

Mitigating Multicollinearity and Overfitting in Car Price Prediction: A comparative Analysis of Regularization and Ensemble Learning Techniques

Dedipy Chatteraj¹ and Sushovon Jana²

¹Eastern Institute for Integrated Learning in Management
DN 9, Salt Lake Sector 5, Kolkata, West Bengal, Pin-700091, INDIA
dedipyachatteraj777@gmail.com

²Department of Applied Statistics
School of Natural and Applied Science
Maulana Abul Kalam Azad University of Technology
Nadia, West Bengal, Pin-741249, INDIA
susostat@gmail.com

ABSTRACT

This study examines the effects of multicollinearity and overfitting on car price prediction using linear regression models and evaluates the effectiveness of regularization techniques in addressing these challenges. Multicollinearity, characterized by high correlations among predictor variables, inflates the variance of estimated regression coefficients and leads to unstable and unreliable predictions. Overfitting arises when a model captures random noise in the training data rather than the true underlying relationships, thereby reducing its generalization capability. To mitigate these issues, ridge regression and lasso regression are employed. Ridge regression introduces an L_2 penalty that shrinks coefficient estimates, reducing variance and improving model stability. Lasso regression incorporates an L_1 penalty, which not only regularizes the model but also performs variable selection by shrinking some coefficients exactly to zero. A comparative analysis conducted on a car price prediction dataset demonstrates that both regularization methods yield more robust, stable, and interpretable models compared to ordinary least squares regression. Furthermore, the study extends the analysis to ensemble learning approaches, including Random Forest and XGBoost, to evaluate their performance in the presence of multicollinearity and overfitting. The findings provide insights into the relative strengths of regularized and ensemble methods for improving predictive accuracy and model reliability.

Keywords: Linear regression, ridge, lasso, multi-collinearity, overfitting, Ensemble learning.

2000 Mathematics Subject Classification: 62J05, 62J07, 94A16.

1 Introduction

In the realm of predictive modeling, linear regression is a foundational technique used extensively due to its simplicity and interpretability. Despite these advantages, its performance

can be substantially compromised in the presence of multicollinearity and overfitting. Multicollinearity (Bager, Roman, Algedih and Mohammed, 2017) arises when two or more explanatory variables exhibit strong linear associations, leading to inflated variances of the estimated regression coefficients. As a consequence, coefficient estimates become unstable and highly sensitive to small perturbations in the data, thereby reducing the reliability and interpretability of the model. In such settings, disentangling the individual contribution of correlated predictors becomes challenging, often resulting in models that lack robustness and generalizability.

Overfitting (Hawkins, 2004), on the other hand, occurs when a model becomes overly complex and captures random fluctuations or noise present in the training dataset rather than the true underlying relationships. Although such a model may demonstrate excellent performance on training data, its predictive accuracy deteriorates significantly when applied to unseen observations. The issues of multicollinearity and overfitting are particularly prominent in datasets characterized by a large number of interrelated features, as commonly observed in car price prediction problems. In practical predictive applications, where accurate forecasting on new data is essential, these challenges hinder the model's capacity to generalize effectively and limit its real-world applicability.

To address these challenges, regularization techniques like ridge and lasso regression have been developed. Ridge regression introduces a penalty proportional to the sum of the squares of the coefficients (L2 penalty) (Goeman, Meijer and Chaturvedi, 2018). This penalty term discourages large coefficients by shrinking them towards zero, thus reducing the variance without significantly increasing bias. The result is a more stable model that is less sensitive to multicollinearity and performs better on new data. Ridge regression mitigates the problem of multicollinearity by constraining the coefficient estimates, thereby enhancing the model's stability and robustness.

Lasso regression (Least Absolute Shrinkage and Selection Operator) goes a step further by adding a penalty proportional to the absolute value of the coefficients (L1 penalty) (Vidaurre, Bielza and Larranaga, 2013). This not only shrinks the coefficients but also forces some of them to be exactly zero, effectively performing variable selection. This characteristic makes lasso regression particularly useful in high-dimensional datasets where it is beneficial to identify and retain only the most important predictors. By performing variable selection, lasso regression simplifies the model, making it more interpretable and reducing the risk of overfitting.

In this study, we perform a comparative analysis of linear regression, ridge regression, and lasso regression on a car price prediction dataset. By examining the performance metrics and the stability of the coefficient estimates, we aim to highlight the advantages of regularization techniques in improving model robustness and predictive accuracy. The findings from this study will provide insights into the practical application of ridge and lasso regression in scenarios characterized by multicollinearity and overfitting, offering a clearer understanding of how these methods enhance model performance. Through this comparative study, we aim to

demonstrate that ridge and lasso regression can significantly improve predictive performance and model interpretability in the presence of multicollinearity and overfitting, thus providing a more reliable tool for car price prediction.

Although our main target was to observe how does and to what extent does the problem of multicollinearity and overfitting are resolved by regularization methods at the end for cross verifying the possible outcomes and an extension of the study two ensemble methods have also been used. Random Forest (Grömping, 2009) from the Bagging part and Extream Gradient Boosting (XG Boost) (Wang, Wang, Zhang, He and Xu, 2022) from the Boosting part

2 Data Description

The dataset used in this study, titled *Car Price Prediction* from Creative Commons, contains comprehensive information about various car features and their corresponding prices. The dataset includes both **numerical** and **categorical** variables, which together facilitate a detailed understanding of the factors influencing car prices.

Categorical variables capture qualitative attributes such as the name of the car manufacturer, type of fuel, body style, aspiration, and engine configuration. These features often reflect brand reputation, design, and mechanical layout, which significantly impact consumer preferences and pricing.

Numerical variables, on the other hand, measure physical and performance-related attributes such as dimensions (e.g., wheelbase, carlength), engine capacity (e.g., enginesize, horsepower), and fuel efficiency (e.g., citympg, highwaympg). These factors are typically correlated with vehicle performance, production cost, and market value.

The dataset consists of 26 predictor variables and one dependent variable (*price*), offering a multidimensional view of the automobile pricing problem.

3 Methodology

This study aims to support an automotive company's entry into the U.S. market by identifying key variables that influence car pricing. The dataset, compiled from a wide range of vehicles across the American market, includes diverse features affecting vehicle valuation.

The core objectives are: (i) to identify significant predictors of car price, and (ii) to assess how well these variables explain the variation in price. The modeling outcomes will assist management in optimizing car design and pricing strategies to align with market dynamics.

To achieve this, a predictive modeling pipeline was developed using both basic and advanced regression techniques. Linear regression was first employed to assess baseline performance

and interpretability. However, challenges like multicollinearity and overfitting—common in high-dimensional datasets—prompted the application of regularization methods such as Ridge and Lasso regression.

These regularized models improve robustness by penalizing large coefficients, thereby reducing model variance. Ridge regression shrinks all coefficients, while Lasso additionally performs feature selection by setting some coefficients to zero. Finally, ensemble methods such as Random Forest (bagging) and XGBoost (boosting) were included to explore their performance under the same data conditions, particularly in mitigating overfitting and handling nonlinear relationships.

The following subsections outline the preprocessing steps applied to the dataset—including scaling, encoding, and outlier handling—followed by a discussion of each regression technique.

3.1 Scaling

Scaling is an essential preprocessing step that standardizes the range of independent variables to ensure that each feature contributes equally to model training. This becomes particularly important for algorithms like linear and regularized regression, which are sensitive to the scale of input variables.

In this study, scaling was performed after detecting and addressing outliers to prevent distortion in the scaled values. Without scaling, features with large magnitudes can disproportionately influence the model, potentially leading to biased coefficient estimates and slower convergence in optimization routines.

Two commonly used methods—Standardization (Z-score normalization) and Min-Max scaling—were considered based on the distribution and range of each variable. Standardization was preferred for normally distributed variables, while Min-Max scaling was used where bounded feature ranges were required. The overall aim was to improve training efficiency, stabilize numerical computations, and ensure meaningful interpretation of model coefficients.

3.2 Encoding

Encoding categorical variables into numerical format is crucial for incorporating them into regression-based models, which require numeric inputs. In this study, most categorical variables were nominal in nature and lacked any inherent ordering. As such, one-hot encoding was employed to convert each category into separate binary features, ensuring no artificial ordinal relationships were introduced. (Seger, 2018)

This transformation preserved the integrity of categorical information while making the dataset compatible with machine learning algorithms. Although one-hot encoding increases the di-

mensionality of the dataset, it provides a robust and interpretable representation that prevents misleading assumptions about category ranking—thereby improving both model performance and generalization.

3.3 Linear Regression

Multiple Linear Regression (MLR) (Uyanık and Güler, 2013) is a foundational statistical method used to model the relationship between a continuous dependent variable and multiple independent variables. It assumes linearity, independence of errors, homoscedasticity, normality of residuals, and minimal multicollinearity. The equation in Linear Regression is:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p + \varepsilon$$

The model estimates coefficients by minimizing the Residual Sum of Squares (RSS) using the Ordinary Least Squares (OLS) method (Kilmer and Rodríguez, 2017). The closed-form solution is obtained from the normal equations:

$$\hat{\beta} = (X^T X)^{-1} X^T y$$

where X is the design matrix, y is the response vector, and $\hat{\beta}$ is the vector of estimated coefficients.

While MLR offers interpretability and simplicity, it is highly sensitive to multicollinearity and overfitting, especially in datasets with correlated predictors. These limitations motivated the application of regularization techniques such as Ridge and Lasso regression, discussed in subsequent sections.

3.4 Ridge Regression

Ridge Regression (McDonald, 2009) also known as Tikhonov regularization, is a linear modeling technique designed to address overfitting and multicollinearity by imposing a penalty on the size of regression coefficients. Unlike standard linear regression, Ridge adds an L_2 penalty term to the loss function, which shrinks coefficients towards zero but does not eliminate them entirely. This results in a model that retains all predictors, albeit with reduced influence from less informative variables. By introducing a bias, Ridge reduces variance and improves generalization, especially when predictors are highly correlated. This makes it particularly suitable for scenarios where all variables are potentially relevant and where multicollinearity would otherwise lead to unstable and unreliable coefficient estimates.

The Ridge regression model modifies the classical linear regression by introducing a regularization term that penalizes large coefficients. The model equation remains:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p + \varepsilon$$

where y is the dependent variable (car price), β_0 is the intercept, $\beta_1, \beta_2, \dots, \beta_p$ are the regression coefficients, $x_1, x_2, \dots, x_p$ are the predictor variables, and ε is the error term.

The objective of Ridge regression is to minimize the penalized residual sum of squares (PRSS):

$$PRSS = \sum_{i=1}^n (y_i - \hat{y}_i)^2 + \lambda \sum_{j=1}^p \beta_j^2$$

Here, the first term measures the model's fit to the data, and the second term, scaled by the regularization parameter λ , penalizes large coefficients to reduce model complexity. This formulation encourages smaller weights, thereby enhancing model robustness and reducing sensitivity to multicollinearity, while retaining all predictors in the model.

To obtain the optimal coefficients in Ridge regression, the normal equations are modified to include the regularization term:

$$(X^T X + \lambda I) \beta = X^T y$$

where X is the matrix of input features (including the intercept column), y is the response vector, β is the coefficient vector, I is the identity matrix, and λ is the regularization parameter. The closed-form solution becomes:

$$\beta = (X^T X + \lambda I)^{-1} X^T y$$

This formulation adjusts the least squares estimation by adding λI , which stabilizes the inversion of $X^T X$, especially in the presence of multicollinearity. The penalty ensures that coefficient estimates are shrunk toward zero without being eliminated, improving generalization while maintaining all predictors in the model.

3.4.1 Effect of Regularization Parameter λ in Ridge Regression

The regularization parameter λ governs the strength of the penalty applied to the regression coefficients and plays a crucial role in determining the trade-off between bias and variance in Ridge regression. When $\lambda = 0$, the penalty term is effectively removed, and Ridge regression simplifies to ordinary least squares (OLS). In this scenario, the model aims solely to minimize the residual sum of squares (RSS), which may lead to overfitting, especially when multicollinearity or noise is present in the data, as it allows the coefficients to grow excessively large in an attempt to closely fit the training data.

In contrast, when λ takes on very large values, the penalty becomes dominant in the loss function. As a result, all coefficients are shrunk substantially toward zero, reducing the model's flexibility. While this helps prevent overfitting and improves stability in the presence of multicollinearity, it may lead to underfitting, as the model becomes too simplistic to capture important patterns in the data.

The most effective performance typically lies in choosing a moderate value of λ , where the model benefits from reduced variance without introducing significant bias. At this optimal

range, the regularization mildly shrinks the coefficients—enough to dampen the effects of multicollinearity and improve generalization, but not so much that it eliminates essential signal from the predictors. In practice, this optimal value of λ is determined through techniques such as cross-validation, where multiple values are tested to identify the one that minimizes validation error. Thus, Ridge regression achieves its strength not only through coefficient shrinkage but also through careful tuning of the regularization parameter.

3.5 Lasso Regression

Lasso regression (Least Absolute Shrinkage and Selection Operator) (Ranstam and Cook, 2018) addresses overfitting and multicollinearity by introducing an L_1 penalty term to the linear regression framework. This penalty not only shrinks coefficient magnitudes but also forces some of them to exactly zero, effectively performing feature selection. As a result, Lasso produces sparse and interpretable models, particularly useful in high-dimensional settings where many predictors may be irrelevant or redundant. By balancing bias and variance through regularization, Lasso simplifies the model, improves generalization to unseen data, and enhances predictive accuracy while automatically identifying the most influential variables.

Lasso regression modifies the standard linear regression model by incorporating an L_1 penalty term that promotes sparsity in the coefficient estimates. The model equation remains:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p + \varepsilon$$

where y denotes the dependent variable (car price), β_0 is the intercept, $\beta_1, \dots, \beta_p$ are the coefficients, $x_1, \dots, x_p$ are the independent variables, and ε is the error term.

The objective of Lasso regression is to minimize the penalized residual sum of squares (PRSS):

$$PRSS = \sum_{i=1}^n (y_i - \hat{y}_i)^2 + \lambda \sum_{j=1}^p |\beta_j|$$

In this formulation, the first term evaluates the model fit, while the second term, scaled by regularization parameter λ , penalizes the absolute values of the coefficients. Unlike Ridge regression, the L_1 penalty can shrink some coefficients entirely to zero, enabling automatic variable selection. This makes Lasso particularly useful when seeking a simpler, interpretable model without compromising predictive performance.

Unlike ridge regression, lasso does not have a closed-form solution (Roth, 2004). Instead, it is typically solved using optimization algorithms such as coordinate descent or least angle regression (LARS).

3.5.1 Coordinate Descent Algorithm

The coordinate descent algorithm for lasso regression iteratively updates each coefficient by minimizing the objective function while keeping the other coefficients fixed. The process can be described as follows:

1. **Initialize** all coefficients β_j to zero.

2. **Repeat** until convergence:

$$\beta_j \leftarrow \arg \min_{\beta_j} \left(\sum_{i=1}^n \left(y_i - \sum_{k \neq j} x_{ik} \beta_k - x_{ij} \beta_j \right)^2 + \lambda |\beta_j| \right) \quad (3.1)$$

for each predictor j (from 1 to p).

This iterative process continues until the changes in the coefficients are smaller than a pre-defined threshold, indicating convergence.

3.5.2 Effect of Regularization Parameter λ in Lasso Regression

In Lasso regression, the regularization parameter λ determines the strength of the penalty applied to the absolute values of the regression coefficients. This parameter plays a central role in controlling overfitting and guiding model simplicity through regularization. When $\lambda = 0$, Lasso regression reduces to ordinary least squares (OLS), where no penalty is applied. In this case, the model purely minimizes the residual sum of squares (RSS), which can lead to overfitting in the presence of noise or multicollinearity, as it permits all coefficients to grow unrestricted in their attempt to closely fit the training data.

As λ increases to very large values, the regularization term dominates the objective function. This leads to aggressive shrinking of the coefficients toward zero, and the model may become overly simplistic, capturing only the most dominant signals while ignoring subtler patterns. Such underfitting results in diminished predictive power and higher bias, particularly if relevant predictors are suppressed excessively.

For intermediate values of λ , Lasso achieves a valuable balance between bias and variance. It reduces the influence of less informative predictors while retaining those with stronger relationships to the response variable. A unique property of Lasso, unlike Ridge regression, is its ability to perform feature selection. As the value of λ increases, some coefficients are not just shrunk but driven exactly to zero. This leads to a sparse solution, effectively eliminating less relevant variables from the model and enhancing interpretability without manual variable selection.

The selection of an appropriate λ is typically performed using cross-validation, wherein multiple values are evaluated to identify the one that yields optimal performance on validation data. This tuning ensures that the Lasso model not only generalizes well but also remains parsimonious, robust, and easy to interpret.

3.6 Advanced Regression Models: Random Forest and XGBoost

Random Forest is an ensemble learning method based on the bagging (Bootstrap Aggregating) principle. It constructs a multitude of decision trees by training each on a random subset of the

data and averaging their outputs to produce the final prediction. During tree construction, a random subset of features is considered at each split, which helps to reduce correlation between trees and mitigate overfitting. This approach improves the model's generalization ability and is particularly effective for datasets with complex, nonlinear interactions and multicollinearity among features. Random Forest also performs internal feature selection and handles missing data relatively well. In this study, Random Forest achieved high predictive accuracy with strong performance on both training and testing sets, indicating its robustness and stability in the presence of noisy predictors and multicollinearity (Li, Zou, Bercibar, Nanini-Maury, Chan, Van den Bossche, Van Mierlo and Omar, 2018).

XGBoost (Extreme Gradient Boosting) is an advanced implementation of gradient boosting, which builds trees sequentially such that each new tree corrects the residual errors of the previous ones. Unlike Random Forest, which reduces variance, XGBoost focuses on reducing both bias and variance by combining boosting with regularization. It incorporates L_1 and L_2 penalties to avoid overfitting, making it more effective in scenarios with complex feature relationships. XGBoost also includes optimized techniques such as tree pruning, parallel processing, and efficient handling of missing values. Although more computationally intensive, it often outperforms other models in terms of accuracy. In this study, XGBoost demonstrated excellent training performance and strong generalization on test data, with slightly higher execution time than Random Forest due to its iterative nature and fine-tuned regularization controls. Overall, both models proved to be powerful extensions to the linear regression framework, capable of capturing nonlinear patterns and enhancing predictive reliability (Avanijaa et al., 2021)

3.7 Model Evaluation

Once the coefficients are estimated, we evaluate the model using metrics such as:

- **R-squared (R^2):** Measures the proportion of variance in the dependent variable that is predictable from the independent variables.

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2},$$

where $\hat{y}_i$ are the predicted values, $\bar{y}$ is the mean of y_i .

- **Mean Squared Error (MSE):** Measures the average squared difference between the observed and predicted values.

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

- **Root Mean Square Error:** The square root of MSE, providing an error metric in the same units as the dependent variable.

$$RMSE = \sqrt{MSE}$$

4 Result Analysis

In this section we have discussed our findings and result. Previously it was discussed how the linear regression algorithm can be extended to ridge and lasso regression as per the problem arises in the real life scenario.

The dataset is split into training set and testing set before fitting the model. 70% of data kept into training set and remaining 30%, are used for testing the dataset. Stratified K-fold cross validation has been used to optimize the training on training set.

The accuracy matrices for the algorithms are discussed, through some measurements like R squared, Root Mean Square Error, Mean Square Error, Mean Absolute Error. The reliability and performance of the machine learning models are discussed using their respective results of the accuracy metrics.

4.1 Exploratory Analysis: Heatmap of Numerical Variables

The heatmap provides a correlation matrix that visualizes the strength and direction of linear relationships among numerical variables. Several strong positive correlations were observed, indicating that key automotive features tend to move together. In figure 4 notably, *wheelbase* and *carlength* exhibit a high correlation (0.87), as do *carlength* and *curbweight* (0.88), suggesting that longer vehicles generally weigh more. Similarly, *carwidth* is strongly correlated with *curbweight* (0.74), reinforcing the idea that broader cars tend to be heavier. In terms of performance features, *horsepower* and *enginesize* are highly correlated (0.75), indicating that more powerful engines tend to be larger in displacement. Furthermore, both *curbweight* and *enginesize* show strong positive correlations with *price* (0.84 and 0.85, respectively), highlighting their significant role in determining vehicle cost.

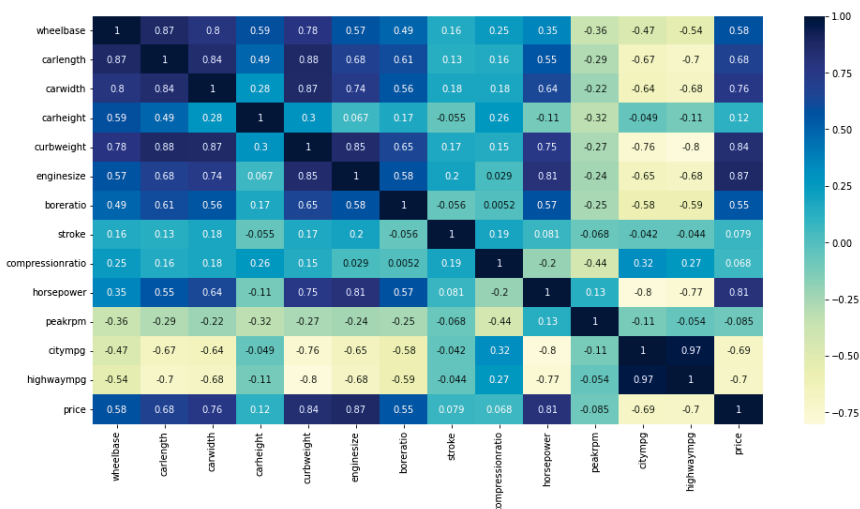


Figure 1: Heatmap

Negative correlations were also prominent, particularly between *citympg* and both *horsepower* (-0.80) and *curbweight* (-0.76), indicating that more powerful and heavier cars are generally less fuel-efficient in city conditions. Likewise, both *citympg* and *highwaympg* show moderate to strong negative correlations with *price* (-0.69 and -0.70, respectively), suggesting that more fuel-efficient vehicles tend to be priced lower. *Horsepower* also shows a strong inverse relationship with *highwaympg* (-0.77), and *curbweight* correlates negatively with highway efficiency (-0.68), reinforcing the trade-off between performance and fuel economy. Overall, these correlations offer valuable insights into the intertwined nature of physical dimensions, engine specifications, efficiency, and pricing within the automobile market.

4.2 Pairplot: Study of Numerical Variables

The pairplot serves as a valuable visual tool to explore bivariate relationships among key numerical features while simultaneously observing the distribution of each individual variable. It revealed strong linear trends between certain variables. For example, in figure 2 *enginesize*, *horsepower*, and *curbweight* all showed clear upward linear relationships with *price*, supporting the observation from the correlation heatmap that these features are directly associated with car cost. These trends confirm that vehicles with larger engines and higher power output tend to be priced higher in the market.

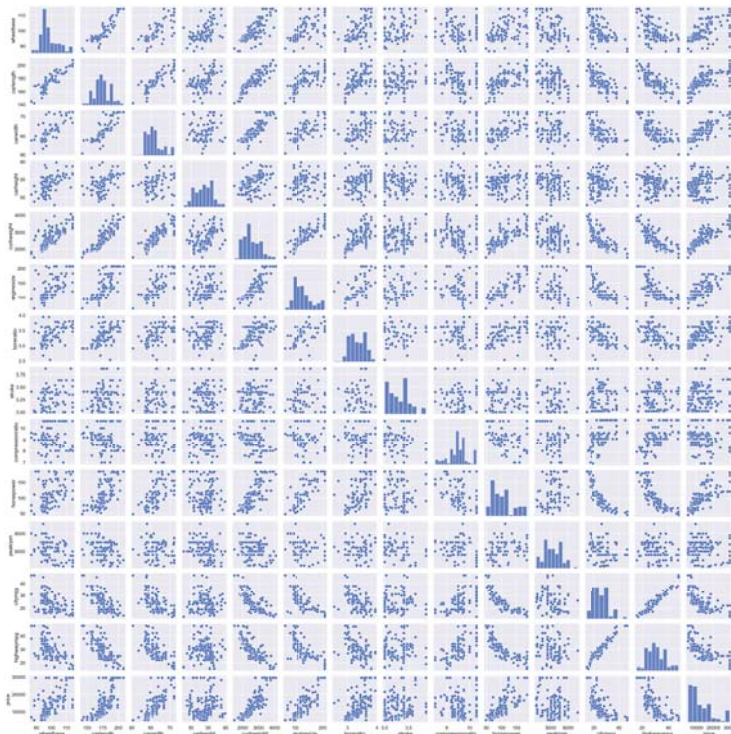


Figure 2: Pairplot

In contrast, features such as *citympg* and *highwaympg* displayed a negative relationship with

price, indicating that cars with better mileage generally belong to lower price segments. Additionally, the pairplot confirmed positive linear dependencies among variables like *wheelbase*, *carlength*, and *curbweight*, suggesting that as car dimensions increase, so does weight. While many relationships appeared linear and intuitive, the plot also helped in visually identifying potential outliers, particularly in variables such as *price*, *horsepower*, and *enginesize*, which may warrant further treatment. Overall, the pairplot not only reinforces the strength of previously observed correlations but also offers a more intuitive, visual understanding of the structural relationships within the dataset.

4.3 Heatmap and Crosstab: A study of Categorical variables

A crosstab (short for cross-tabulation) is a table that displays the frequency distribution of two or more categorical variables. It allows you to examine the relationship between these variables by showing how often each combination of categories occurs. A crosstab of drivewheel versus carbody has been plotted for the clear understanding of the customer choice and their preferences.

4.3.1 Crosstab of car manufacturer & risk-taking levels

Based on the crosstab between car manufacturer and risk-taking levels (Figure 3), cars have been categorized the cars into different risk categories (Much More Risky, Very Risky, Risky, Normal, Safety, Very Safety) and identified the manufacturers with the most cars in each category. The crosstab analysis between car manufacturers and risk-taking levels (as represented by the 'symboling' variable) reveals several meaningful patterns. Notably, manufacturers such as Audi and BMW are exclusively associated with moderate-risk vehicles, suggesting a brand identity that balances performance with safety. In contrast, companies like Honda, Isuzu, and Jaguar are entirely concentrated in the low-risk category, which may reflect a strategic focus on producing consumer-friendly, safety-oriented vehicles. Brands like Buick, Dodge, and Volvo display representation in both low and moderate-risk classes, indicating a diversified product lineup that targets varying customer risk preferences.

riskrating	0, Much more Risky Car	1, Very Risky Car	2, Risky Car	3, Normal Car	4, Safety Car	5, Very Safety Car	All
manufacturer							
alfa-romeo	2	0	1	0	0	0	3
audi	0	3	3	1	0	0	7
bmw	0	1	1	6	0	0	8
buick	1	0	1	2	4	0	8
chevrolet	0	1	1	1	0	0	3
dodge	1	0	7	0	1	0	9
honda	0	2	4	7	0	0	13
isuzu	0	1	1	2	0	0	4
jaguar	0	0	0	3	0	0	3
mazda	4	0	7	6	0	0	17
mercury	0	0	1	0	0	0	1
mitsubishi	5	3	4	0	1	0	13
nissan	2	1	10	5	0	0	18
pegasus	0	0	0	11	0	0	11
plymouth	1	0	5	0	1	0	7
porciche	4	0	1	0	0	0	5
renault	0	1	0	1	0	0	2
saab	3	3	0	0	0	0	6
subaru	0	3	0	9	0	0	12
toyota	2	6	7	10	7	0	32
volkswagen	2	7	0	3	0	0	12
volvo	0	0	0	0	8	3	11
All	27	32	54	67	22	3	205

Figure 3: Risk-taking v/s Manufacturer

On the other hand, certain manufacturers such as Porsche, Saab, and Peugeot have vehicle models across all three risk levels—low, moderate, and high. This suggests that these companies offer a broader range of vehicles that cater to a wide spectrum of consumers, from safety-conscious buyers to thrill-seekers. Interestingly, manufacturers like Chevrolet and Nissan are found only in the low and high-risk segments but have no models in the moderate category, pointing to a more polarized product distribution strategy. Mazda also spans all three risk levels, possibly reflecting a flexible market approach that accommodates different buyer personas.

Overall, this cross-tabulation offers critical insights into how risk propensity—indicated by the symboling score—is embedded within the strategic positioning of car brands. It highlights not only the variation in design and safety orientation across manufacturers but also provides valuable context for understanding consumer segmentation and branding in the automobile market.

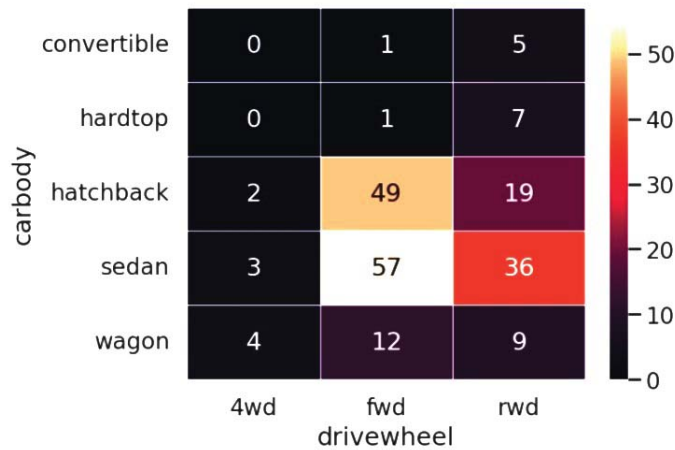


Figure 4: Heatmap: Carbody v/s Drivewheel

The heatmap examining the relationship between car body types and drivewheel configurations reveals important structural and performance-related design patterns across different automobile categories. Front-wheel drive (fwd) emerges as the most dominant configuration, particularly among sedans and hatchbacks. This is expected, as fwd systems are more fuel-efficient, cost-effective, and provide better traction under everyday driving conditions—making them ideal for mass-market, low-to-moderate risk vehicles. Sedans, in particular, show a strong preference for fwd systems, reinforcing their position as practical and safety-oriented cars. Hatchbacks also predominantly use fwd, consistent with their role as compact, urban-friendly vehicles.

Rear-wheel drive (rwd), in contrast, is mostly found in car types associated with performance and luxury, such as convertibles, hardtops, and premium sedans. These configurations of-

fer superior handling, power distribution, and driving dynamics, which explains their application in higher-end or sportier vehicle models. The presence of rwd is notably significant among convertibles, which are also commonly turbocharged and risk-classified as high symboling—suggesting a clear alignment of rwd systems with performance and aspirational design.

All-wheel drive (4wd) is relatively rare in the dataset but appears selectively across body types like wagons and occasionally in hatchbacks or sedans. The presence of 4wd in wagons likely reflects a utilitarian or off-road capability focus, offering better traction in varying terrain conditions. Interestingly, body types like hardtops and convertibles show no occurrences of 4wd, likely due to design constraints and the sport/luxury orientation of such models, which do not prioritize all-terrain handling.

The heatmap further reveals that sedans and hatchbacks show the most diverse drivewheel usage, albeit still dominated by fwd. In contrast, convertibles and hardtops exhibit less variety and adhere more closely to rwd setups. Wagons display a moderate spread across fwd and 4wd, possibly balancing urban utility and outdoor adaptability. Overall, the analysis highlights that drivewheel configuration is intricately tied to vehicle body type, reflecting engineering priorities such as performance, cost, consumer safety, and terrain compatibility. These associations also help manufacturers strategically position their products within segmented automobile markets.

4.4 Inspection of Outliers & Removal

A boxplot is a graphical representation of the distribution of a dataset that can be used to identify outliers.

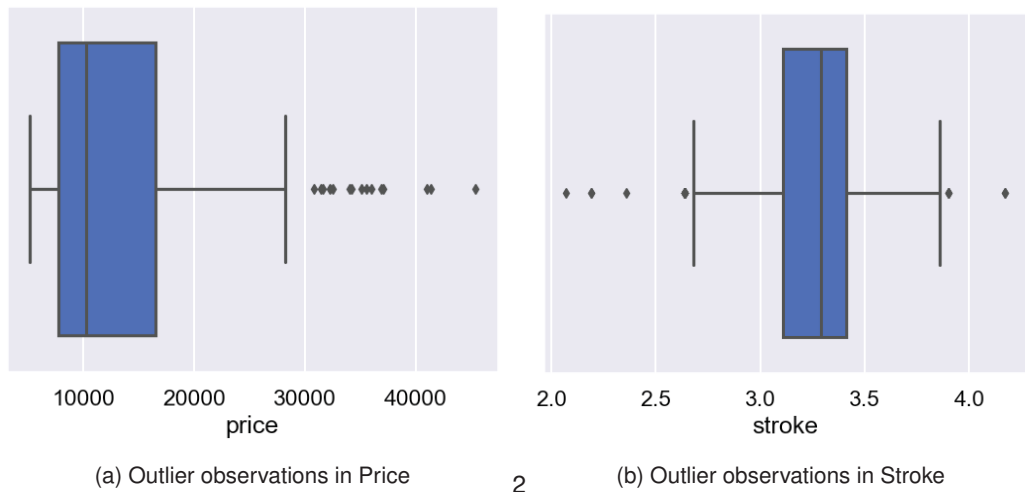


Figure 5: Outlier Analysis of key variables

The visual inspection of boxplots for numerical variables revealed the presence of extreme

outliers. In particular, figure 5 the *price* variable displayed multiple high-end outliers, likely representing luxury vehicles that could disproportionately influence the model's performance if left untreated. Similarly, the *stroke* variable showed isolated high values that appeared inconsistent with the overall distribution, suggesting the presence of measurement or entry anomalies. These observations necessitated a careful strategy for outlier treatment prior to model building to ensure robustness and prevent distortion in coefficient estimation.

4.4.1 Winsorizing: A process of removal of outlier

To handle outliers without discarding valuable data, the Winsorizing technique was employed. Outliers were first identified using the Interquartile Range (IQR) method, where any value lying below $Q1 - 1.5 \times IQR$ or above $Q3 + 1.5 \times IQR$ was flagged as an outlier. Rather than removing these points, Winsorizing replaces extreme values with the nearest acceptable boundary values—specifically, the lower and upper limits defined by the IQR rule. This approach preserves the sample size and overall distribution shape while mitigating the influence of extreme values that could skew model estimates or reduce generalization performance.

This process ensures that extreme values are pulled back within the acceptable range, reducing their impact. This can be ensured from the following figures (Figure 6 a & b) of the same variables that had been shown in the previous figures of 5 a & b)

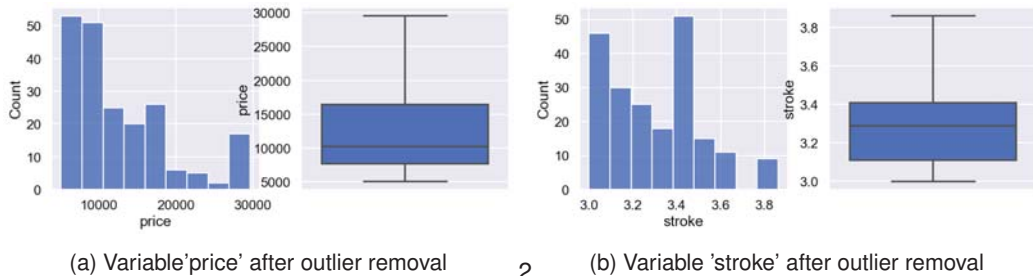


Figure 6: Removal of outliers from variables

4.5 Data Visualization and Exploratory Data Analysis

The dataset contains categorical information about the safety rating of cars, which we converted to a numerical scale to reflect their hierarchical order. The ranking of car safety from highest to lowest, as inferred from the dataset, is as follows:

- Very Safety Car
- Safety Car
- Much More Risky Car
- Normal Car

- Risky Car
- Very Risky Car

This ranking indicates that cars with higher safety ratings are generally more expensive, reflecting consumer preferences for safety features (Figure 7 a). Incorporating this hierarchy into the model helps capture the inherent value consumers place on safety, improving the accuracy of price predictions.

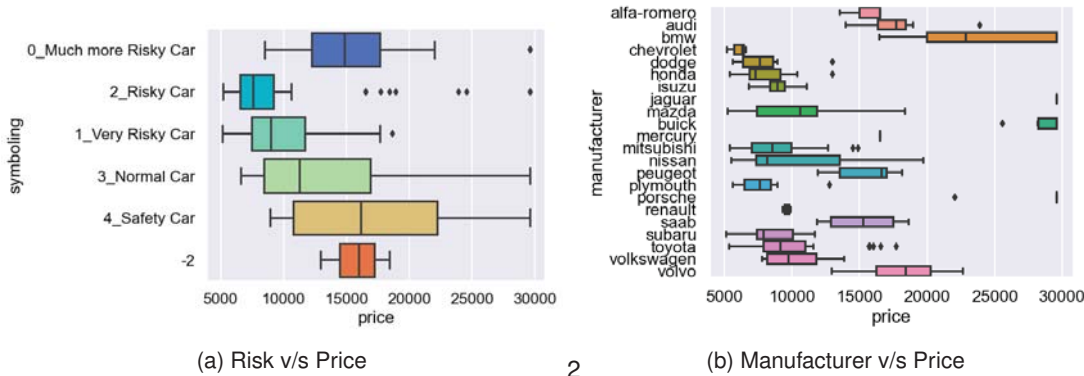


Figure 7: Effect of safety features and manufacturer on price of a car

From figure 7 b, it was evident that certain car brands consistently had higher prices compared to others. In particular, Porsche, Jaguar, and Buick emerged as the brands with the most expensive cars. This observation aligns with market trends where luxury and high-performance brands command higher prices due to their superior build quality, brand reputation, advanced technology, and exclusive features.

Including brand information in the model is crucial as it accounts for brand prestige and market positioning. By recognizing that Porsche, Jaguar, and Buick are luxury brands, the model can more accurately predict car prices, reflecting the premium associated with these brands.

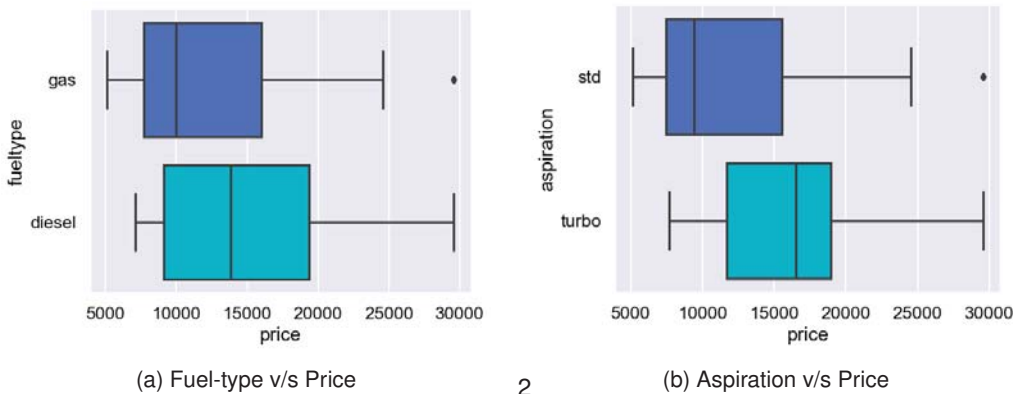


Figure 8: Effect of type of fuel and aspiration on price of a car

The visual analysis of Figure 8 a reveals that diesel vehicles, on average, are priced slightly higher than their petrol counterparts. This may reflect the relatively higher fuel efficiency and engine durability associated with diesel cars, which can contribute to increased manufacturing costs and thus a higher market price. However, the overall price range for both fuel types shows considerable overlap, indicating that fuel type alone is not a dominant factor in determining car price.

Figure 8 b illustrates that cars equipped with turbo aspiration tend to have a wider and generally higher price range compared to those with standard aspiration. Turbocharged vehicles are typically associated with enhanced engine performance and are often found in sportier or premium segments. This correlation suggests that aspiration type has a stronger influence on pricing than fuel type, as manufacturers may charge a premium for vehicles offering increased power and driving experience. Together, these observations highlight that while fuel type has a moderate impact, aspiration plays a more significant role in influencing car prices.

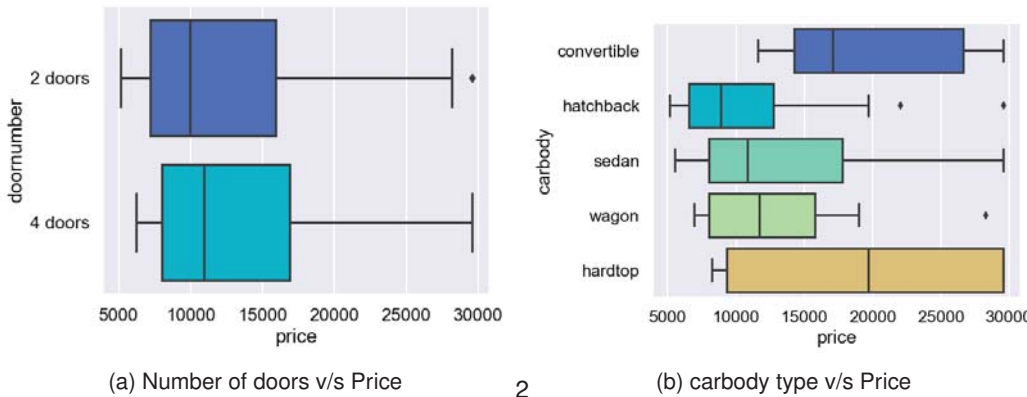


Figure 9: Effect of number of doors and carbody on price

Figure 9 a indicates that the number of doors does not significantly influence car price. Both two-door and four-door vehicles show overlapping price distributions, with only marginal differences in average pricing. This suggests that door count is likely a secondary design attribute rather than a key pricing determinant, possibly influenced more by style or segment expectations than production cost or performance.

In contrast, Figure 9 b shows that car body type plays a much more substantial role in price variation. Convertibles and hardtops exhibit higher price ranges, reflecting their premium positioning and specialized design features. Sedans and wagons occupy the mid-range, offering a balance between utility and comfort, while hatchbacks are generally positioned at the lower end of the price spectrum, likely due to their compact size and efficiency-oriented design. These distinctions affirm that carbody type is a strong indicator of the vehicle's market segment and thus has a more direct influence on pricing compared to the number of doors.

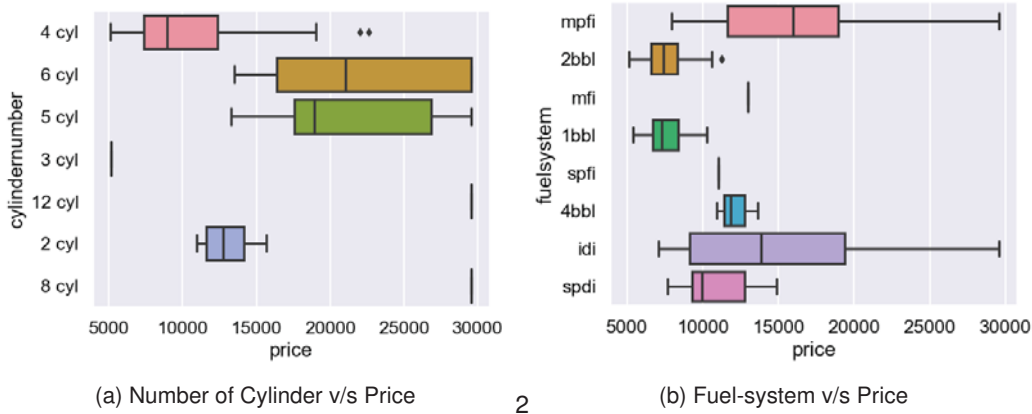


Figure 10: Effect of number of cylinders and fuel system on price of a car

The visualization of Figure 10 a highlights a clear upward trend in car prices with an increase in the number of engine cylinders. Vehicles with three, four, or five cylinders are generally concentrated in the lower price segment, while those with six, eight, and twelve cylinders are placed significantly higher on the price scale. This suggests a strong association between cylinder count and engine performance, where more cylinders typically translate into greater power and, consequently, a higher market valuation.

Figure 10 b examines the relationship between fuel system types and pricing. Common systems such as MPFI and 2BBL appear across a wide range of price levels, indicating their use in both budget and mid-range vehicles. In contrast, specialized systems like idi, mfi, and 4BBL are found in more narrowly defined, often higher-priced segments, likely due to their technical advantages or association with specific engine configurations. Although fuel system alone may not determine price, it serves as an indirect indicator of vehicle complexity and engineering level, influencing where a car is positioned in the pricing spectrum.

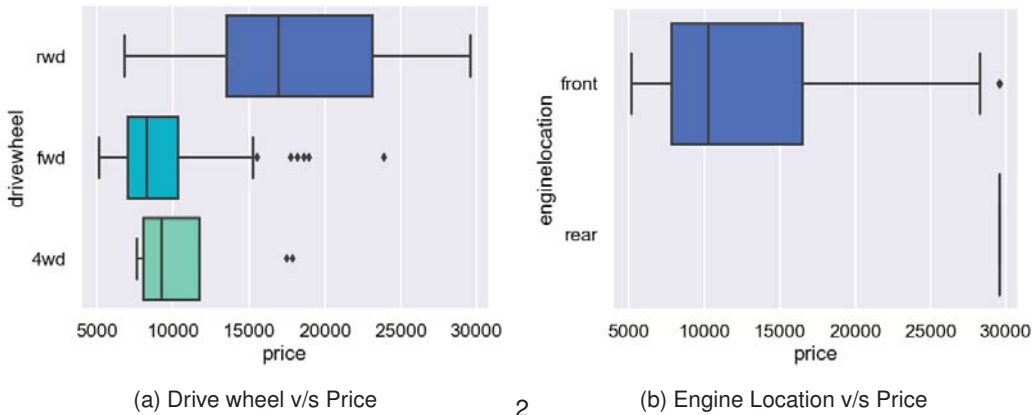


Figure 11: Effect of drivewheel and engine location on price of a car

Figure 11 a reveals a notable distinction in pricing across different drivewheel configurations. Front-wheel drive (fwd) vehicles are the most common and typically fall within the lower to mid-price range, reflecting their efficiency, affordability, and popularity in mass-market segments. Rear-wheel drive (rwd) cars, on the other hand, are associated with higher prices, likely due to their presence in premium and performance-oriented models where enhanced driving dynamics are prioritized. All-wheel drive (4wd) vehicles appear in a limited number of cases but tend to occupy the mid to upper price tiers, possibly due to their use in specialized utility or off-road vehicles.

Figure 11 b shows a stark contrast in pricing based on engine location. Vehicles with rear engine placement tend to be priced significantly higher than those with front engines. This pattern aligns with the fact that rear-engine designs are often used in high-performance or luxury cars, where engineering complexity and brand positioning justify a premium price. In contrast, front-engine vehicles dominate the dataset and are spread across the full pricing spectrum, especially in the low to mid-range. Overall, both drivewheel configuration and engine location emerge as influential factors in shaping car pricing, reflecting broader design philosophies and market segmentation strategies.

4.6 Model Performance and Analysis

4.6.1 Linear Regression results

In evaluating the linear regression model fitted on the car price dataset, several key metrics provide insight into its performance, particularly in the presence of multicollinearity. Below is a brief analysis based on the provided accuracy metrics:

Table 1: Performance of Linear Regression on Training and Testing Sets

Metrics	Training Set	Testing Set
R^2	0.98	0.84
RSS	210406502.35	702234087.12
MSE	1471374.14	11327263.40
RMSE	1213.00	3365.60

In the context of table 1 the R^2 value indicates the proportion of variance in the dependent variable that is predictable from the independent variables. An R^2 value of approximately 0.976 on the training set suggests that the model explains 97.57% of the variance in car prices on the training data, indicating excellent fit on the training set. However, the R^2 value drops to approximately 0.838 on the testing set, implying that the model explains only 83.83% of the variance in car prices on the test data. This significant drop indicates that the model may be overfitting, capturing noise in the training data rather than the underlying patterns.

RSS measures the total squared deviation of the predicted values from the actual values. A lower RSS indicates a better fit. The much lower RSS on the training set compared to the

testing set suggests that the model fits the training data much better than the test data, reinforcing the evidence of overfitting.

MSE is the average of the squared differences between the predicted and actual values. Like RSS, a lower MSE indicates better model performance. The MSE on the training set is significantly lower than that on the testing set, indicating the model performs much better on the training data. This disparity further supports the conclusion of overfitting.

The performance of the linear regression model on the car price dataset reveals a clear case of overfitting. The high R^2 value on the training set, combined with the much lower R^2 value on the testing set, suggests that while the model fits the training data very well, it does not generalize as effectively to unseen data. This observation is corroborated by the significant differences in RSS and MSE between the training and testing sets. The presence of multicollinearity likely exacerbates this issue by inflating the variance of the coefficient estimates, leading to unstable predictions on new data.

To mitigate overfitting and improve model generalization, techniques such as regularization (e.g., Ridge or Lasso regression) or employing more robust algorithms like XGBoost or Random Forest could be beneficial.

4.6.2 Ridge Regression results

Following the initial linear regression model, Ridge regression was applied to the car price dataset to address the multicollinearity issues and mitigate overfitting. Below is an evaluation of the model's performance based on the provided accuracy metrics:

Table 2: Performance Metrics for Ridge Regression Model

Metrics	Training Set	Testing Set
R^2	0.95	0.93
RSS	355957028.15	229722494.96
MSE	2489209.99	3705201.51
RMSE	1577.61	1924.89

In the context of table 2 the R^2 value for the Ridge regression model on the training set is approximately 0.959, indicating that the model explains 95.89% of the variance in car prices on the training data. This is slightly lower than the R^2 value observed with the linear regression model, which is expected due to the regularization effect of Ridge regression. Importantly, the R^2 value on the testing set is approximately 0.922, which is much closer to the training R^2 value compared to the linear regression model. This suggests that the Ridge regression model generalizes better to unseen data, reducing overfitting.

RSS measures the total squared deviation of the predicted values from the actual values. The RSS for the Ridge regression model on the training set is higher compared to the linear

regression model, which is expected as regularization tends to decrease the flexibility of the model. However, the RSS on the testing set is significantly lower than that of the linear regression model, indicating improved performance and better generalization.

MSE is the average of the squared differences between the predicted and actual values. The MSE values for both the training and testing sets are higher in the Ridge regression model compared to the linear regression model on the training set but significantly lower on the testing set. This reduction in MSE on the testing set highlights the model's improved ability to generalize to new data.

The application of Ridge regression on the car price dataset demonstrates the effectiveness of regularization in addressing multicollinearity and reducing overfitting. The training and testing R^2 values are closer to each other, indicating better generalization compared to the linear regression model. Although the RSS and MSE values are higher on the training set, they are substantially lower on the testing set, further emphasizing the improved performance of the Ridge regression model in predicting car prices on unseen data.

By incorporating the L_2 penalty, Ridge regression shrinks the coefficients of less important features, thus stabilizing the model and enhancing its predictive accuracy on new data. This balance between model flexibility and regularization proves beneficial in overcoming the limitations observed in the initial linear regression model.

4.6.3 Lasso Regression results

Following the implementation of Ridge regression, Lasso regression was applied to the car price dataset to further address multicollinearity and overfitting concerns. The Lasso regression model employs L_1 regularization, which can shrink some coefficients to zero, effectively performing feature selection. Here's a brief evaluation of the model's performance based on the provided metrics:

Table 3: Performance Metrics for Lasso Regression Model

Metrics	Training Set	Testing Set
R^2	0.93	0.88
RSS	200866611.54	385884068.02
MSE	3239785.85	6223936.58
RMSE	1799.94	2494.78

From table 3 the R^2 value for the Lasso regression model on the training set is approximately 0.93, indicating that the model explains 93.15% of the variance in car prices on the training data. This value is lower than the R^2 values observed for both the linear and Ridge regression models, which is expected due to the L_1 regularization effect. On the testing set, the R^2 value is approximately 0.88, slightly lower than the Ridge regression but still quite close to the training R^2 value. This suggests good generalization with reduced overfitting.

RSS measures the total squared deviation of the predicted values from the actual values. The RSS for the Lasso regression model on the training set is higher compared to both the linear and Ridge regression models, indicating a less tight fit on the training data. However, the RSS on the testing set is lower than that of the linear regression model, though slightly higher than Ridge regression, suggesting better generalization to unseen data compared to the initial linear model.

MSE is the average of the squared differences between the predicted and actual values. The MSE values for both the training and testing sets are higher in the Lasso regression model compared to the Ridge regression model but lower than those for the linear regression model. This indicates a balanced trade-off between fitting the training data and maintaining good generalization to the testing data.

The application of Lasso regression on the car price dataset demonstrates the effectiveness of L_1 regularization in addressing multicollinearity and reducing overfitting. While the R^2 value is slightly lower than that of Ridge regression, the close proximity of the R^2 values between the training and testing sets indicates good generalization.

The higher RSS and MSE values on the training set compared to Ridge regression are expected due to the feature selection property of Lasso, which can lead to a sparser model with some coefficients shrunk to zero. This sparsity helps in improving model interpretability by retaining only the most significant features, though it may result in a slightly less tight fit.

From the above tables overall, Ridge regression has effectively balanced the model's complexity and generalization ability, making it a robust choice for predictive modeling in the presence of multicollinearity. While Ridge regression showed slightly better predictive accuracy, Lasso's ability to perform feature selection can be particularly valuable when dealing with high-dimensional datasets. The following figures (Figure 12) are obtained from the residuals obtained from the Ridge Regression and it can be noticed that the residuals have met the assumptions:

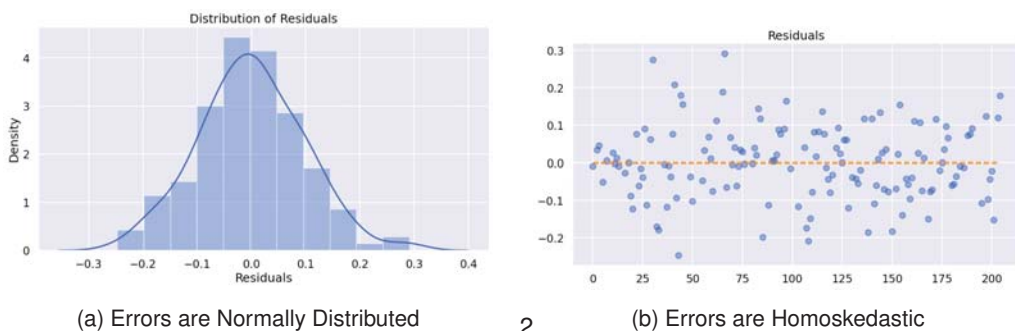


Figure 12: Residuals of Ridge regression

4.7 Performance of advance Models: Bagging & Boosting

The performance comparison between Random Forest (bagging) and XGBoost (boosting) revealed meaningful insights into their respective strengths. The Random Forest model demonstrated better generalization, as indicated by a lower difference between training and test errors. This highlights the model's ability to balance bias and variance effectively while resisting overfitting, even with complex datasets. It performed robustly across different subsets of the data and required less computational time due to its parallel tree construction.

On the other hand, XGBoost achieved slightly higher accuracy on the training data, benefiting from its iterative tree-building approach and sophisticated optimization with regularization. However, this came at the cost of longer execution time and a marginally larger generalization gap, suggesting a higher sensitivity to overfitting if not carefully tuned. Despite these trade-offs, XGBoost remains a powerful tool for capturing complex patterns and feature interactions. Together, both models showcase the trade-off between computational efficiency, generalization ability, and predictive accuracy in ensemble-based approaches.

4.7.1 Reasons for higher Execution time

The higher execution time observed in the XGBoost model can be attributed to several computational and structural factors inherent to boosting algorithms (Lai, Shahri, Mohamad, Rahman and Rambli, 2021). First, unlike bagging methods, boosting algorithms such as XGBoost build trees sequentially rather than in parallel. This means that each tree must wait for the output of the previous one, inherently increasing the training time. Second, XGBoost incorporates regularization techniques, including both L_1 and L_2 penalties, to control model complexity and reduce overfitting. These additional optimization steps introduce extra computation at each iteration. Third, the model evaluates a vast number of split possibilities during the tree-building process, especially in the presence of many features, which significantly extends training time. Finally, hyperparameter tuning in XGBoost is more extensive compared to simpler models like Random Forest or Linear Regression, often requiring repeated cross-validation to identify optimal values for parameters such as learning rate, max depth, and number of estimators. Collectively, these factors contribute to the longer execution time of XGBoost, albeit with improved predictive performance.

Table 4: Comparison of Model Performance Metrics for Random Forest and XGBoost

Metrics	Random Forest	XGBoost
R^2 (Training set)	0.93	0.95
R^2 (Testing set)	0.89	0.91
RSS (Training set)	128632841.41	21431372.15
RSS (Testing set)	385913477.33	409282624.61
MSE (Training set)	899530.36	149869.74
MSE (Testing set)	6224410.92	6601332.66
RMSE (Training set)	948.44	387.13
RMSE (Testing set)	2494.87	2569.31

The table 4 both Random Forest and XGBoost are powerful models that provide high accuracy but at the cost of increased computational complexity and execution time. Random Forest seems to have a slight edge in this specific car price dataset in terms of generalization performance, while XGBoost's more extensive regularization and optimization steps lead to longer execution times but also contribute to its robustness and accuracy in many scenarios.

4.7.2 Final Observations

The final refined model produced insightful findings that are both statistically and practically meaningful. Notably, the coefficients for *enginesize*, *curbweight*, and *horsepower* were all positive and significant, confirming that these factors play a direct role in driving car prices upward. Larger engine sizes and higher horsepower generally reflect more powerful, premium vehicles, while higher curbweight may indicate greater material usage or enhanced build quality, both of which justify a higher price point.

Interestingly, variables such as *citympg* and *highwaympg* exhibited negative coefficients. This suggests that cars with higher fuel efficiency tend to be more affordable, potentially catering to budget-conscious or environmentally focused consumers. These inverse relationships align with the common trade-off in the automotive industry between performance and efficiency.

Moreover, the variable *carwidth* was also found to have a positive and meaningful coefficient, indicating that broader vehicles—often perceived as more stable and spacious—are priced higher. The model effectively excluded irrelevant or redundant predictors through the regularization process, leading to a simpler, more interpretable solution. Ultimately, the findings validate the model's ability to distinguish between performance-driven luxury cars and cost-effective fuel-efficient models, aligning well with industry pricing logic and consumer expectations.

5 Related Work

Predicting car prices using regression techniques is a well-researched area in machine learning and statistics. Numerous studies have explored various methodologies and their effectiveness in different contexts. Below, we summarize some notable works relevant to this study.

One of the fundamental studies on linear regression is the work by (Seber and Lee, 2012), who provide a comprehensive overview of linear regression models, their assumptions, and their applications. Their research highlights the strengths and limitations of linear regression, particularly in the presence of multicollinearity, which is a common issue in car price prediction datasets.

"Ridge Regression: Biased Estimation for Nonorthogonal Problems," by (Hoerl and Kennard, 1970) introduced ridge regression as a solution to multicollinearity in multiple regression analysis. Their work demonstrated how adding a penalty term to the regression reduces variance and improves model stability, a concept that has since been widely adopted in car price prediction models to handle correlated features effectively.

"Regression Shrinkage and Selection via the Lasso," (Tibshirani, 1996) presented the Lasso (Least Absolute Shrinkage and Selection Operator) method. This work is pivotal in the field of regression analysis, as it combines regularization with feature selection, making it highly relevant for datasets with many predictors. Lasso's ability to produce sparse models has made it particularly useful in car price prediction, where some features may have little to no impact on the target variable.

Recent comparative studies, such as "Used car price prediction using different machine learning algorithms" (Bharambe, Bagul, Dandekar and Ingle, 2022) provide empirical evidence on the performance of these regression techniques in real-world scenarios. These studies typically find that while linear regression can suffer from overfitting and multicollinearity, ridge and lasso regression offer improved predictive performance and model interpretability.

Furthermore, advanced studies such as Car Price Prediction using Machine Learning Techniques" (Gegic, Isakovic, Keco, Masetic and Kevric, 2019) explore the application of machine learning algorithms beyond traditional regression techniques. Their research includes comparisons of regression methods with other machine learning approaches, providing insights into the conditions under which each method performs best.

The literature indicates a robust interest in regression techniques for car price prediction, with each method offering unique advantages. Linear regression serves as a foundational model, while ridge and lasso regression address its limitations by incorporating regularization. Comparative studies and machine learning applications provide valuable benchmarks and highlight the continued evolution of predictive modeling in this domain. This study's approach to evaluating linear, ridge, and lasso regression aligns with these studies, aiming to leverage their

findings for enhanced predictive accuracy and model robustness.

6 Conclusion

This study developed a comprehensive regression-based framework to predict car prices using a real-world dataset comprising both numerical and categorical features. Through a structured modeling pipeline, the analysis began with linear regression to establish a baseline, followed by Ridge and Lasso regression to address multicollinearity and overfitting. Advanced ensemble methods—Random Forest and XGBoost—were employed to enhance predictive performance through bagging and boosting strategies.

The analysis revealed that features such as *enginesize*, *curbweight*, *horsepower*, and *carwidth* significantly influence car prices, while fuel efficiency metrics like *citympg* and *highwaympg* showed an inverse relationship. Regularization helped streamline the model by retaining only the most relevant predictors. Among the models tested, Random Forest offered the best balance between accuracy and generalization, whereas XGBoost yielded the highest training performance with a slight overfitting tendency and longer execution time.

In summary, the study demonstrates how statistical and machine learning methods can be effectively applied to real-world automotive pricing problems, offering both predictive power and business interpretability. These insights are valuable for automobile manufacturers and marketers seeking to align product features with customer value and pricing strategy.

7 Acknowledgment

We extend our gratitude to Creative Commons for providing the invaluable dataset used in this research. We sincerely thank the Associate Editors and the reviewers for their careful evaluation and constructive feedback.

References

- Avanijaa, J. et al. 2021. Prediction of house price using xgboost regression algorithm, *Turkish Journal of Computer and Mathematics Education (TURCOMAT)* **12**(2): 2151–2155.
- Bager, A., Roman, M., Algedih, M. and Mohammed, B. 2017. Addressing multicollinearity in regression models: a ridge regression application.
- Bharambe, P. P., Bagul, B., Dandekar, S. and Ingle, P. 2022. Used car price prediction using different machine learning algorithms, *Int. J. Res. Appl. Sci. Eng. Technol* **10**: 773–778.
- Gegic, E., Isakovic, B., Keco, D., Masetic, Z. and Kevric, J. 2019. Car price prediction using machine learning techniques, *TEM Journal* **8**(1): 113.

- Goeman, J., Meijer, R. and Chaturvedi, N. 2018. L1 and L2 penalized regression models, *Vignette R Package Penalized*. URL <http://cran.r-project.org/web/packages/penalized/vignettes/penalized.pdf>.
- Grömping, U. 2009. Variable importance assessment in regression: linear regression versus random forest, *The American Statistician* **63**(4): 308–319.
- Hawkins, D. M. 2004. The problem of overfitting, *Journal of chemical information and computer sciences* **44**(1): 1–12.
- Hoerl, A. E. and Kennard, R. W. 1970. Ridge regression: Biased estimation for nonorthogonal problems, *Technometrics* **12**(1): 55–67.
- Kilmer, J. T. and Rodríguez, R. L. 2017. Ordinary least squares regression is indicated for studies of allometry, *Journal of evolutionary biology* **30**(1): 4–12.
- Lai, S. B. S., Shahri, N., Mohamad, M. B., Rahman, H. and Rambli, A. B. 2021. Comparing the performance of adaboost, xgboost, and logistic regression for imbalanced data, *Mathematics and Statistics* **9**(3): 379–385.
- Li, Y., Zou, C., Berecibar, M., Nanini-Maury, E., Chan, J. C.-W., Van den Bossche, P., Van Mierlo, J. and Omar, N. 2018. Random forest regression for online capacity estimation of lithium-ion batteries, *Applied energy* **232**: 197–210.
- McDonald, G. C. 2009. Ridge regression, *Wiley Interdisciplinary Reviews: Computational Statistics* **1**(1): 93–100.
- Ranstam, J. and Cook, J. A. 2018. Lasso regression, *Journal of British Surgery* **105**(10): 1348–1348.
- Roth, V. 2004. The generalized lasso, *IEEE transactions on neural networks* **15**(1): 16–28.
- Seber, G. A. and Lee, A. J. 2012. *Linear regression analysis*, John Wiley & Sons.
- Seger, C. 2018. An investigation of categorical variable encoding techniques in machine learning: binary versus one-hot and feature hashing.
- Tibshirani, R. 1996. Regression shrinkage and selection via the lasso, *Journal of the Royal Statistical Society Series B: Statistical Methodology* **58**(1): 267–288.
- Uyanik, G. K. and Güler, N. 2013. A study on multiple linear regression analysis, *Procedia-Social and Behavioral Sciences* **106**: 234–240.
- Vidaurre, D., Bielza, C. and Larranaga, P. 2013. A survey of L1 regression, *International Statistical Review* **81**(3): 361–387.
- Wang, R., Wang, L., Zhang, J., He, M. and Xu, J. 2022. Xgboost machine learning algorithm performed better than regression models in predicting mortality of moderate-to-severe traumatic brain injury, *World Neurosurgery* **163**: e617–e622.