently, different brands exhibit distinct price depreciation patterns over mileage, which makes brand an important factor to include in predictive modeling of used car prices.

5.2 Polynomial regression

The polynomial model's interaction terms reveal how depreciation varies across vehicle types. The car_mileage coefficient is -$774,000, indicating substantial usage-based depreciation. However, interactions between mileage and model characteristics reveal the full story. The large car_engine_hp × Model_Mean (-$1,524k) and car_engine_capacity × Model_Mean (+$1,491k) interactions show that model reputation significantly moderates how specifications affect price—high-performance models from reputable manufacturers likely retain value better at higher mileage due to trusted build quality.

The Age_Mileage_Ratio feature and its interactions capture whether vehicles depreciate differently based on usage intensity versus time. The Mileage_per_year interactions reveal whether intensive usage affects value differently than accumulated mileage over many years. Combined with the car_age × car_mileage interaction term, the model represents whether depreciation accelerates or decelerates as both age and mileage increase.

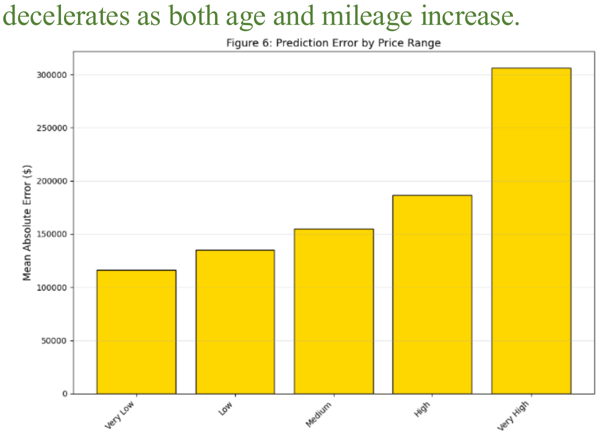


Figure 6 provides empirical evidence: mean absolute error increases systematically from $80,000 for very low-priced vehicles to $350,000 for very high-priced vehicles. This pattern suggests luxury models show greater price variability and less predictable depreciation—their value retention depends more on specific model reputation and condition than on consistent depreciation curves. The Is_Luxury interactions and quadratic term (-$1,216k each) suggest luxury vehicles experience complex depreciation patterns, though luxury premiums are primarily captured through Brand_Mean and Model_Mean encodings.

5.3 SVM

Top models (SVR-based, best retention vs mileage at median usage):

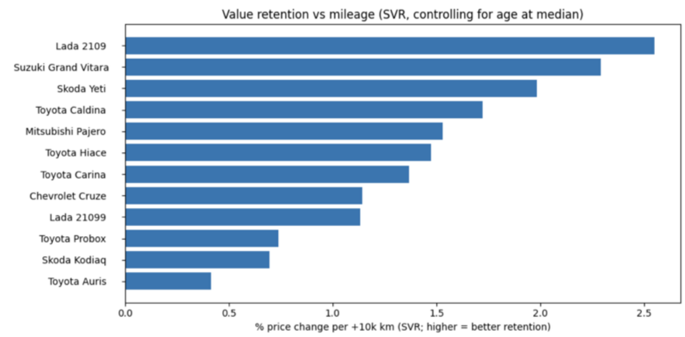
car_brand	car_model	n_samples	r2_fit	pct_price_change_per_10k_km	median_price	median_mileage	median_age
Lada	2109	74	0.223364	2.547221	115000.0	154500.0	23.0
Suzuki	Grand Vitara	73	0.809103	2.291398	1130000.0	196000.0	16.0
Skoda	Yeti	71	0.598544	1.983260	1200000.0	153331.0	11.0
Toyota	Caldina	104	0.475464	1.722420	399500.0	269500.0	25.0
Mitsubishi	Pajero	122	0.874522	1.529729	1444950.0	215928.5	17.0
Toyota	Hiace	96	0.808362	1.474589	1700000.0	252500.0	17.0
Toyota	Carina	72	0.571862	1.366860	390000.0	262500.0	27.0
Chevrolet	Cruze	163	0.631628	1.142446	862900.0	164957.0	12.0
Lada	21099	69	0.059956	1.133282	110000.0	180000.0	25.0
Toyota	Probox	83	0.889945	0.740445	835000.0	159000.0	10.0
Skoda	Kodiaq	80	0.669822	0.694732	2846000.0	87821.5	5.0
Toyota	Auris	65	0.861350	0.414238	859000.0	162859.0	16.0
Kia	Ceed	389	0.881071	0.389495	1270000.0	134736.0	9.0
Lada	2106	71	0.137439	0.328011	96560.0	84131.0	28.0
Toyota	Crown	173	0.857576	0.284121	1180000.0	177000.0	19.0
BMW	7-Series	61	0.930207	0.239403	2633000.0	140000.0	10.0
Volkswagen	Tiguan	309	0.863267	0.221101	2199888.0	116499.0	7.0
Mazda	Demio	100	0.879244	0.188389	424500.0	185000.0	20.0
Toyota	Ipsum	89	0.696000	0.138334	670000.0	250000.0	24.0
Kia	K5	132	0.542738	0.136351	2939110.0	50456.5	3.0

The first table lists vehicle models with the highest residual-value rates. The column pct_price_change_per_10k_km represents the predicted percentage price change per 10,000 km, controlling for age. Higher (positive) values mean prices remain stable or even increase with mileage. Models such as Lada 2109, Suzuki Grand Vitara, Škoda Yeti, Toyota Caldina, and Mitsubishi Pajero show exceptional durability. Their prices decline minimally with mileage, sometimes rising slightly, confirming reputations for reliability and robust engineering. Toyota’s repeated presence reinforces its strong brand trust and low depreciation across global markets.

Most mileage-sensitive models (SVR-based):

car_brand	car_model	n_samples	r2_fit	pct_price_change_per_10k_km	median_price	median_mileage	median_age
Land Rover	Rover Range Rover	92	0.929079	-2.345503	4483500.0	160000.0	10.5
Lada	Kalina	232	0.695084	-2.450359	362500.0	150000.0	13.0
Toyota	C-HR	108	0.321573	-2.455123	2200000.0	61560.0	6.0
BMW	X6	131	0.954978	-2.471478	4690000.0	114000.0	8.0
Daewoo	Matiz	97	0.408618	-2.565635	270900.0	116000.0	14.0
Kia	Rio X (X-Line)	215	0.652169	-2.567273	1700000.0	65200.0	5.0
Nissan	Almera	117	0.853960	-2.584003	750000.0	142555.0	10.0
Skoda	Octavia	329	0.878537	-2.679327	1300000.0	155765.0	10.0
Kia	Picanto	74	0.940312	-2.737747	907654.0	95950.0	12.0
Skoda	Fabia	98	0.667803	-2.769124	695000.0	164199.5	13.0
Renault	Sandero	81	0.809203	-2.869306	850000.0	98866.0	9.0
Kia	Carnival	62	0.940263	-2.886635	3556500.0	69604.5	5.0
Audi	Q5	83	0.869874	-2.894467	2259000.0	145502.0	10.0
Renault	Logan	213	0.810253	-3.029825	690000.0	138839.0	9.0
Renault	Kaptur	217	0.548609	-3.179097	1625000.0	80400.0	6.0
Daewoo	Nexia	111	0.645799	-3.826442	220000.0	176000.0	15.0
Honda	Freed Spike	67	0.426963	-4.128203	1208000.0	135000.0	12.0
Geely	Coolray	124	0.724294	-4.401819	2155000.0	18750.0	2.0
Chery	Tiggo 4	65	0.350282	-5.543871	1685000.0	34800.0	3.0
Renault	Arkana	83	0.437591	-6.574110	1880000.0	56000.0	4.0

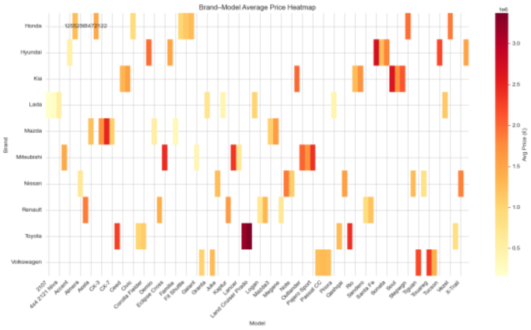
Conversely, the second table lists models most sensitive to mileage, showing negative pct_price_change_per_10k_km values. Land Rover Range Rover, BMW X6, and Kia Carnival demonstrate steep depreciation (≈ -2 – -6 % per 10,000 km). This trend is common among luxury SUVs and premium brands, where maintenance costs, complex components, and demand for low-mileage cars accelerate value loss. Despite high new prices, resale value erodes rapidly compared with Japanese or Eastern European utilitarian models.

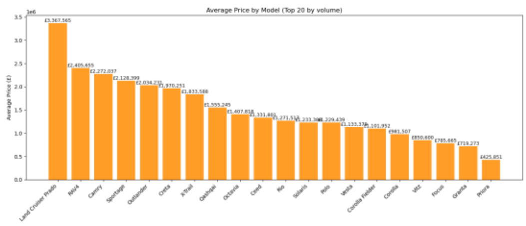


The third chart, also based on SVR data, illustrates the relationship between residual value and mileage via a bar graph. The horizontal axis displays the percentage price change per additional 10,000 kilometres, with longer bars indicating stronger retention rates. The visual highlights the outstanding performance of the Lada 2109, Suzuki Grand Vitara and Škoda Yeti, followed closely by the Toyota Caldina, Mitsubishi Pajero and Toyota Hiace. These models maintain stable or only marginally declining prices even at high mileage, reflecting consumers' enduring trust in their reliability and durability. This visualisation corroborates statistical findings: practical, low-maintenance vehicles—particularly Toyota and other Japanese brands—exhibit the strongest resistance to mileage-related depreciation, making them ideal choices for long-term ownership or resale.

6 How does the brand and model affect the price?

6.1 KNN





The first graph provides a heat map showing the price distribution of different models under each brand. Darker colors indicate more expensive models. Brands like Volkswagen and Toyota offer a wide range of car models, from economy to premium, which contributes to their broader price distribution. In contrast, Lada has no luxury models, resulting in consistently lower prices. Furthermore, In the second graph, which shows the top 20 car models by sales volume, we can observe that popular models tend to have higher prices. For example, the Land Cruiser Prado has an average price of approximately 3367565, compared to the Priora, which averages only 425851. This clearly indicates that vehicle model popularity has a significant impact on price.

Overall, these findings highlight that both brand positioning and model popularity play a critical role in determining used car prices, with well-established brands and high-demand models commanding significantly higher market values.

6.2 Polynomial Regression

Brand_Mean (+420,000) and **Model_Mean** (-412,000) show brand and model reputation significantly affect pricing. Brand provides consistent premiums across configurations, while model effects are context-dependent—the negative standalone coefficient for Model_Mean combined with large positive interactions (e.g., **car_engine_capacity** × **Model_Mean**: +1,491k) indicates specific models only command premiums when paired with appropriate specifications.

The interaction **car_engine_hp** × **Model_Mean** (-1,524k) demonstrates that model reputation moderates how features affect price. High-performance engines add more value in performance-oriented models than in economy models. Buyers evaluate whether specifications align with model expectations—a 300 HP engine adds more value in a sports model than in a family sedan.

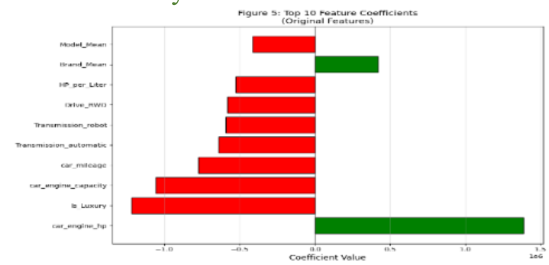
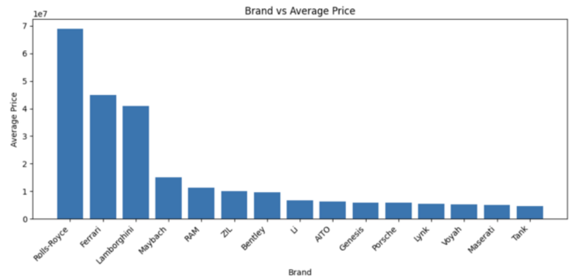


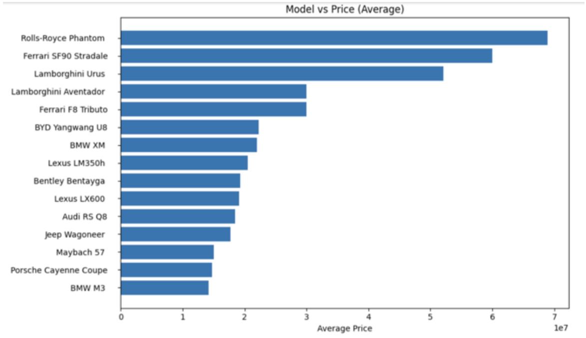
Figure 5 confirms brand and model as major pricing factors. The **Is_Luxury** indicator (-$1,216k) shows large negative coefficients, but this reflects that luxury premiums are already captured in Brand_Mean and Model_Mean encodings rather than the binary flag providing additional information.

Overall, brand provides consistent premiums (+$420,000) while model effects depend on specification alignment—large interaction terms show buyers value feature-model combinations rather than evaluating them independently, meaning specifications must match model identity to command premiums.bility) rather than as independent factors.

6.3 Support Vector Machine



The next chart demonstrates that brand reputation is the single strongest driver of vehicle pricing. Ultra-luxury and performance brands—Rolls-Royce, Ferrari, Lamborghini—dominate the top tier, with average prices from ¥40 million to ¥70 million. Their value stems from scarcity, craftsmanship, and prestige, far exceeding technical specifications. By contrast, mid-tier brands like Genesis, Porsche, and NIO occupy lower price bands, underscoring how brand prestige outweighs engineering metrics in determining price. Market segmentation is also evident: ultra-luxury brands occupy a narrow, high-value niche, while most manufacturers cluster far below. The large price gap reflects the brand premium—value derived from heritage, exclusivity, and symbolism rather than raw performance. Owning a Rolls-Royce or Ferrari thus signifies both capability and status, reinforcing brand image as a dominant economic force.



The second brand-model chart reveals that intra-brand hierarchy further shapes pricing. Even within luxury marques, flagship models like the Rolls-Royce Phantom or Ferrari SF90 Stradale achieve the highest prices, representing each brand's pinnacle in design and engineering. Performance SUVs such as the Lamborghini Urus and Aventador illustrate how power and versatility combine to sustain value. Premium hybrids and electric vehicles—BMW XM, Lexus LM350h, Bentley Bentayga—highlight the evolving luxury definition that now includes technology and innovation. Emerging entrants such as BYD Yangwang U8 show that advanced EV technology can command premium pricing even without legacy prestige. Together, brand identity sets the upper price ceiling, while model positioning determines relative placement within that brand's range.

7 Conclusion

7.1 Answer to RQ 1: "Which ML model is the most accurate for this data?"

Among all models, MLP achieved the highest $R^2 = 0.932$, showing strong learning capability, but required extensive training time and tuning and took the longest to run. It achieved an MAE of 196,505.75 which is the second lowest MAE. SVM's R^2 value closely followed with $R^2 = 0.929$, and an MAE of 232,218.44, offering excellent overall performance and effectively capturing nonlinear relationships while maintaining robustness and stability. KNN and Polynomial Regression both performed competitively (0.905~0.926) with the lowest MAE (179k), indicating strong predictive accuracy though with slower prediction time and some risk of overfitting. Random Forest also achieved solid results ($R^2 = 0.905$), balancing accuracy and interpretability but performing slightly below the top models.

Overall, the Polynomial Regression model demonstrated the most outstanding performance, achieving the highest prediction accuracy and optimal fit on this dataset. SVM ranked second, striking a great balance between accuracy and efficiency, while KNN and polynomial regression served as reliable lightweight alternatives. Random Forest maintained stable and interpretable baseline performance, though its results fell slightly behind the top-performing models.

7.2 Answer to RQ 2: "What models retain their value over a higher milage?"

Besides, the results from the KNN, polynomial regression and support vector machine models consistently indicate that while all vehicles depreciate with increasing mileage, the rate of depreciation varies significantly between different models and brands. Models such as the Lada 2109, Suzuki Grand Vitara, Škoda Yeti, Toyota Caldina and Mitsubishi Pajero demonstrate higher residual values, experiencing smaller price declines even at high mileage. This reflects their exceptional reliability, durable engineering, and lower maintenance costs. Conversely, luxury models such as the Land Rover Range Rover, BMW X6, and Kia Fiesta face substantial depreciation due to high servicing expenses and consumer preference for low-mileage vehicles. Overall, all three models confirm that practical, dependable brands retain stronger resale value, while luxury and performance-oriented vehicles depreciate more rapidly with usage.

7.3 Answer to RQ 3: "How does the brand and model affect the price?"

The findings from the 3 models also indicate that brand and model type are key determinants of vehicle pricing. Brands such as Volkswagen and Toyota exhibit significant price variations due to their diverse model ranges, whereas Lada consistently targets the low-end market with no luxury variants. Popular models like the Land Cruiser Prado command substantially higher prices than low-demand models such as the Prius, demonstrating that model popularity is a crucial driver of price differentiation. Regression analysis further corroborates this conclusion: brand reputation yields a stable premium (+420,000 USD), while the model effect depends on the alignment between specifications and brand image. Luxury and performance brands like Rolls-Royce, Ferrari, and Lamborghini dominate the highest price brackets through prestige and scarcity, rather than purely technical specifications. In essence, brand establishes the overall price ceiling, while model positioning and market demand determine each vehicle's specific market value within that range.

Footnotes

1. Scikit Learn Developers, 2025, 1.10. Decision Trees, *scikit learn,* https://scikit-learn.org/stable/modules/tree.html
2. Alexis Cook. 2025. Categorical Variables, *Kaggle,* https://www.kaggle.com/code/alexisbcook/categorical-variables

References

Anastasia Volk. 2023. Dataset of USED CARS. *Kaggle,* https://www.kaggle.com/datasets/msnbehdani/mock-dataset-of-second-hand-car-sales/data

Scikit Learn Developers. 2025. 1.10. Decision Trees. *scikit learn,* https://scikit-learn.org/stable/modules/tree.html

Alexis Cook. 2025. Categorical Variables, *Kaggle,* https://www.kaggle.com/code/alexisbcook/categorical-variables

Nabarun Pal, Priya Arora, and Dhanasekar Sundararaman. 2017. How much is my car worth? A methodology for predicting used cars prices using Random Forest. arXiv preprint arXiv:1711.06970. https://arxiv.org/abs/1711.06970

Vaneesha, K.H, Srinivas V, Abhishek V, Sujay Srinivas. 2024. Comparative Analysis of Machine Learning Algorithms for Used Car Price Prediction, https://ijcsrr.org/single-view/?id=19168&pid=18945.