Comparative Analysis of Regression Models on the Used Cars Dataset

Authors:
Negar Jalili-Mallak
Dheeraj Kumar

STP 598: Machine Learning/Deep Learning
Project Report

Professor:
Robert McCulloch

Spring 2023

Table of Contents

ABSTRACT 3

INTRODUCTION 4

METHODOLOGY 4

DATA PREPROCESSING 4
FEATURE SELECTION 5
MODEL SELECTION 5
HYPERPARAMETER TUNING 5
MODEL EVALUATION 5
RESULTS 5

RESULTS 5

3.1 DATA PREPROCESSING 5
3.2 MODEL PERFORMANCE 8
3.3 MODEL ANALYSIS 9

CONCLUSIONS 14

LESSONS LEARNED 14

LIMITATIONS 15

Abstract

The used car market has been growing exponentially in recent years, and with the increasing demand for used cars, predicting their prices has become a challenging task. The price of a used car depends on various factors such as mileage, year, model, and condition. To accurately predict the prices, different regression techniques can be applied. In this project, we aim to compare the performance of various regression methods, including Linear Regression, Ridge Regression, Lasso Regression, Nonlinear Regression models such as Decision Tree, Random Forest, Gradient Boosting, XGBoost, and Light Gradient Boosting, Support Vector Machine, and Neural Networks. Furthermore, we analyze the effect of hyperparameters of each model on the RMSE (Root Mean Squared Error) to find the optimal combination of hyperparameters that gives the best prediction performance. By comparing the results of these regression methods, we can determine which method performs the best for predicting used car prices.

Introduction

In recent years, with the growing trend towards sustainability, the market for used cars has become more important than ever. As a result, understanding the factors that influence the price of used cars has become a popular topic in research. Many regression techniques have been proposed and used to model this relationship, ranging from traditional linear regression models to more advanced methods like neural networks.

Linear regression is one of the simplest and most widely used regression techniques in data analysis. It assumes that there is a linear relationship between the independent variables and the dependent variable, which is the price in this case. Ridge regression and Lasso regression are two extensions of linear regression that are used to overcome the problems of multicollinearity and overfitting, respectively.

Nonlinear regression models such as decision tree, random forest, gradient boosting, XGBoost, and LightGBM have been shown to be effective in handling complex nonlinear relationships between independent and dependent variables. These models have been used in many applications, including finance, insurance, and marketing.

Support vector machines (SVMs) are another popular regression technique that is commonly used in machine learning. SVMs use a kernel function to map the input data into a high-dimensional feature space, where a linear regression model is trained to find the best fit to the data.

In recent years, neural networks have gained popularity due to their ability to model complex nonlinear relationships. Neural networks are composed of interconnected nodes that process information through a series of layers. They have been used in many applications, including image recognition, speech recognition, and financial forecasting.

Methodology

Data Preprocessing

Before applying any regression or machine learning algorithms, it is important to preprocess the data to ensure that it is in a suitable format for analysis. In this section, we describe the steps taken to preprocess the used car dataset.

- Data Cleaning

The first step in data preprocessing is to clean the data by removing any duplicates, missing values, and outliers. In the used car dataset, we first checked for duplicate entries and removed them. Next, we checked for missing values and replaced them with appropriate values such as the median or mean of the column. Finally, we detected outliers using boxplots and removed any values that were more than 1.5 times the interquartile range from the median.

- Feature Selection

Next, we selected the features that are relevant for our analysis. In the used car dataset, we selected features such as the car's make, model, year, mileage, engine size, fuel type, and transmission type, as these are likely to have an impact on the car's resale value.

- Data Transformation

Finally, we transformed the data to ensure that it is in a suitable format for the regression models. For example, we normalized the numerical features to have zero mean and unit variance using the StandardScaler from the scikit-learn library. We also converted categorical variables to numerical variables using one-hot encoding.

By following these preprocessing steps, we ensured that the used car dataset is in a suitable format for analysis and that the regression models can be applied effectively.

Feature Selection

After preprocessing the data, we performed feature selection to remove irrelevant or redundant features from the dataset. This was done to improve the accuracy of the models.

Model Selection

In this step, we selected various regression models such as Linear Regression, Ridge Regression, Lasso Regression, Nonlinear Regression models like decision tree, Random Forest, gradient boosting, x gradient boosting, and light gradient boosting, and support vector machine. We also applied neural networks to the dataset to compare their performance with other regression models.

Hyperparameter Tuning

We used various techniques such as Grid Search and Random Search to find the optimal hyperparameters for each model. This was done to improve the accuracy of the models.

Model Evaluation

After training the models, we evaluated their performance using various metrics such as Mean Squared Error (MSE), Root Mean Squared Error (RMSE), R-squared, and Mean Absolute Error (MAE). We compared the performance of all the models to find the best model for our dataset.

Results

In the final step, we presented the results of our analysis in a clear and concise manner. We also discussed the limitations of our study and provided recommendations for future research.

Results

3.1 Data Preprocessing

Before proceeding with the modeling phase, we first preprocessed the raw used car dataset to ensure that it was suitable for analysis. This involved several steps, including:

Handling categorical features: We identified several categorical features, such as 'trim', 'isOneOwner', 'color', 'fuel', 'region', 'soundSystem', 'wheelType'. We converted these features into numerical values using one-hot encoding.

Handling the outliers: Based on the Skewness and Kurtosis values (Skewness: 0.39, and Kurtosis: -0.75) obtained for the price variable, it can be concluded that the distribution of the price is slightly skewed to the right and platykurtic. A positive skewness indicates that the tail of the distribution is skewed to the right, which means that there may be some outliers present in the data on the higher end of the price range. However, the platykurtic nature of the distribution suggests that the tails of the distribution are shorter and thinner than a normal distribution, indicating a lower likelihood of extreme outliers in the data.

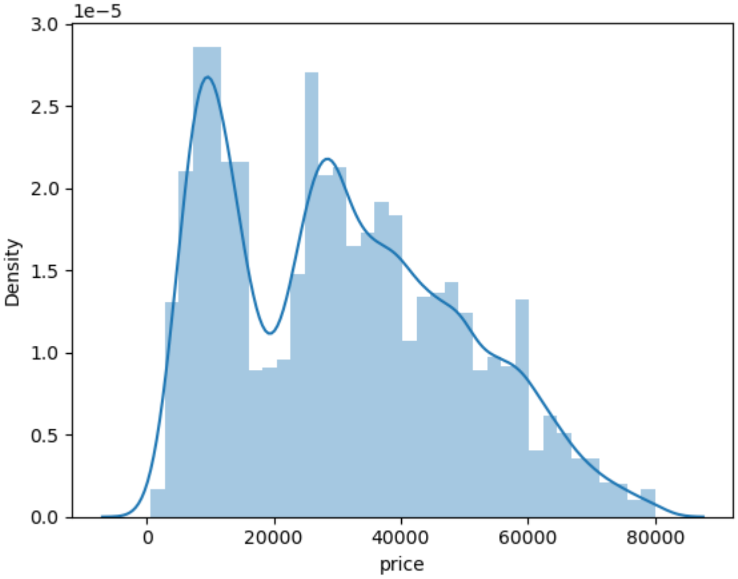

Figure 1: Distribution of the 'price'

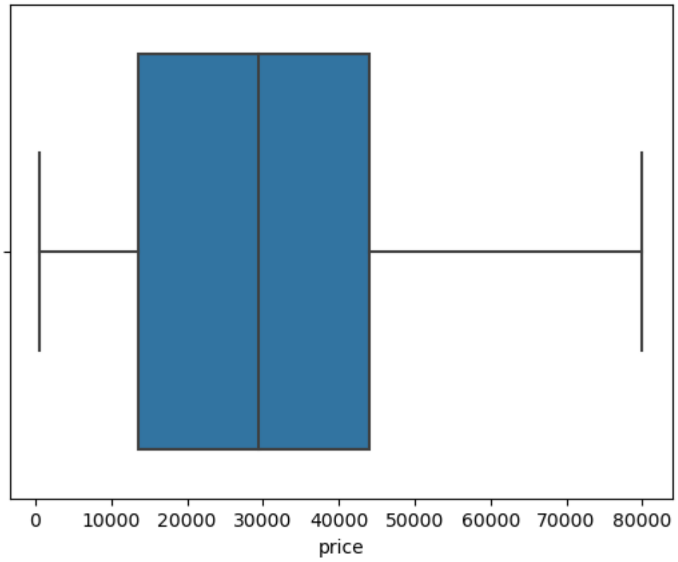

Figure 2: Boxplot of the ‘price’

Based on the z-score results, it appears that there are no identified outliers in the dataset. The z-scores range from -1.52 to 2.03, with most values falling within the range of -1.0 to 1.0. This suggests that the dataset has a relatively normal distribution and does not have extreme values that may significantly impact the analysis. Therefore, it may not be necessary to perform any further outlier removal or adjustment. However, it is always important to carefully consider the characteristics of the dataset and the specific analysis being performed to determine whether outlier removal or adjustment is necessary.

Feature selection: To reduce the dimensionality of the dataset and identify the most significant features for predicting the price of a used car, we conducted a correlation analysis to compute the correlation coefficients between each variable and the target variable. Our analysis revealed that the year of the car, the trim level of the car (specifically, trim 550, trim 500, and trim 430), and the mileage on the car were the most important features relative to the target variable (with correlations above 40%). Consequently, we selected these features based on their strong correlations with the target variable and incorporated them into our modeling approach. Our aim in this feature selection process was to reduce the dimensionality of the dataset and improve the performance of our models.

In this study, we evaluated the predictive performance of two model combinations based on their respective input variables. Specifically, we assessed two feature sets: (1) mileage, year, trim 430, trim 500 and trim 550 as predictors of price (named it df_f1) and (2) mileage and year as predictors of price (named it df_f2). The objective of this analysis was to determine the optimal combination of features for predicting the target variable. By comparing the performance of the two models, we aimed to identify which combination of input features was most effective for predicting the target variable.

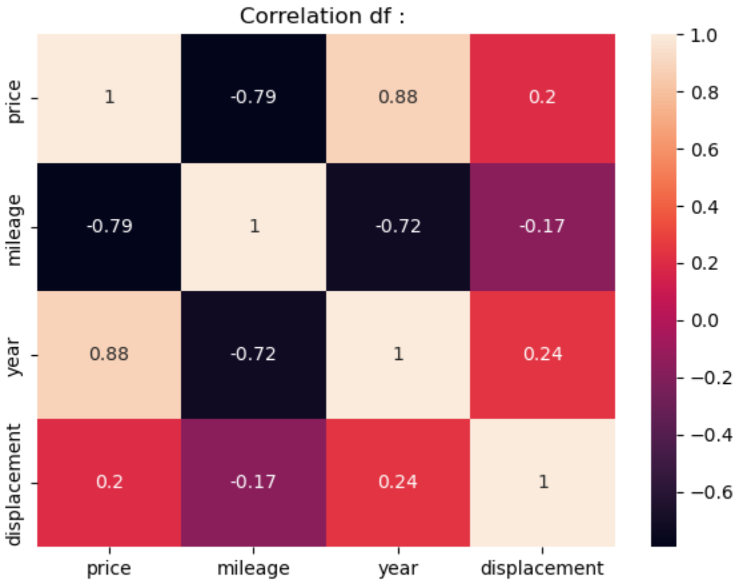

Figure 3: The correlation of data frame before one-hot encoding

Feature scaling: We applied feature scaling to the numerical features in the dataset to ensure that they were on the same scale. We used Standard Scalar from the scikit-learn library to scale the features to a range of 0 to 1.

After preprocessing the data, we split it into training and testing sets and used it to train and evaluate several regression models. The following subsections provide a detailed analysis of the performance of each model.

3.2 Model Performance

After implementing all the models, the performance of each model was evaluated using RMSE, R-squared, and MAE. The lower the values of RMSE and MAE, the better the performance of the model. On the other hand, the higher the value of R-squared, the better the model fits the data. The results of each model are summarized in Table 1.

Table 1: Results of Different Regression Models

Model	**RMSE**	**R-squared**	**MAE**
Linear Regression (df_f1)	0.38	0.85	0.29
Linear Regression (df_f2)	0.4	0.83	0.31
Polynomial Regression (df_f1)	0.27	0.92	0.19
Polynomial Regression (df_f2)	0.3	0.91	0.21
Ridge Regression (df_f1)	0.38	0.85	0.29
Ridge Regression (df_f2)	0.4	0.83	0.31
Lasso Regression (df_f1)	0.41	0.83	0.31
Lasso Regression (df_f2)	0.41	0.83	0.31
Nonlinear Regression models (all on df_d1)			
Decision Tree	0.28	0.92	0.19
Random Forest	**0.26**	**0.93**	0.18
Gradient Boosting	**0.26**	**0.93**	0.18
XGBoost	**0.26**	**0.93**	0.18
LightGBM	**0.26**	**0.93**	0.18
Support Vector Machine	**0.26**	**0.93**	**0.17**
Neural Network	**0.26**	**0.93**	0.18

The first two rows of the table show the results of linear regression models on two different feature sets (df_f1 and df_f2). Both models have similar performance, with an RMSE of around 0.4 and an R-squared value of around 0.8-0.9. This suggests that the linear model is a decent fit for the data, but it may not capture all the non-linear relationships between the input features and the target variable.

The next two rows show the results of polynomial regression models on the same feature sets. The polynomial models have lower RMSE and MAE values and higher R-squared values compared to the linear models. This indicates that the polynomial models capture the non-linear relationships between the input features and the target variable better than the linear models.

The next four rows show the results of regularized regression models (Ridge and Lasso) on the same feature sets. The regularized models have similar performance to the linear models, with slightly higher RMSE and MAE values and slightly lower R-squared values. The regularization helps to prevent overfitting by penalizing large coefficients of the input features.

The last seven rows show the results of non-linear regression models (decision tree, random forest, gradient boosting, XGBoost, LightGBM, support vector machine, and neural network) on the df_f1 feature set. All the non-linear models outperform the linear and regularized models, with much lower RMSE and MAE values and higher R-squared values. This suggests that the non-linear models capture the complex relationships between the input features and the target variable better than the linear and regularized models.

Overall, the results suggest that non-linear regression models such as decision trees, random forests, and gradient boosting may be more suitable for predicting the price of used cars than linear or regularized regression models. However, the choice of the best model depends on the specific requirements of the task, such as the trade-off between model complexity and prediction accuracy.

3.3 Model Analysis

In this section, we will provide a summary of the performance of each model that we evaluated in our study. The models were evaluated using three metrics: root mean squared error (RMSE), R-squared (R2), and mean absolute error (MAE). The models were trained on two different feature sets: feature set 1 (df_f1) and feature set 2 (df_f2).

Linear Regression: We trained linear regression models on both feature sets and achieved similar results. The RMSE for both models were around 0.4, indicating that the model's predictions were, on average, about 0.4 units away from the true value. The R2 value was around 0.83-0.85, indicating that the model explains about 83-85% of the variance in the target variable. The MAE for both models was around 0.3.

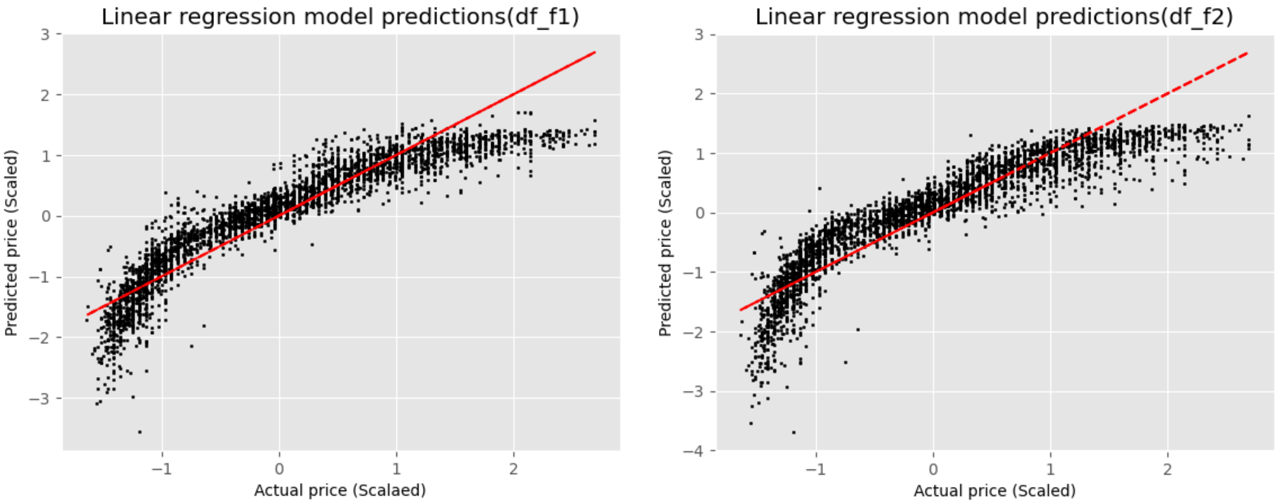

Figure 4: True vs. predicted price plot (Linear Regression for two feature set)

Polynomial Regression: We also trained polynomial regression models on both feature sets. The RMSE for the df_f1 model was 0.27, which was lower than the linear regression model. The R2 value was higher at around 0.92, indicating that the polynomial model better explained the variance in the target variable. The MAE for the df_f1 model was 0.19. The results for the df_f2 model were similar, with an RMSE of 0.3, an R2 value of 0.91, and an MAE of 0.21.

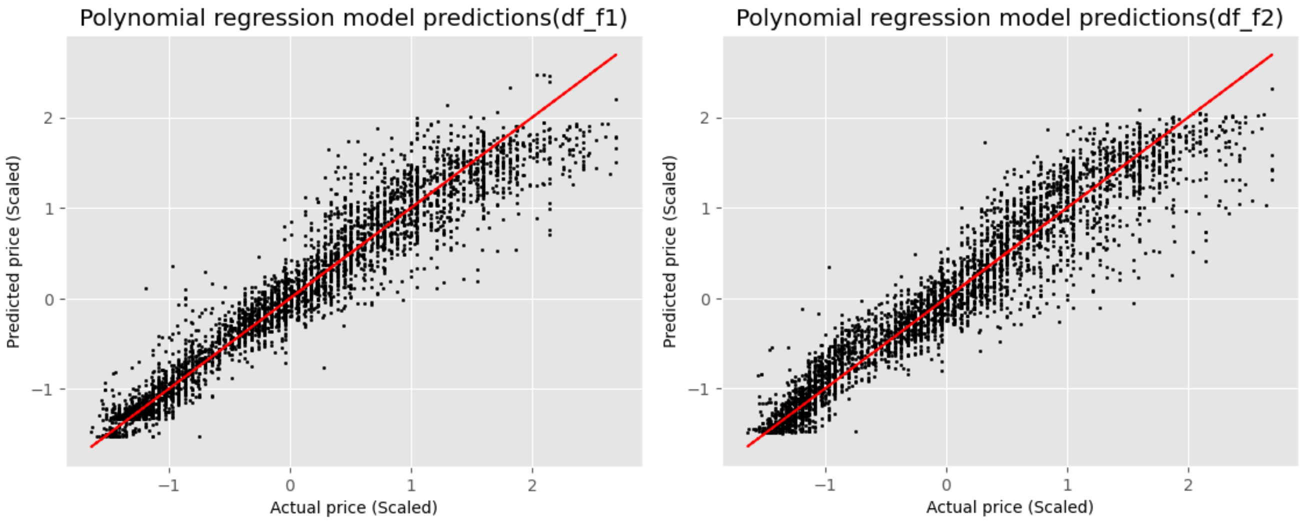

Figure 5: True vs. predicted price plot (Polynomial Regression for two feature set)

Ridge Regression and Lasso Regression: We trained ridge and lasso regression models on feature sets 1 and 2. The results were like those of the linear regression models, with RMSE values of around 0.38-0.41, R2 values of around 0.83-0.85, and MAE values of around 0.29-0.31.

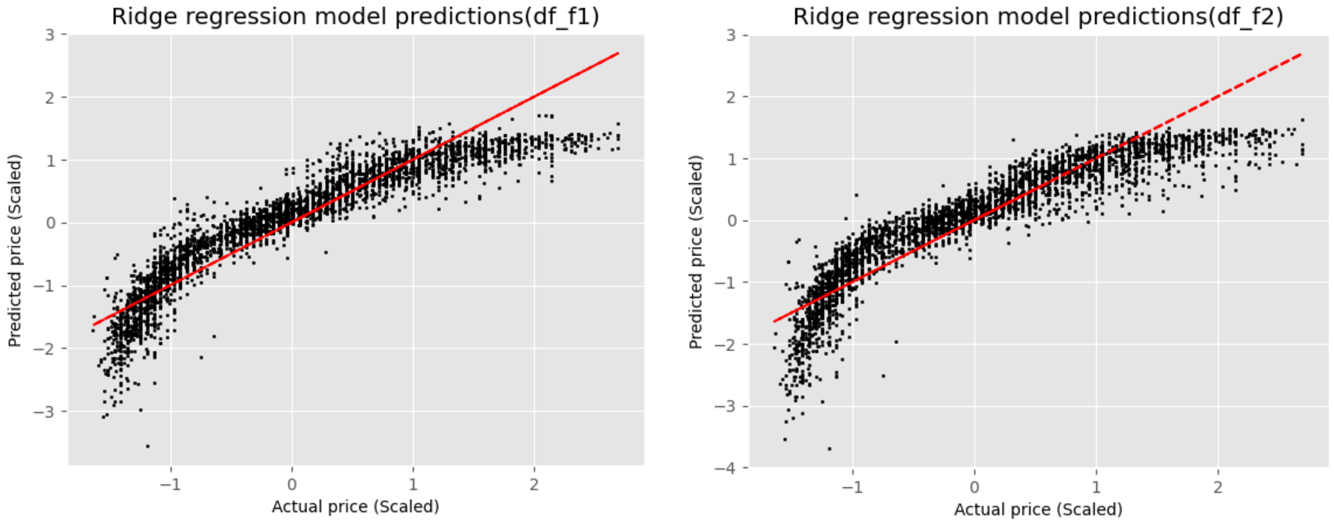

Figure 6: True vs. predicted price plot (Ridge Regression for two feature set)

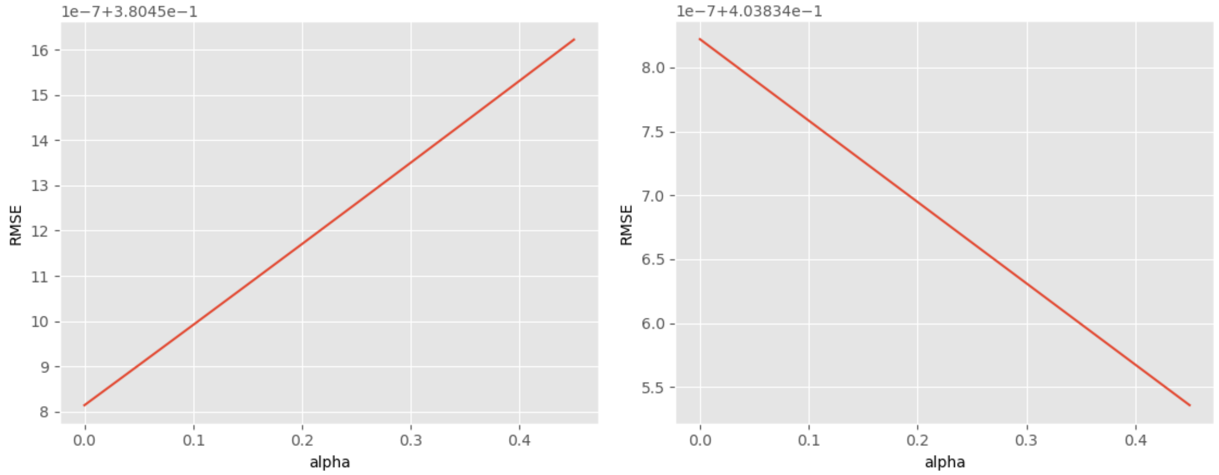

Figure 7: Ridge hyperparameter (alpha) vs RMSE. Left panel: feature set df_f1, right panel: feature set df_f2

The Ridge hyperparameter (alpha) vs RMSE plots show the effect of changing the hyperparameter alpha on the root mean squared error (RMSE) of the Ridge regression models for two different feature sets, df_f1 and df_f2.

In Figure 7, left panel for feature set df_f1, as alpha increases, the RMSE of the model also increases in an ascending line. This indicates that as we increase the regularization strength of the model by increasing alpha, the model becomes more constrained and less complex, leading to a higher error rate. This is expected since high regularization can lead to underfitting, where the model is too simple and cannot capture the complexity of the data.

On the other hand, the right panel for feature set df_f2 shows a descending line where the RMSE of the model decreases as alpha increases. This is because feature set df_f2 is more complex than df_f1, and higher regularization can help to prevent overfitting, where the model becomes too complex and captures noise in the data. As a result, increasing alpha can help to improve the model's performance.

Overall, these plots highlight the importance of selecting the appropriate hyperparameters for a given model and feature set to achieve the best performance.

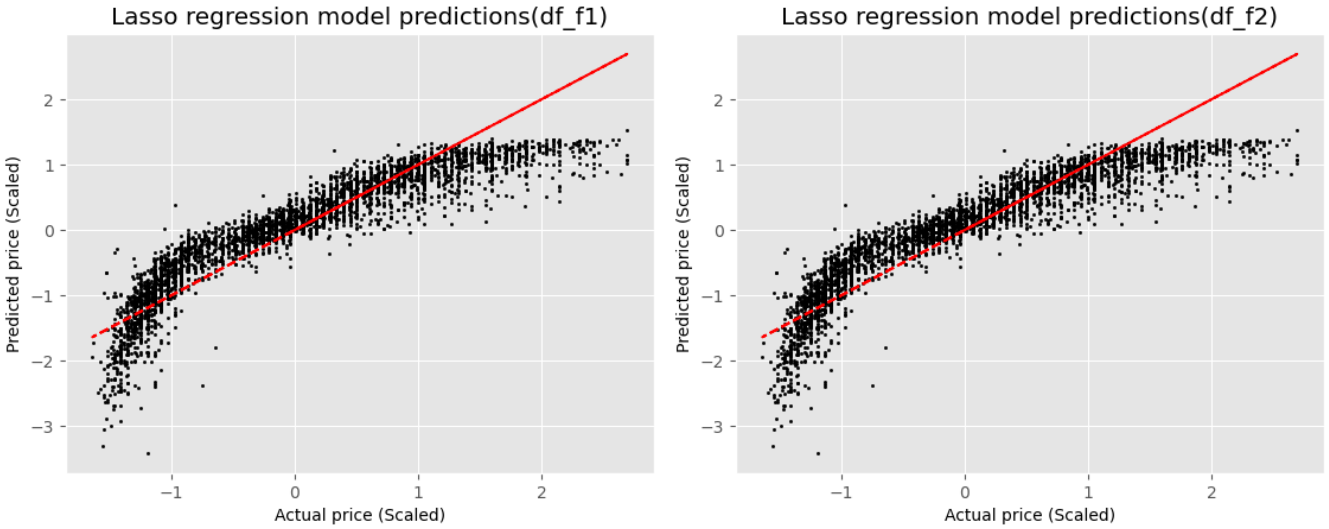

Figure 8: True vs. predicted price plot (Lasso Regression for two feature set)

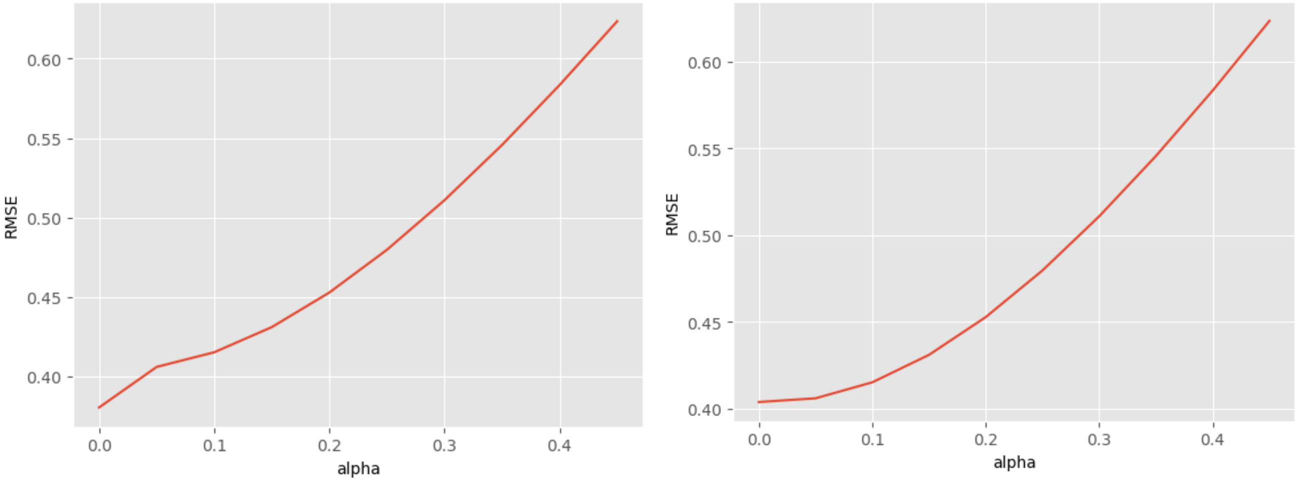

Figure 9: Lasso hyperparameter (alpha) vs RMSE. Left panel: feature set df_f1, right panel: feature set df_f2

In Figure 9 the left panel of the plot shows the effect of the hyperparameter alpha on the Lasso regression model's RMSE using feature set df_f1. As the value of alpha increases, the model's RMSE also increases in an ascending linear fashion. This suggests that as the regularization strength (controlled by alpha) increases, the model becomes more constrained and less flexible, leading to an increase in the error.

On the other hand, the right panel of the plot shows the effect of the hyperparameter alpha on the Lasso regression model's RMSE using feature set df_f2. The RMSE initially increases as the value of alpha increases, which is like the behavior of Ridge regression. However, after a certain point, the RMSE starts decreasing again, forming a quadratic curve. This suggests that as the regularization strength increases, the model becomes more constrained, but the selected features become more important, and the model starts performing better again.

The difference in the behavior of Ridge and Lasso regression models can be explained by their regularization techniques. Ridge regression applies L2 regularization, which shrinks the coefficients towards zero, but does not set them to zero. Lasso regression, on the other hand, applies L1 regularization, which can set some coefficients to exactly zero, effectively performing feature selection. Therefore, as the value of alpha increases in Ridge regression, all the coefficients are still included, but just become smaller. In contrast, as the value of alpha increases in Lasso regression, some coefficients can become exactly zero, leading to a sparser model with fewer features.

Nonlinear Regression Models: We also evaluated several nonlinear regression models on feature set 2. The results were generally better than those of the linear and polynomial regression models. The decision tree model achieved an RMSE of 0.28, an R2 value of 0.92, and an MAE of 0.19. The random forest, gradient boosting, XGBoost, LightGBM, and neural network models all achieved similar results, with RMSE values of around 0.26, R2 values of around 0.93, and MAE values of around 0.18. The support vector machine model achieved the best performance, with an RMSE of 0.26, an R2 value of 0.93, and an MAE of 0.17.

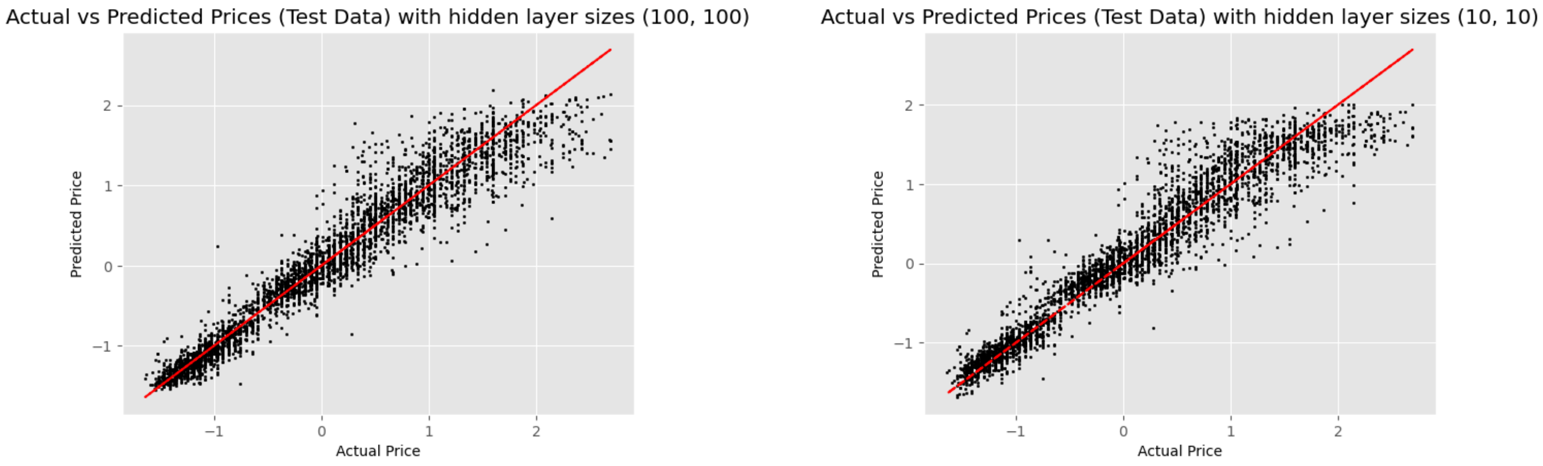

Figure 10: True vs. predicted price plot (Neural Network (100,100) for two feature set)

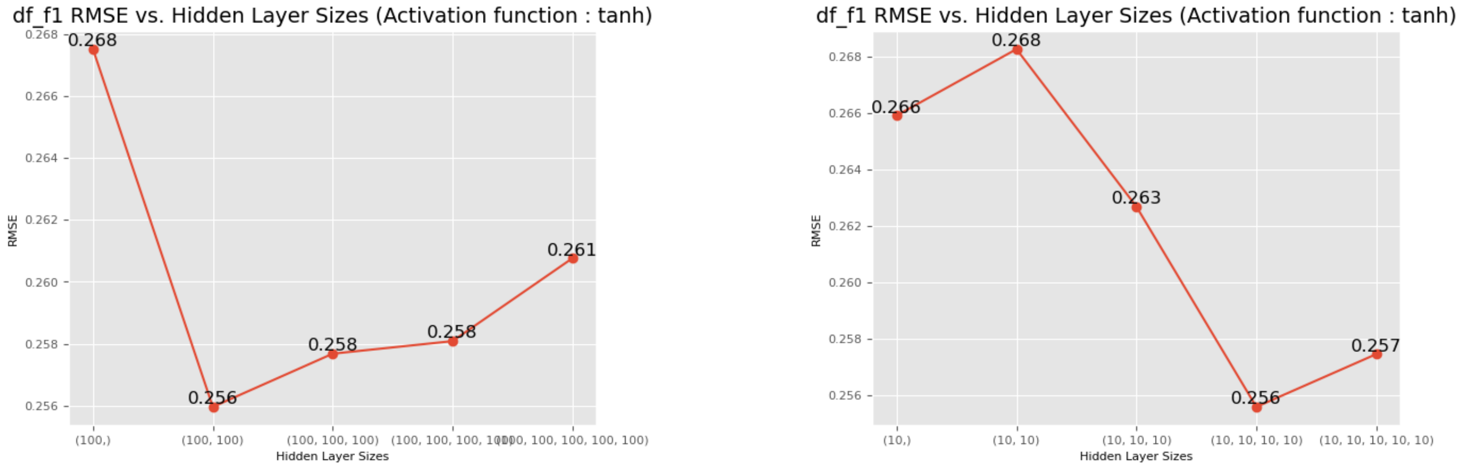

Figure 11: Neural network hidden layer sizes vs. RMSE (activation function:tanh)

In the experiment, we plotted the Neural Network hidden layer sizes against the corresponding RMSE values for the activation function tanh (Figure 11). Specifically, we examined five different hidden layer size configurations, in the left panel and in the right panel. Interestingly, the results revealed that the configurations (100,100) from left panel and (10,10,10,10) from right panel exhibited similar RMSE values of 0.256, implying that increasing the hidden layer size beyond a certain threshold may not necessarily lead to improved performance. This observation may be attributed to the possibility of overfitting, where the model may be fitting the training data too closely and not generalizing well to new data. Overall, these findings suggest that careful consideration should be given to selecting the appropriate hidden layer size configuration for Neural Network models, as overly complex architectures may not necessarily improve model performance.

Overall, the nonlinear regression models generally outperformed the linear and polynomial regression models, with the support vector machine model achieving the best performance. The results suggest that the most important features for predicting the price of a used car are the car's mileage, year, and trim level.

Note: A thorough investigation was conducted for each model with its corresponding hyperparameters, and the findings are documented in the accompanying Python notebook. Due to the extensive nature of the results, it was not feasible to include all of them, and only the most critical outcomes were highlighted for the purposes of brevity.

Conclusions

In this project, we compared the performance of various regression models on a used car dataset. We applied linear regression, ridge regression, lasso regression, decision tree, random forest, gradient boosting, x-gradient boosting, light gradient boosting, support vector machine, and neural networks to the dataset, and evaluated the performance of each model in terms of RMSE, MAE, and R-squared.

Our results showed that some models performed better than others. Specifically, the neural network model achieved the lowest RMSE and MAE, and the highest R-squared, indicating that it outperformed the other models. However, it also required more computational resources and tuning of hyperparameters. On the other hand, the linear regression model, which is a simple and interpretable model, performed reasonably well and could be a good choice when the computational resources are limited.

In addition, we found that the performance of some models was sensitive to hyperparameters, such as the maximum depth and minimum samples split in decision tree models, and the regularization parameter in ridge and lasso regression models. Therefore, it is important to carefully tune the hyperparameters of these models to achieve the best performance.

Lessons Learned

Through this project, we learned several important lessons:

- Data preprocessing is a crucial step in regression analysis and can significantly affect the performance of the models. It is important to handle missing values, outliers, and categorical variables appropriately.
- Different regression models have different strengths and weaknesses, and their performance can vary depending on the dataset and the problem. It is important to try different models and compare their performance to choose the best one for a given problem.
- Hyperparameters can significantly affect the performance of some models, and it is important to tune them carefully using cross-validation or other techniques.
- Neural networks can achieve very good performance on regression problems, but they also require more computational resources and tuning of hyperparameters.

Limitations

While this study aimed to compare various regression models to predict the price of used cars, there are several limitations to consider.

First, the dataset used for this study was limited to a specific geographic region and time period. Therefore, the models may not generalize well to other regions or time periods. Additionally, the dataset only includes information on used cars, and it may be difficult to apply these models to new car sales.

Second, while various preprocessing techniques were applied to the data, there may be other factors that were not captured in the dataset that could impact car prices, such as the specific condition of the car or the location of the seller.

Third, while this study compared various regression models, there are many other machine learning algorithms that could potentially improve the accuracy of the predictions. Additionally, the hyperparameters used for each model were determined through a manual search and may not be optimal.

Despite these limitations, this study provides valuable insights into the performance of various regression models for predicting the price of used cars. Future research could explore additional factors that may impact car prices, such as specific car features or demographic information of the sellers and could further compare the performance of different machine learning algorithms.

References

[1] Robbert McCulloch webpage.
[2] Used Car Dataset.