جامعة بيرزيت

BIRZEIT UNIVERSITY

Faculty of Engineering & Technology Electrical & Computer Engineering Department

Machine Learning and Data Science - ENCS5341

Assignment 2

Regression Analysis and Model Selection

Prepared by: QOSSAY RIDA 1211553

MAI BEITNOBA 1210260

Instructor: Ismail Khater

Date: 22/11/2024

Abstract

This assignment focuses on predicting car prices using data from YallaMotors website. The dataset includes key features like engine capacity, horsepower, top speed, and brand. Various regression models, including Linear Regression, LASSO, Ridge Regression and Polynomial Regression, were used to explore how these features influence car prices. To improve the models' accuracy and reliability, techniques like feature selection and hyperparameter tuning were applied, ensuring precise predictions while minimizing the risk of overfitting.

Table of Contents

Abstract...I
Table of Contents II
Table of figures IV
List of Tables V
Introduction..1
Data Preprocessing..2
Regression with numerical features only..7
Gradient Decent Regression..7
Closed form solution 7
LASSO Regression..8
Ridge Regression..9
Polynomial Regression 11
Degree 2..11
Degree 3..12
Degree 4..13
Degree 5..13
Degree 6 and Above 14
Conclusion 16
Radial Basis Function (RBF) Kernel Regression..16
Feature Selection Using Forward Selection..16
Comparison of All Models..17
Regression with All features..19
Gradient Decent Regresson..19
Closed Form Solution..20
LASSO Regression..20
Ridge Regression..21
Polynomial Regression 23
Radial Basis Function..26
Feature Selection with Forward Selection 26
Comparison of all models 27

Closed-form solution 28

Gradient Descent 28

LASSO.. 29

Ridge Regression.. 30

Polynomial Regression 31

Radial Basis Function.. 34

Feature Selection with Forward Selection 35

Comparison Between All Models 35

Conclusion 36

Table of figures

Figure 1:Price Vs Seats scatter plot 2
Figure 2:Price Vs top speed scatter plot 3
Figure 3:Price Vs top speed scatter plot - outliers removed 3
Figure 4:Price Vs Cylinder scatter plot 4
Figure 5:Price Vs horse power scatter plot 4
Figure 6:Price Vs horse power scatter plot - outliers removed 4
Figure 7:Price Vs Engine Capacity scatter plot 5
Figure 8:Price Vs Engine Capacity scatter plot - outliers removed 5
Figure 9:LASSO - Prediction Vs Actual values 9
Figure 10:Ridge - Prediction Vs Actual values 11
Figure 11:Polynomial Degree 2 - Prediction Vs Actual values 12
Figure 12:Polynomial Degree 3 - Prediction Vs Actual values 12
Figure 13:Polynomial Degree 4 - Prediction Vs Actual values 13
Figure 14:Polynomial Degree 5 - Prediction Vs Actual values 14
Figure 15:Polynomial Degree 6 - Prediction Vs Actual values 15
Figure 16:Polynomial Degree 7 - Prediction Vs Actual values 15
Figure 17:Polynomial Degree 8 - Prediction Vs Actual values 15
Figure 18:Polynomial Degree 9 - Prediction Vs Actual values 15
Figure 19:Polynomial Degree 10 - Prediction Vs Actual values 15
Figure 20:LASSO - Prediction Vs Actual values 21
Figure 21:Ridge - Prediction Vs Actual values 22
Figure 22:Polynomial Degree 2 - Prediction Vs Actual values 23
Figure 23:Polynomial Degree 3 - Prediction Vs Actual values 24
Figure 24:Polynomial Degree 4 - Prediction Vs Actual values 25
Figure 25:Polynomial Degree 5 - Prediction Vs Actual values 25
Figure 26:Polynomial Degree 6 - Prediction Vs Actual values 25
Figure 27:Polynomial Degree 7 - Prediction Vs Actual values 25
Figure 28:Polynomial Degree 8 - Prediction Vs Actual values 25
Figure 29:Polynomial Degree 9 - Prediction Vs Actual values 25
Figure 30:Polynomial Degree 10 - Prediction Vs Actual values 26
Figure 31:LASSO - Actual Vs Predicted 30
Figure 32: Ridge - scatter plots 31
Figure 33:Polynomial Degree 2 - Prediction Vs Actual values 33
Figure 34:Polynomial Degree 3 - Prediction Vs Actual values 33
Figure 35:Polynomial Degree 4 - Prediction Vs Actual values 33
Figure 36:Polynomial Degree 5 - Prediction Vs Actual values 33
Figure 37:Polynomial Degree 6 - Prediction Vs Actual values 33
Figure 38:Polynomial Degree 7 - Prediction Vs Actual values 33
Figure 39:Polynomial Degree 8 - Prediction Vs Actual values 34
Figure 40:Polynomial Degree 9 - Prediction Vs Actual values 34
Figure 41:Polynomial Degree 8 - Prediction Vs Actual values 34

List of Tables

Table 1:Gradient Descent Coefficients 7

Table 2:closed form coefficients 8

Table 3:LASSO coefficients 9

Table 4:Ridge Coefficients 10

Table 5: All models Comparison 17

Table 6:Gradient Descent Coefficients 20

Table 7:closed form Coefficients 20

Table 8:LASSO coefficients 21

Table 9:Ridge coefficients 22

Table 10: All Models Comparison 27

Table 11:closed form coefficients 28

Table 12:Gradient descent coefficients 29

Table 13:LASSO coefficients 29

Table 14:Ridge coefficients 31

Table 15: Polynomial Regression Metrics 32

Table 16: All models Comparison 35

Introduction

This assignment focuses on predicting car prices by analyzing the relationships between various car features using regression models. The dataset, sourced from YallaMotors, contains 6,308 entries with nine features, including engine capacity, number of cylinders, horsepower, top speed, seating capacity, brand, and country.

Data preprocessing was an essential step. Price values recorded in multiple currencies were standardized, missing data was handled carefully, and categorical variables like brand and country were encoded for compatibility with machine learning algorithms. The models' performance was also evaluated both with and without these categorical features.

A variety of regression models, both linear and nonlinear, were applied to study how these features impact car prices. Feature selection and regularization techniques were used to improve model accuracy while keeping the models efficient and interpretable. Additionally, regression models were developed to predict an alternative feature, Engine Capacity, for a deeper understanding of the data.

Data Preprocessing

We started the data cleaning process with 6,308 rows and 9 columns, as detailed below:

```
#   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   car name         6308 non-null   object
 1   price            6308 non-null   object
 2   engine_capacity  6308 non-null   object
 3   cylinder         5684 non-null   object
 4   horse_power      6308 non-null   object
 5   top_speed        6308 non-null   object
 6   seats            6308 non-null   object
 7   brand            6308 non-null   object
 8   country          6308 non-null   object
```

The data consisted entirely of object types, requiring significant preprocessing.

Starting with the Seats column, which initially had values like “8 Seater.” We simplified this by removing the text and keeping just the number. If a value in this column didn’t follow the expected format, it was replaced with NaN. To handle missing values, we calculated the mean number of seats and used it to fill in the gaps.

To better understand this feature, we also plotted the relationship between car prices and the number of seats, as shown in the figure below.

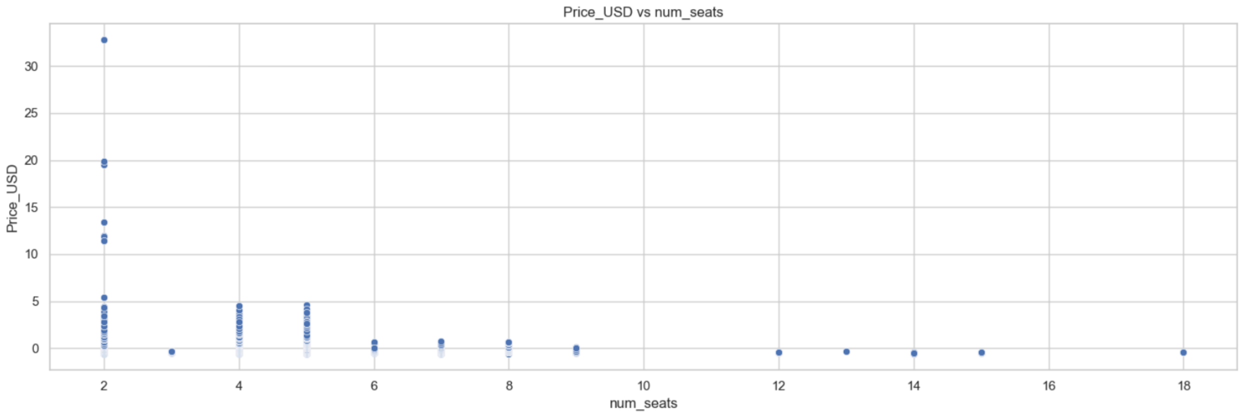

Figure 1:Price Vs Seats scatter plot

We applied a similar cleaning process to the Top Speed column. Initially, the values in this column were in a non-numeric format, so we converted them to numeric values. Any missing entries were filled with the mean of the Top Speed feature. Afterward, we ensured all values were consistent by converting them to integers. To better understand the relationship between top speed and car prices, we visualized this feature in a plot.

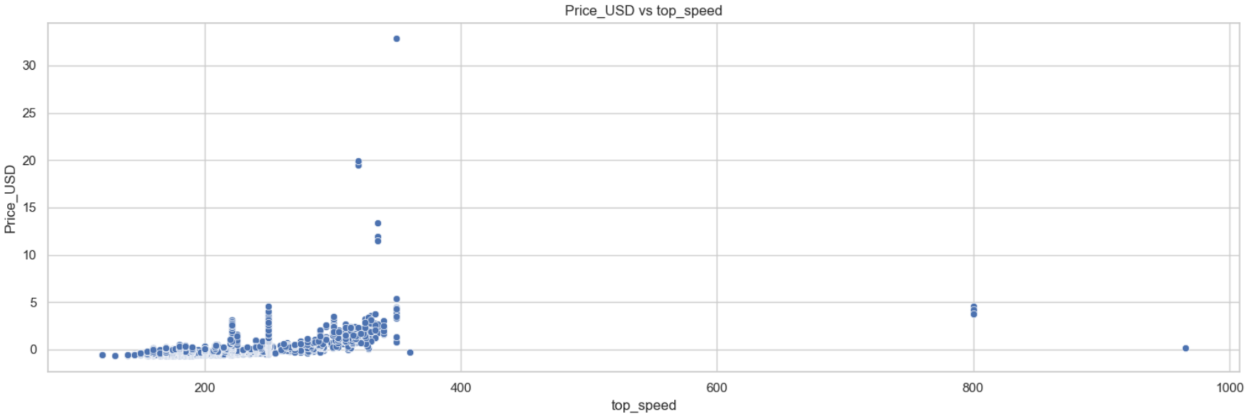

Figure 2:Price Vs top speed scatter plot

The initial plot revealed some outliers in the Top Speed feature, which could distort the analysis. To address this, we identified and removed outliers using the interquartile range (IQR) method. Specifically, values falling below Q1 − 1.5 × IQR or above Q3 + 1.5 × IQR were considered outliers and excluded. After removing these extreme values, we re-plotted the data. The revised plot showed a more reliable distribution, enhancing the quality of our analysis.

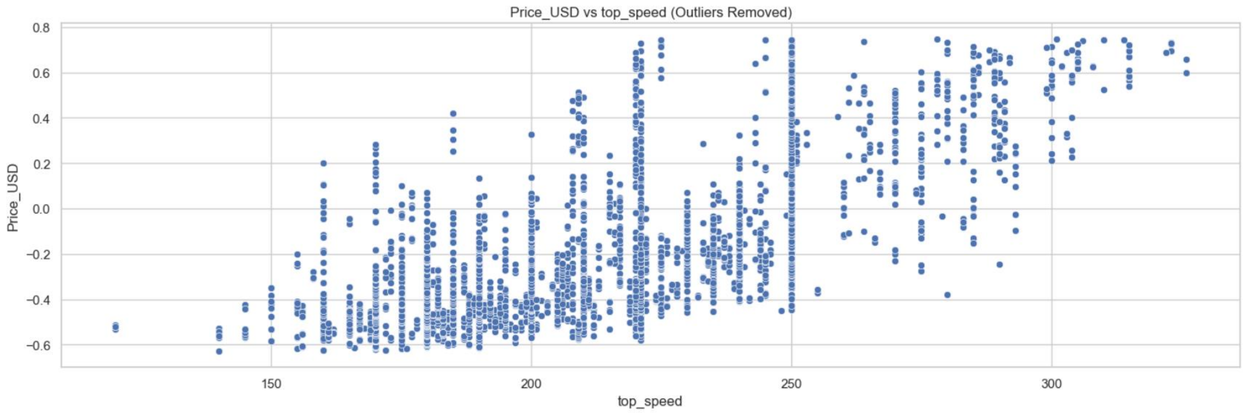

Figure 3:Price Vs top speed scatter plot - outliers removed

Next, we processed the Cylinder feature to ensure it was suitable for analysis. First, we converted the values in this column to numeric format. Then, we calculated the mean of the feature to handle missing values. Since the number of cylinders typically falls into discrete categories (4, 6, 8, or 12), we rounded the computed mean to the nearest of these standard values.

The missing values were then filled with this adjusted number, and all entries in the column were converted to integers for consistency. Finally, we plotted the Cylinder feature to visualize its distribution and identify its relationship with car prices, as shown below.

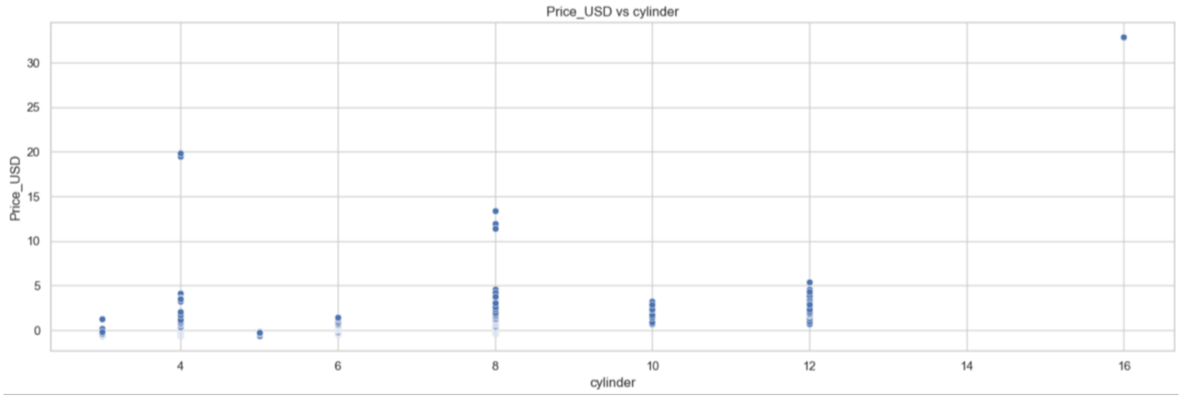

Figure 4:Price Vs Cylinder scatter plot

We then processed the Horsepower feature by converting its values to numeric format. To handle missing data, we calculated the mean of the feature and used it to fill any gaps. Afterward, the entire column was converted to integers for consistency.

When plotting the Horsepower feature, we noticed some outliers that could affect the model's performance. To address this, we applied the Interquartile Range (IQR) method to identify and remove these outliers. The cleaned data was then re-plotted, resulting in a more consistent and reliable distribution, as shown below.

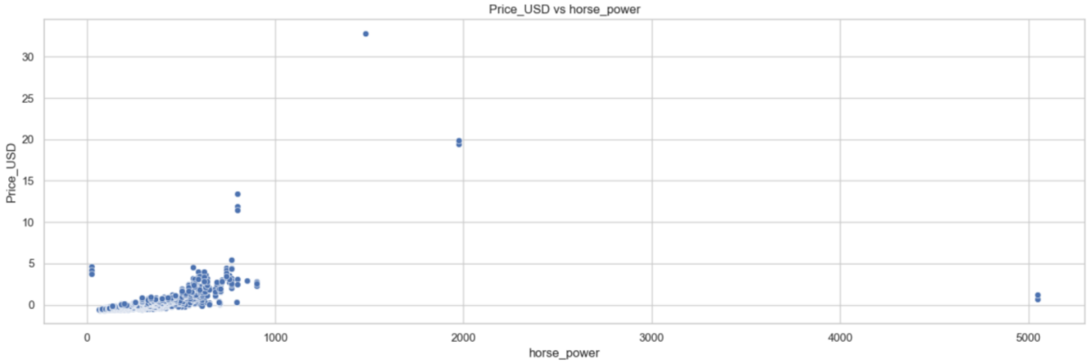

Figure 5:Price Vs horse power scatter plot

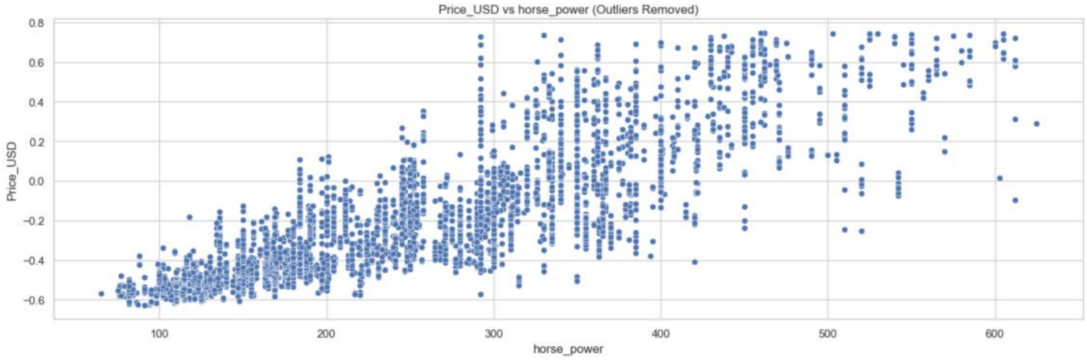

Figure 6:Price Vs horse power scatter plot - outliers removed

The Engine Capacity feature went through several preprocessing steps to ensure consistency and usability. First, we converted its values to numeric format. Some entries, which were recorded in smaller units, were standardized by dividing them by 1,000 to convert them into liters.

Next, we calculated the mean of the engine capacities and used it to fill any missing data. After ensuring completeness, the entire column was converted to integers for uniformity.

When we plotted the feature, we identified some outliers that could skew the analysis. Using the Interquartile Range (IQR) method, we detected and removed these outliers. The cleaned data was then re-plotted, resulting in a more consistent and meaningful distribution, as shown below.

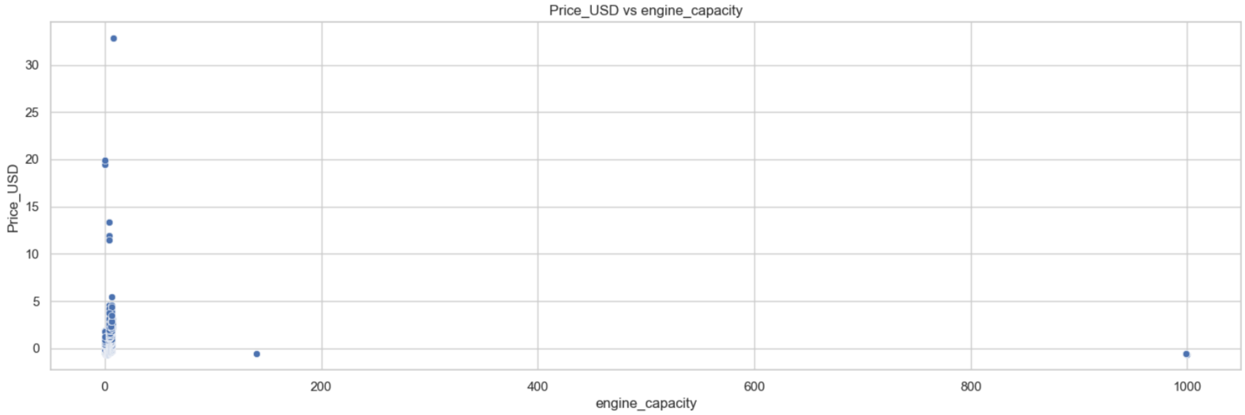

Figure 7:Price Vs Engine Capacity scatter plot

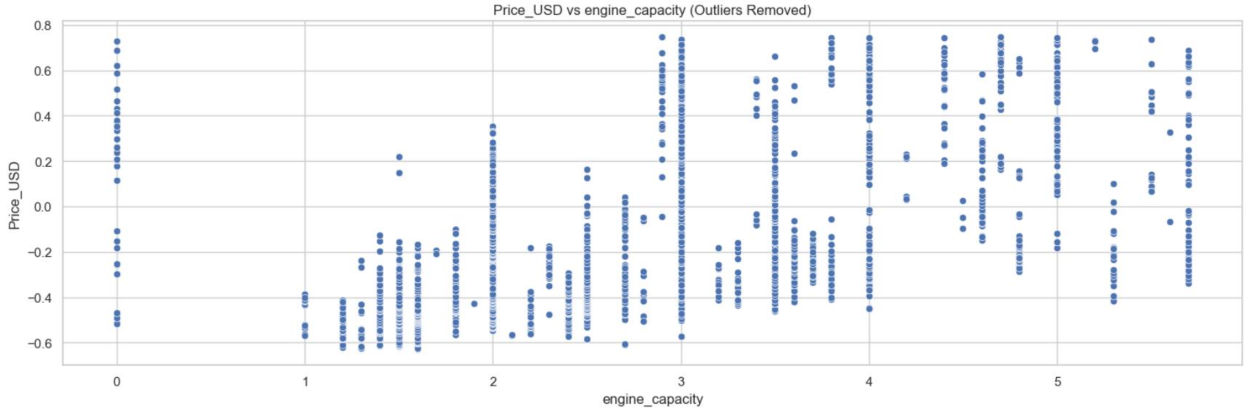

Figure 8:Price Vs Engine Capacity scatter plot - outliers removed

The Price column required extensive preprocessing to standardize and clean the data. Each entry in this column included both the currency and the amount, so we began by extracting the currency from the data. Rows where the amount was missing were removed since approximately 1,000 entries lacked this crucial information, accounting for a significant portion of the dataset. This step ensured that only valid price data remained for analysis.

To create consistency, we converted all prices into a common currency, USD, using the appropriate exchange rates for the extracted currencies. Once standardized, we applied z-score normalization to the price values, transforming them into a normalized scale for better comparability during the machine learning process.

After cleaning the dataset, we removed redundant columns to ensure that only the relevant features remained for the regression models. Specifically, we dropped the 'Price', 'Currency', 'Amount', and 'Seats' columns, as they were no longer needed after preprocessing.

With these adjustments, the dataset is now ready for regression analysis. The final dataset includes 9 columns, 4,425 rows in total, with all columns being non-null.

```
Data columns (total 9 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   car name         4425 non-null   object 
 1   engine_capacity  4425 non-null   float64
 2   cylinder         4425 non-null   int64  
 3   horse_power      4425 non-null   int64  
 4   top_speed        4425 non-null   int64  
 5   brand            4425 non-null   object 
 6   country          4425 non-null   object 
 7   num_seats        4425 non-null   int64  
 8   Price_USD        4425 non-null   float64
```

Regression with numerical features only

In this section, we focus on building regression models using only the numerical features of the dataset, excluding categorical variables such as brand and country. The dataset consists of five numerical predictors: engine capacity, cylinder count, horsepower, top speed, and number of seats. These features were standardized to ensure uniform scaling, making the models less sensitive to varying units and magnitudes.

Gradient Decent Regression

Gradient Descent was used to optimize the coefficients of a linear regression model for the dataset. This method aimed to reduce the mean squared error (MSE) through a step-by-step process. The algorithm started with all coefficients, including a bias term, set to zero. These values were updated repeatedly using a learning rate of 0.01 over 1,000 steps (epochs). In each step, batch gradient descent was applied, meaning the entire training set was used to calculate the gradient. The gradient was computed as the average product of the training data's transpose and the differences between predicted and actual values (residuals). The coefficients were updated by subtracting the scaled gradients, slowly moving toward the best solution.

The model's performance was measured using Mean Squared Error (MSE), Mean Absolute Error (MAE), and R^2 scores on both the validation and test datasets. On the validation set, the Gradient Descent model achieved an MSE of 0.025, an MAE of 0.116, and an R^2 score of 0.747, showing a good fit. On the test set, the model performed slightly better, with an MSE of 0.022, an MAE of 0.110, and an R^2 of 0.781. These results show the model can work well with new, unseen data.

The final coefficients obtained through Gradient Descent were:

Table 1:Gradient Descent Coefficients

Bias Term	Engine Capacity	Cylinder Count	Horsepower	Top Speed	Number of Seats
-0.19666118	-0.0114911	0.05086498	0.16568283	0.10901287	0.00350892

These coefficients reveal how each numerical feature affects the predicted price. Horsepower and top speed had a relatively stronger positive impact compared to the other features.

The Gradient Descent method proved to be an effective way to optimize the model. It achieved good results while keeping the model easy to understand.

Closed form solution

The closed form solution, was used to calculate the best coefficients for the linear regression model. This method directly solves the regression equation without needing an iterative process. Using matrix algebra, the closed form solution finds the coefficients that minimize the mean squared error (MSE) using the formula $\theta = (X^T X)^{-1} X^T y$. Where, X represent the feature matrix (including a bias term), and y is the target variable.

This approach was applied to the training data, calculating the coefficients in a single step. The model then made predictions on the validation and test datasets to measure its performance.

The closed form solution gave strong results on both validation and test datasets. On the validation set, the model had an MSE of 0.024, an MAE of 0.115, and an R^2 of 0.751, showing a good fit. On the test set, the results were slightly better, with an MSE of 0.022, an MAE of 0.109, and an R^2 of 0.785. These results suggest that the model performs well on unseen data.

The coefficients calculated using the closed form solution were:

Table 2:closed form coefficients

Bias Term	Engine Capacity	**Cylinder Count**	**Horsepower**	**Top Speed**	**Number of Seats**
-0.19666967	-0.03180407	0.05975642	0.18505718	0.09629585	0.00241054

As with Gradient Descent, horsepower and top speed showed a stronger positive effect compared to other features.

The closed form solution is a straightforward and efficient approach, especially for smaller datasets. Unlike iterative methods, it doesn't require hyperparameter tuning or repeated updates. However, it can struggle with numerical instability if the feature matrix is singular or poorly conditioned. Fortunately, in this case, the dataset was well-structured, allowing for accurate coefficient calculation.

LASSO Regression

LASSO regression, which applies L_1 regularization, effectively shrinks some coefficients to zero, making it a powerful tool for feature selection. To find the optimal regularization parameter (λ), we used GridSearchCV with the following values for λ: [0.001,0.01,0.1,1,10,100].

We performed 5 fold cross validation using negative mean squared error (MSE) as the scoring metric. The best value of λ was found to be 0.001, indicating minimal shrinkage, which preserves critical features while reducing overfitting.

Using the optimal regularization parameter λ=0.001, the LASSO model was evaluated on the validation set, achieving a Mean Squared Error (MSE) of 0.024, a Mean Absolute Error (MAE) of 0.115, and an R^2score of 0.750. This indicates that the model explains 75% of the variance in the validation set, with low errors that reflect its accuracy. On the unseen test set, the model's performance was consistent with the validation results, further confirming its generalizability. The test set metrics included an MSE of 0.022, an MAE of 0.109, and an R^2score of 0.785.

The LASSO model's coefficients after L_1 regularization were as follows:

Table 3:LASSO coefficients

Engine Capacity	Cylinder Count	Horsepower	Top Speed	Number of Seats
−0.02296	0.05346	0.18176	0.09666	0.00036

These results reveal that features like Horsepower (0.18176) and Top Speed (0.09666) are the most influential, while features such as Number of Seats (0.00036) and Engine Capacity (-0.02296) were shrunk towards zero, indicating limited relevance.

The scatter plot further illustrates these results, showing predicted values plotted against actual target values for both the validation and test sets. The points closely follow the diagonal "Perfect Fit" line, with only minor deviations, demonstrating the accuracy of LASSO's predictions.

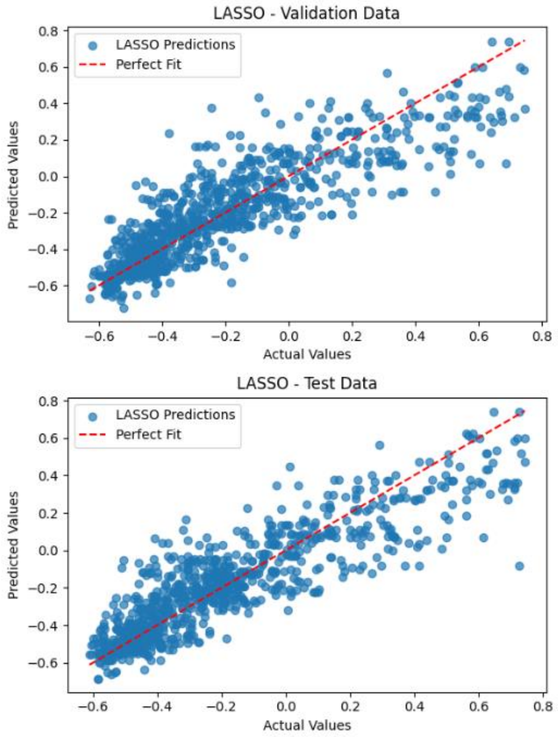

Figure 9:LASSO - Prediction Vs Actual values

LASSO regression delivers strong predictive performance while enhancing interpretability by selecting only the most important features. With an optimal λ of 0.001, it achieves a perfect balance between underfitting and overfitting, capturing complex patterns in a simple and easy to understand way.

Ridge Regression

Ridge regression, also known as L2 regularization, helps prevent overfitting by penalizing the sum of squared coefficients. Unlike LASSO, Ridge shrinks coefficients closer to zero but doesn't eliminate any features, making it ideal for situations where all features are expected to contribute to the target variable.

To determine the best regularization parameter (λ), we used GridSearchCV with a range of λ values from 0.001 to 100. After 5 fold cross validation using negative mean squared error (MSE) as the scoring metric, the optimal λ was found to be 10. This relatively high λ indicates strong regularization, which helps control large coefficients and reduces the effects of multicollinearity.

Using the optimal λ=10, the Ridge regression model was evaluated on the validation set, achieving a Mean Squared Error (MSE) of 0.024, a Mean Absolute Error (MAE) of 0.115, and an R² score of 0.751. These metrics show that Ridge regression explains 75.1% of the variance in the validation data, with low errors indicating robust predictive performance. On the unseen test set, the model maintained consistent accuracy, achieving an MSE of 0.022, an MAE of 0.109, and an R² score of 0.785.

The coefficients from the Ridge regression model after applying L2 regularization were:

Table 4:Ridge Coefficients

Engine Capacity	**Cylinder Count**	**Horsepower**	**Top Speed**	**Number of Seats**
-0.02962	0.05908	0.18273	0.09708	0.00225

These values show that features like Horsepower (0.18273) and Top Speed (0.09708) have the strongest influence on predictions, while others, like Engine Capacity (-0.02962) and Number of Seats (0.00225), contribute less but are still included in the model.

The scatter plot for Ridge regression further illustrates the model's performance by comparing the predicted vs. actual target values for the validation and test sets. The points align closely with the diagonal "Perfect Fit" line, indicating strong predictive accuracy and minimal error.

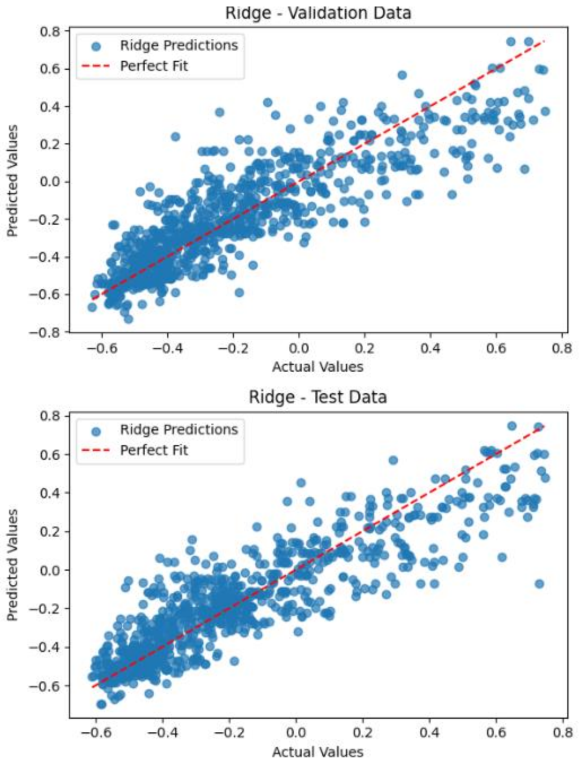

Figure 10:Ridge - Prediction Vs Actual values

Ridge regression achieves strong predictive accuracy by effectively managing multicollinearity and preserving all features. With an optimal λ of 10, it balances model complexity and generalization.

Polynomial Regression

Polynomial regression expands the feature space by introducing polynomial combinations of the input variables, enabling the model to capture more complex relationships. The degree of the polynomial determines the complexity of the model. The results for degrees ranging from 2 to 10 are discussed below, along with the associated scatter plots visualizing actual versus predicted values.

Degree 2

For polynomial regression of degree 2, the model performed well with validation metrics of MSE = 0.022**,** MAE = 0.107, and R^2 = 0.771, showing that it captured 77.1% of the variance in the validation data. The test metrics were similarly strong with MSE = 0.020**,** MAE = 0.102, and R^2 = 0.807, indicating good generalizability. The scatter plot showed a strong alignment of predicted values with actual values, confirming the model's accuracy.

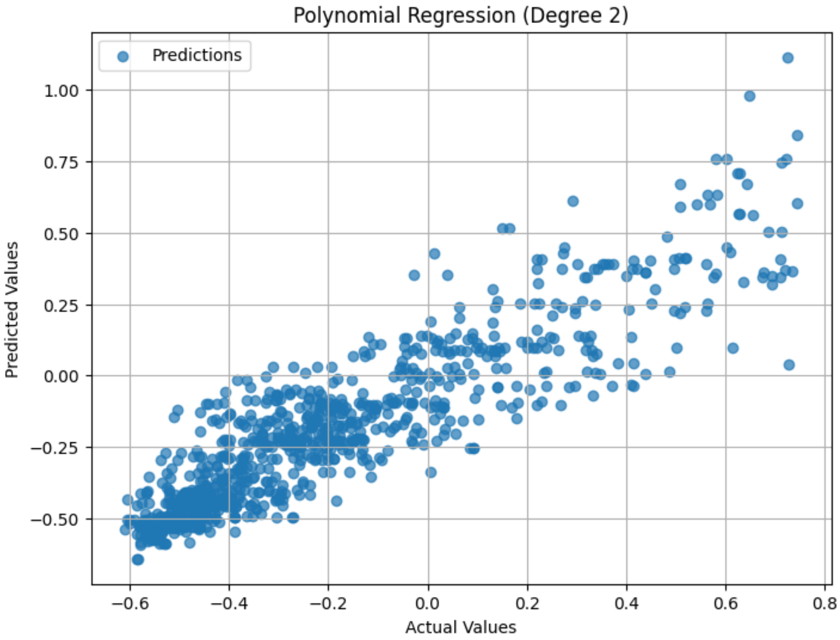

Figure 11:Polynomial Degree 2 - Prediction Vs Actual values

Degree 3

A degree 3 polynomial model achieved better validation metrics than degree 2, with MSE = 0.020**,** MAE = 0.103, and R² = 0.791, capturing 79.1% of the variance. However, test performance slightly declined with MSE = 0.028**,** MAE = 0.103, and R² = 0.728. The scatter plot revealed slightly more variance in predictions compared to degree 2, especially for extreme values.

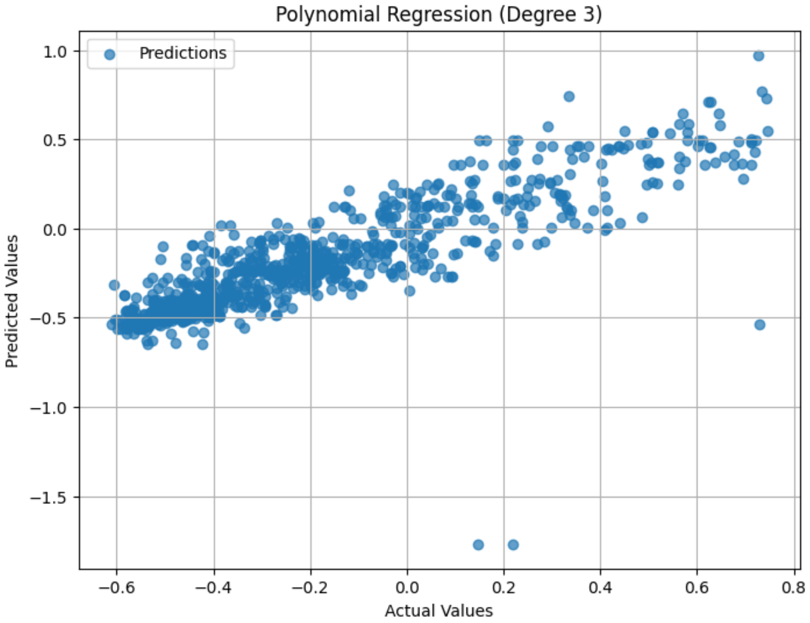

Figure 12:Polynomial Degree 3 - Prediction Vs Actual values

Degree 4

The degree 4 model improved both validation and test results. Validation metrics were MSE = 0.024, MAE = 0.099, and R² = 0.752, while the test metrics were MSE = 0.019**,** MAE = 0.095**,** and R² = 0.818, the best test performance across all models. The scatter plot demonstrated highly accurate predictions closely aligned with actual values, indicating that this degree effectively captured the data's underlying patterns.

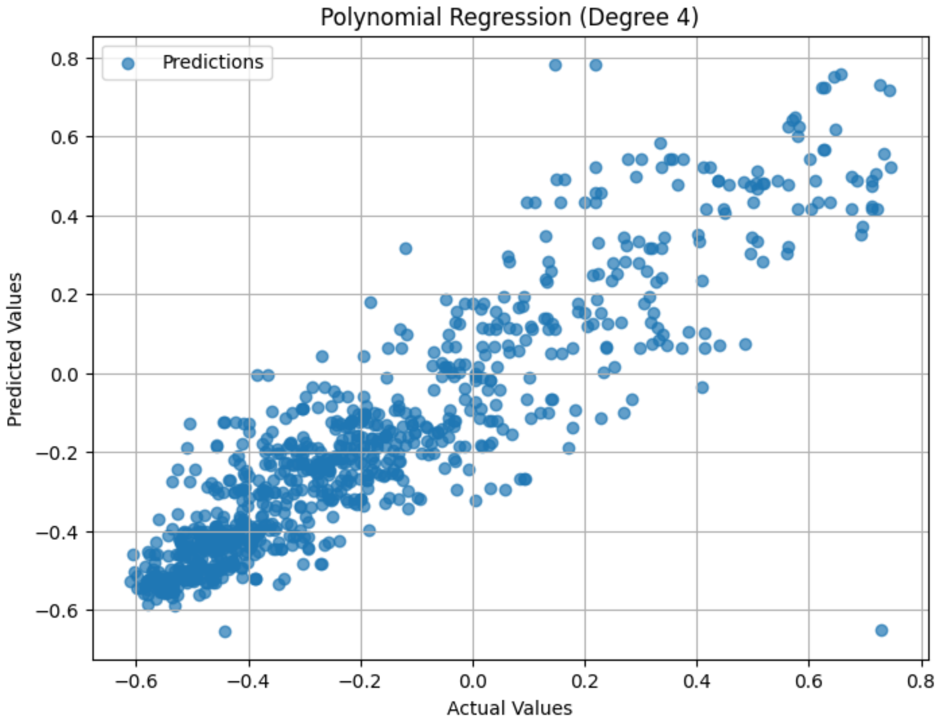

Figure 13:Polynomial Degree 4 - Prediction Vs Actual values

Degree 5

Performance dropped sharply with a degree 5 model. Validation metrics showed MSE = 0.234**,** MAE = 0.125, and R² = -1.399, indicating overfitting, as the model failed to generalize. Test results were even worse, with MSE = 0.524**,** MAE = 0.134, and R² = -4.100. The scatter plot demonstrates a noticeable deviation from the diagonal alignment seen in lower degree models, indicating poor predictions.

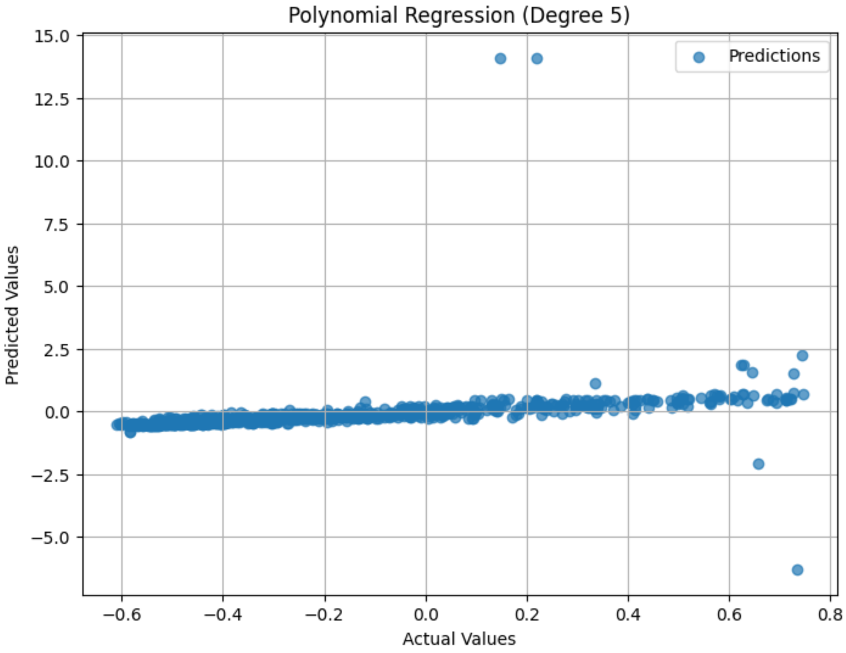

Figure 14:Polynomial Degree 5 - Prediction Vs Actual values

Degree 6 and Above

Starting from degree 6, the models showed clear signs of catastrophic overfitting, with both validation and test metrics.

For degree 6, the validation metrics were MSE = 16134.256**,** MAE **=** 7.080, and R² **= -** 165772**.**874, while the test metrics were MSE **=** 6156**.**370, MAE **=** 4.465, and R² **= -**59894.787. These results indicate that the model could no longer generalize to new data and failed to capture meaningful patterns.

As the degree increased to **7**, the performance worsened exponentially. Validation metrics reached MSE = 17663709741.232**,** MAE = 9221.398, and R² = -181488480969.764, with test metrics soaring to MSE = 56774483753.602**,** MAE = 16680.668**,** and R² = -552363233996.828. The predictions completely diverged from actual values, signaling extreme instability.

At degree 8, the metrics remained disastrous. Validation results were MSE = 3530094511.328**,** MAE = 5322.729, and R² = -36270494698.571, while test results were MSE = 44872510806.232**,** MAE = 11988.515, and R² = -436568041623.165. Despite the drop in scale, these values still reflect severe overfitting.

For degree 9, the instability continued, with validation metrics of MSE = 317469121041.960**,** MAE = 49600.162, and R² = -3261884925481.893, alongside test metrics of MSE = 218005928763.823**,** MAE = 32901.050, and R² = -2120996121518.926. The predictions were entirely disconnected from actual values, making the model useless for practical purposes.

Finally, at degree 10, the metrics reached their worst. The validation metrics were MSE = 9702524435931.209, MAE = 254286.458, and R² = -99690067786183.516, while test metrics skyrocketed to MSE = 5324157292875.734, MAE = 141386.286, and R² = -51799127815395.250. By this stage, the model had entirely broken down.

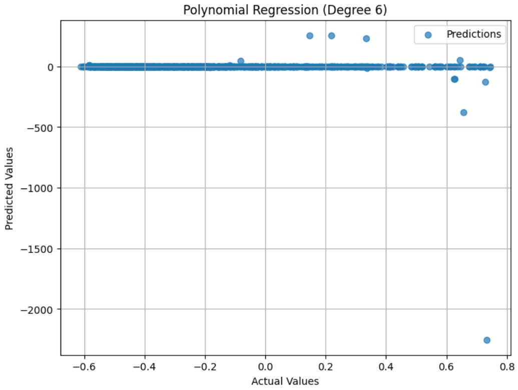

Figure 15:Polynomial Degree 6 - Prediction Vs Actual values

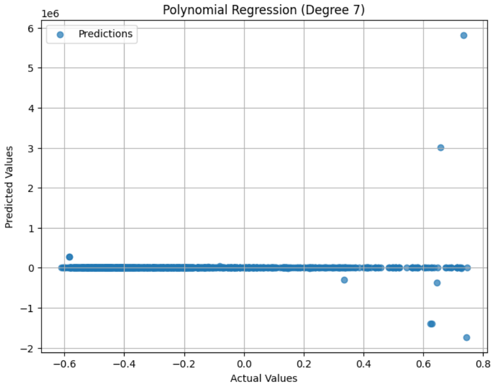

Figure 16:Polynomial Degree 7 - Prediction Vs Actual values

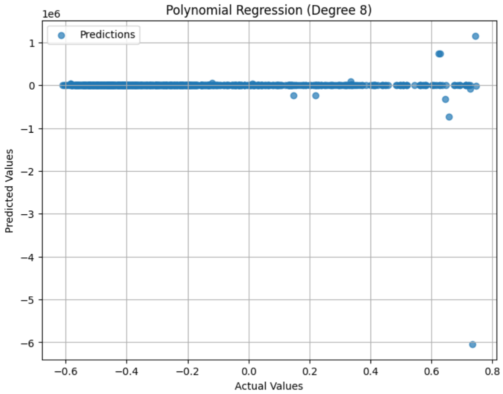

Figure 17:Polynomial Degree 8 - Prediction Vs Actual values

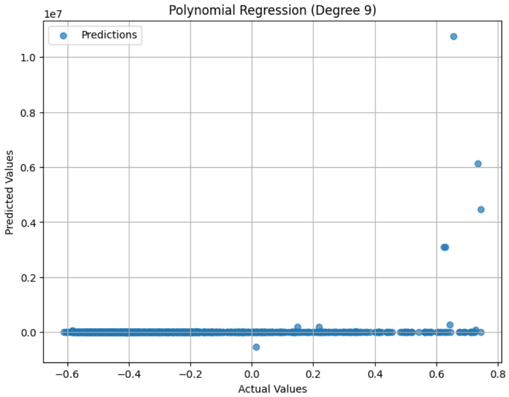

Figure 18:Polynomial Degree 9 - Prediction Vs Actual values

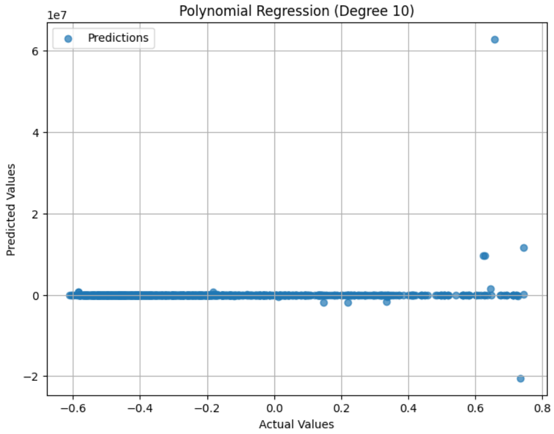

Figure 19:Polynomial Degree 10 - Prediction Vs Actual values

The scatter plots for these higher degree models confirmed the instability, with predictions scattered far from the actual values. This dramatic overfitting and numerical instability highlight the dangers of using overly complex polynomial models without proper regularization or validation.

Conclusion

Polynomial regression is a powerful tool for capturing nonlinear relationships in data, but its effectiveness depends heavily on selecting the appropriate degree of complexity. In this case, models with degrees 2 and 4 provided the best balance, achieving strong R² scores and low error metrics on both validation and test sets. These degrees effectively captured nonlinear patterns while maintaining good generalization.

However, higher degree models (5 and above) introduced excessive complexity, leading to overfitting. This was evident from their poor performance on validation and test data, with negative R² scores and scattered predictions that failed to align with the actual target values.

Radial Basis Function (RBF) Kernel Regression

The RBF Kernel Regression model was fine tuned using GridSearchCV with 5 fold cross validation. The optimization process explored different values for the hyperparameters C (0.1, 1, 10, 100) and gamma (0.01, 0.1, 1, 10). The best combination of parameters was identified as C = 10 and gamma = 1, providing a good balance between model complexity and regularization.

On the validation set, the optimized model performed impressively. It achieved a Mean Squared Error (MSE) of 0.019, a Mean Absolute Error (MAE) of 0.093, and an R² Score of 0.800. These metrics highlight the model's ability to capture the underlying patterns in the data accurately while maintaining low prediction errors.

When evaluated on unseen test data, the model continued to demonstrate strong predictive performance. It achieved an MSE of 0.015, an MAE of 0.089, and an R² Score of 0.853. Notably, the slightly better performance on the test set suggests that the model is robust and generalizes well to new data without overfitting.

In conclusion, the RBF Kernel Regression model is both accurate and reliable, excelling at capturing complex relationships in the data. Its robust performance on both validation and test sets makes it a highly effective choice for tasks that require precise predictions and strong generalizability.

Feature Selection Using Forward Selection

The process of feature selection was performed using forward selection to identify the most informative features for a Linear Regression model. The goal was to iteratively select features that maximized the model’s performance on the validation set, measured by the R² score. The process started with no features selected, considering all features as potential candidates. At each step, the algorithm evaluated the remaining features by temporarily adding each to the set of currently selected features. A Linear Regression model was trained with the resulting combination, and its performance was assessed by calculating the R² score on the validation set. The feature that provided the highest improvement in the R² score was added to the selected set.

The first feature selected was Feature 2, which achieved a validation R² of 0.6969, making the largest initial contribution to the model's performance. In the next iteration, feature 3 was selected, increasing the validation R² to 0.7373. This was followed by Feature 1, which brought the validation R² up to 0.7446. Finally, feature 0 was added, yielding a final validation R² of 0.7509. At this point, the maximum number of features (4) had been reached, and the selection process was concluded. The selected features were [2, 3, 1, 0].

A final Linear Regression model was then trained using only the selected features. When evaluated on the test set, the model achieved an R² score of 0.7855, demonstrating good generalization to unseen data. This result indicates that the forward selection method successfully identified a subset of features that effectively capture the relationships in the dataset while avoiding the inclusion of irrelevant or redundant features.

Comparison of All Models

The table below shows summary of the results, highlighting their performance on validation and test datasets.

Table 5: All models Comparison

Model	Validation R²	Validation MSE	Validation MAE	Test R²	Test MSE	Test MAE
Closed form Solution	0.751	0.024	0.115	0.785	0.022	0.109
Gradient Descent	0.747	0.025	0.116	0.781	0.022	0.110
LASSO	0.750	0.024	0.115	0.785	0.022	0.109
Ridge	0.751	0.024	0.115	0.785	0.022	0.109
Polynomial (Degree 3)	0.791	0.020	0.103	0.728	0.028	0.103
Polynomial (Degree 4)	0.752	0.024	0.099	0.818	0.019	0.095
RBF Kernel	0.800	0.019	0.093	0.853	0.015	0.089

The evaluation of different models revealed unique strengths and limitations. Linear models, including both closed form and gradient descent approaches, provided a reliable baseline with a test R² of 0.785 and a test MSE of 0.022. While these models are simple and efficient, they struggled to capture the more complex nonlinear relationships in the data, limiting their predictive power for intricate patterns.

Regularized regression methods, like LASSO and Ridge, added a layer of stability compared to standard linear regression. Ridge regression showed a slight edge in generalization over LASSO, making it more robust. However, these models still fell short in addressing nonlinear complexities, making them better suited for straightforward scenarios where interpretability and stability are more critical than capturing complex relationships.

Polynomial regression, on the other hand, excelled at modeling nonlinear patterns. The degree 3 model improved validation performance but faced overfitting issues, as seen in its lower test R² of 0.728. In contrast, the degree 4 model performed significantly better, achieving a test R² of 0.818 and the second lowest test MSE of 0.019. It struck a good balance between capturing nonlinear patterns and maintaining computational efficiency, making it a solid choice for tasks requiring a moderate level of complexity.

The standout performer, however, was RBF Kernel Regression. It achieved exceptional results, with a validation R² of 0.800, a test R² of 0.853, and the lowest test MSE (0.015) and MAE (0.089). This model excelled at handling complex nonlinear patterns, delivering the highest accuracy and generalization. While RBF Kernel Regression is the top choice for tasks needing precise predictions, Polynomial Regression (degree 4) and Ridge Regression remain excellent alternatives for scenarios emphasizing simplicity, interpretability, or computational efficiency.

Regression with All features

categorical features (brand and country**)** were transformed using frequency encoding. This approach replaces each categorical value with its frequency in the dataset, indicating how often a specific brand or country appears. The result of this encoding added two new continuous columns: brand_freq and country_freq, which represent these frequencies as fractions of the total entries.

After the encoding process, the original brand and country columns were dropped, leaving a refined feature set. The final features include:

```
RangeIndex: 4425 entries, 0 to 4424
Data columns (total 7 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   engine_capacity  4425 non-null   float64
 1   cylinder         4425 non-null   int64  
 2   horse_power      4425 non-null   int64  
 3   top_speed        4425 non-null   int64  
 4   num_seats        4425 non-null   int64  
 5   brand_freq       4425 non-null   float64
 6   country_freq     4425 non-null   float64
```

dtypes: float64(3), int64(4)

The target variable, Price**_**USD, remained unchanged.

As with the earlier analysis, the dataset was divided into training (60%), validation (20%), and test (20%) subsets. With frequency encoded features, the models can now be re-evaluated to explore their performance using this enhanced representation.

Gradient Decent Regresson

The results from Gradient Descent show clear differences between including and excluding frequency encoded categorical features like brand and country.

When these features were included, the model performed better, with a validation R^2 of 0.759 and a test R^2 of 0.791. The model also had lower errors, with validation and test MSE values of 0.023 and 0.021, respectively. The mean absolute error (MAE) improved as well, dropping to 0.111 for validation and 0.105 for the test set.

Looking at the coefficients of the model offers more insights. With frequency encoding, the categorical features, brand and country, had positive contributions, with coefficients of 0.0282 and 0.0203. Meanwhile, the other features, like engine capacity, cylinder count, and horsepower, had similar coefficients in both approaches, highlighting their consistent role in the model's predictions.

Table 6:Gradient Descent Coefficients

Bias Term	Engine Capacity	Cylinder Count	Horsepower	Top Speed	Number of Seats	Brand	Country
-0.19666118	-0.01966612	0.0573225	0.16310361	0.10233201	0.00528305	0.02821893	0.0203463

In summary, adding frequency encoded categorical features improved the model's ability to understand the data and make better predictions.

Closed Form Solution

The performance of the Closed form Solution demonstrates a notable improvement with the inclusion of frequency encoded categorical features. With frequency encoding, the model achieved a validation R^2 of 0.764 and a test R^2 of 0.795, with validation and test MSE values of 0.023 and 0.021, respectively. The MAE values were also lower, at 0.111 for validation and 0.104 for testing. These metrics indicate that the model benefited from the additional information provided by frequency encoded features, resulting in better accuracy and reduced prediction errors.

The model's coefficients further demonstrate the impact of frequency encoding. When the categorical features were included, their coefficients were 0.0302 for brand and 0.0206 for country, showing that they positively influenced predictions. Without these features, the numerical variables like horsepower and top speed remained significant contributors, but their coefficients shifted slightly. This suggests that while numerical features are crucial, the encoded categorical data provided additional context that strengthened the model's predictions.

Table 7:closed form Coefficients

Bias Term	Engine Capacity	Cylinder Count	Horsepower	Top Speed	Number of Seats	Brand	Country
-0.19666967	-0.04387671	0.06947495	0.18378358	0.0879187	0.00435051	0.03018102	0.02063164

Overall, the addition of frequency encoded categorical features enhanced the Closed form Solution's accuracy and reduced prediction errors.

LASSO Regression

LASSO regression performed significantly better when frequency encoded categorical features were included in the model. With these features, the model achieved a validation R^2 of 0.762 and a test R^2 of 0.795, along with lower MSE values of 0.023 and 0.021 for validation and testing, respectively. The MAE values also dropped to 0.111 for validation and 0.104 for testing, indicating improved accuracy and fewer prediction errors.

The scatter plots provide a clear picture of the model's performance. In the validation dataset, the predicted values align closely with the actual values along the ideal fit line, showing that the model effectively captures the underlying patterns in the data. While there are some minor deviations at the extremes. The test dataset follows a similar pattern, with predictions also closely matching the actual values.

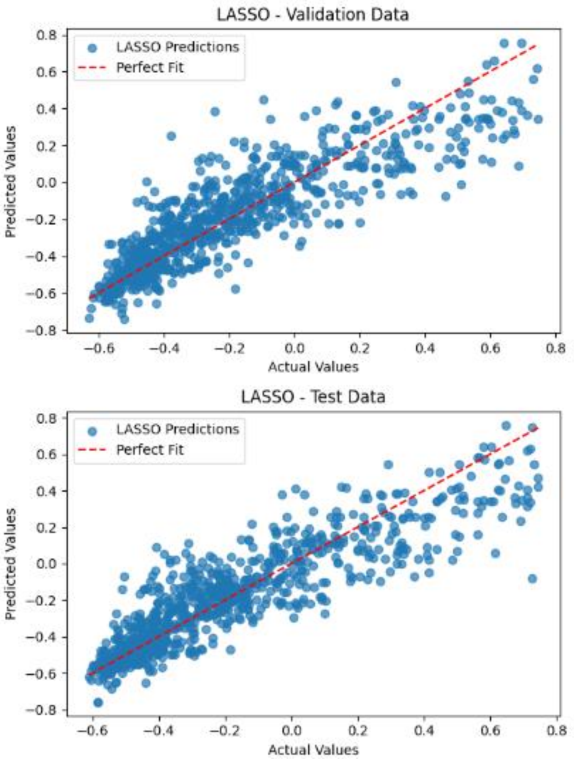

Figure 20:LASSO - Prediction Vs Actual values

The coefficients also highlight the contribution of the encoded features. The frequency encoded variables, brand and country, had positive coefficients of 0.0288 and 0.0194, respectively, showing their value in improving predictions. Meanwhile, numerical features like horsepower and top speed remained significant, with coefficients of 0.1806 and 0.0886, although slightly adjusted compared to the model without encoded features.

Table 8:LASSO coefficients

Engine Capacity	**Cylinder Count**	**Horsepower**	**Top Speed**	**Number of Seats**	**Brand**	**Country**
-0.03442131	0.06268918	0.18059659	0.08864936	0.00221199	0.02879559	0.01935714

The inclusion of frequency encoded categorical features boosted LASSO regression's performance, resulting in better predictive accuracy and a stronger ability to generalize across datasets.

Ridge Regression

The Ridge regression model showed significant improvement in performance when frequency encoded categorical features were included. With these features, the model achieved a validation

R² of 0.763 and a test R² of 0.795, with MSE values of 0.023 and 0.021 for validation and testing, respectively. The MAE values also dropped to 0.111 and 0.104 for validation and testing, demonstrating that incorporating frequency encoded features made the model more precise and reduced prediction errors.

Looking at the coefficients further explains these results. With frequency encoding, the categorical features contributed meaningfully to the model's predictions, with coefficients of 0.0302 for brand frequency and 0.0206 for country frequency. At the same time, numerical features like horsepower (0.1835) and top speed (0.0880) remained influential but were slightly adjusted due to the inclusion of categorical variables.

Table 9:Ridge coefficients

Engine Capacity	Cylinder Count	Horsepower	Top Speed	Number of Seats	Brand	Country
-0.04361191	0.06936924	0.18353378	0.08801794	0.00433352	0.0301645	0.02062516

The scatter plots provide a visual understanding of the model's predictions. For the validation dataset, the predicted values align closely with the actual ones, forming a tight cluster along the ideal fit line, with only minor deviations at the extremes. This indicates that the model effectively captures the data's patterns. In the test dataset, the predictions show a similar trend but with slightly broader spread, which is consistent with the marginal drop in R² from validation to testing. This slight variance is typical and indicates that the model generalizes well.

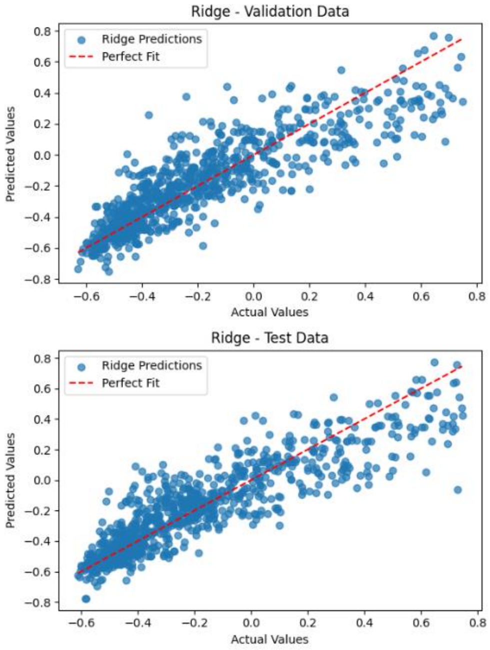

Figure 21:Ridge - Prediction Vs Actual values

In conclusion, Ridge regression improved with frequency encoded categorical features, resulting in more accurate and reliable predictions.

Polynomial Regression

The results for polynomial regression reveal the impact of incorporating frequency encoded categorical features, as well as the varying effects of increasing the polynomial degree. Notably, models with lower degrees, such as 2 and 3, performed well, while higher degrees led to significant overfitting and degraded performance.

degree 2 produced strong results, achieving a validation R² of 0.791 and a test R² of 0.823, with MSE values of 0.020 and 0.018, respectively. The MAE values were also low at 0.102 and 0.096, indicating a well-fitted model with robust predictive accuracy.

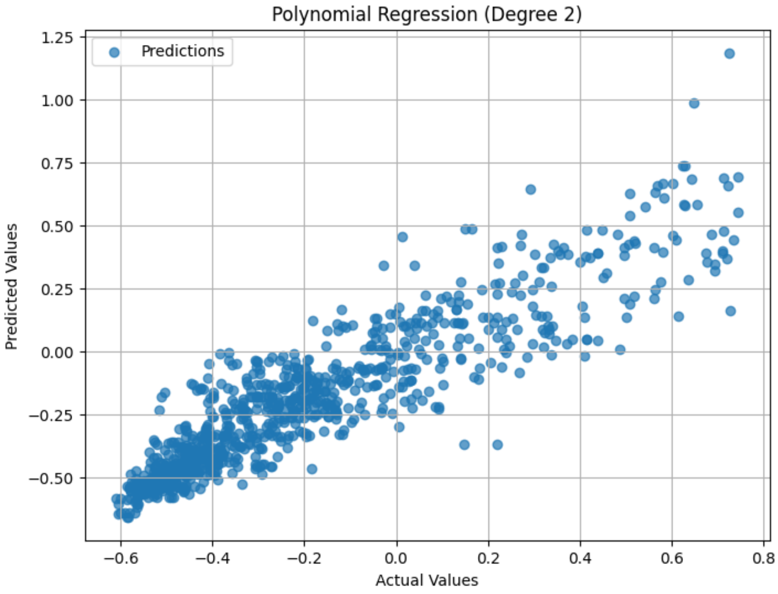

Figure 22:Polynomial Degree 2 - Prediction Vs Actual values

Degree 3 marginally improved the validation R² to 0.802 and reduced the validation MAE to 0.097. However, its test R² dropped significantly to 0.577, accompanied by an increased test MSE of 0.044. This suggests that degree 3 began to exhibit signs of overfitting, particularly on the test dataset.

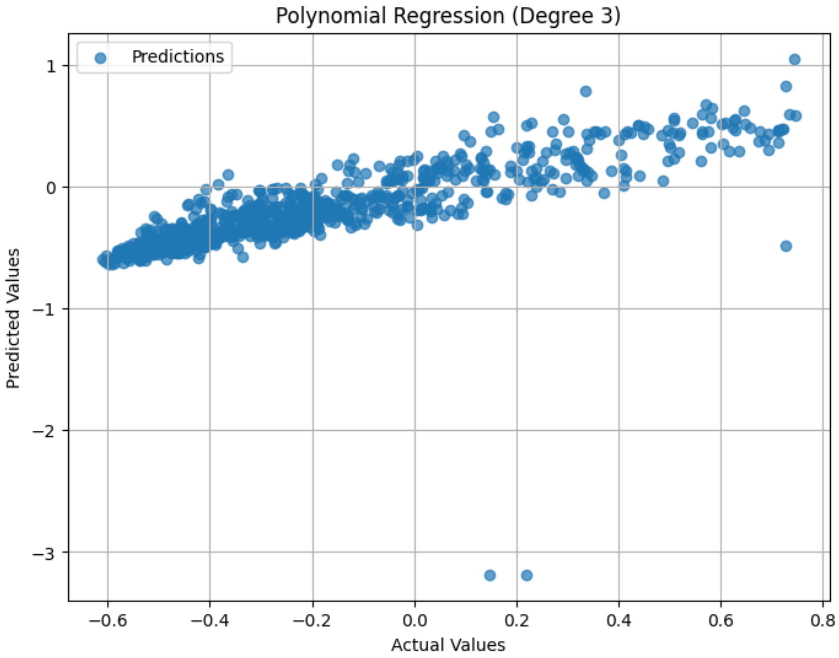

Figure 23:Polynomial Degree 3 - Prediction Vs Actual values

For polynomial degrees higher than 3, the model's performance worsened significantly. At degree 4, the validation MSE was 0.087, the MAE was 0.106, and the R² dropped to 0.108. The test results were similarly poor, with an MSE of 0.081, an MAE of 0.111, and an R² of just 0.210.

As the degree increased further, from 5 to 10, the model exhibited extreme errors and instability, with R² values turning negative—clear signs of severe overfitting. For instance, at degree 6, the validation R² plummeted to an astronomically low -910068662605336.1, with a validation MSE of 88574154210046.7. The test set fared even worse, showing an R² of -5782364636636195.0 and a test MSE of 594338556431499.8. These metrics highlight how increasing complexity made the model unmanageable, with predictions becoming wildly inaccurate.

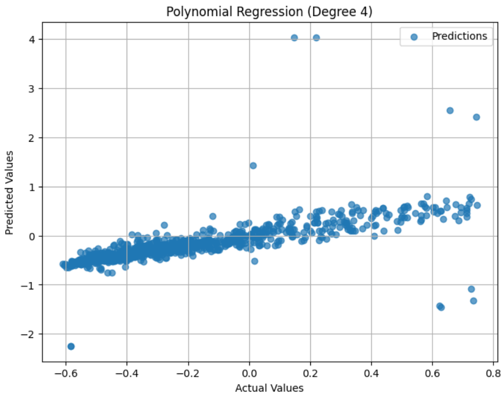

Figure 24:Polynomial Degree 4 - Prediction Vs Actual values

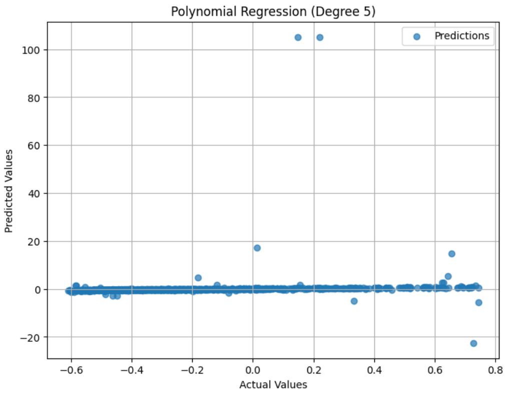

Figure 25:Polynomial Degree 5 - Prediction Vs Actual values

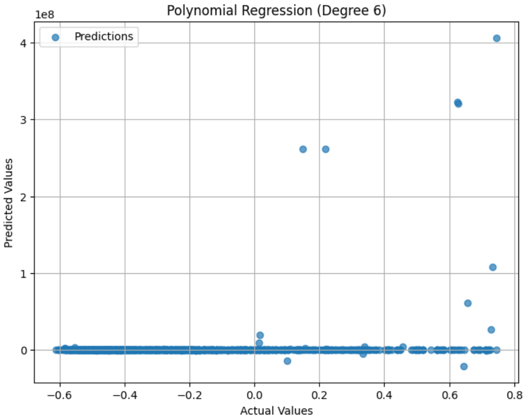

Figure 26:Polynomial Degree 6 - Prediction Vs Actual values

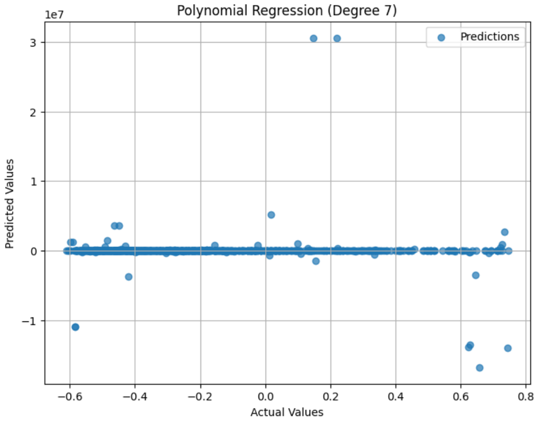

Figure 27:Polynomial Degree 7 - Prediction Vs Actual values

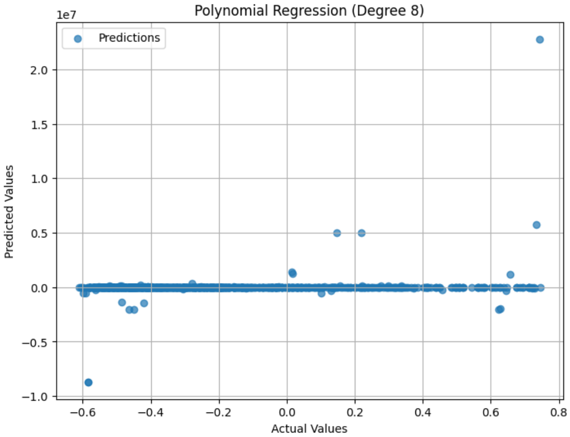

Figure 28:Polynomial Degree 8 - Prediction Vs Actual values

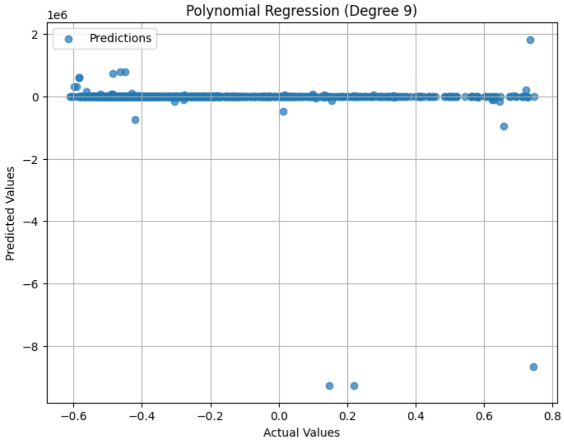

Figure 29:Polynomial Degree 9 - Prediction Vs Actual values

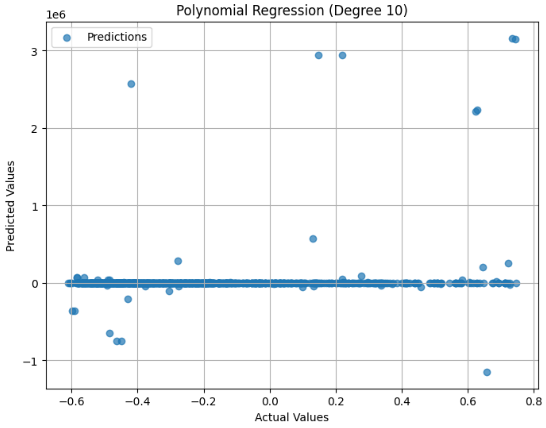

Figure 30:Polynomial Degree 10 - Prediction Vs Actual values

Radial Basis Function

The results for the RBF Kernel Regression model demonstrate its strong performance, both with and without frequency encoded categorical features. When categorical features were included, the model achieved a validation MSE of 0.016, MAE of 0.090, and R² of 0.832. On the test set, the MSE was also 0.016, MAE was 0.089, and R² reached 0.843. These results highlight the model's robust ability to capture nonlinear relationships in the data. The optimal hyperparameters for the best-performing RBF Kernel model were identified as {'C': 10, 'gamma': 0.1}.

In contrast, without categorical features, the RBF Kernel Regression model also performed well, with a validation R² of 0.800, MSE of 0.019, and MAE of 0.093. On the test set, it achieved an R² of 0.853, MSE of 0.015, and MAE of 0.089. Interestingly, the test R² without categorical features slightly outperformed the model with encoded features, though the validation performance was slightly lower.

These results indicate that the inclusion of frequency encoded categorical features did not drastically impact the RBF Kernel model's performance, as it already excels in capturing complex relationships. However, the encoded features did contribute to a slight improvement in the validation metrics.

Feature Selection with Forward Selection

The results of the forward feature selection process reveal the incremental impact of each selected feature on the model's validation R² score. Initially, the model selected feature 2, which yielded a validation R² of 0.6968. Adding feature 3 improved the validation R² to 0.7373, followed by feature 1, which further increased it to 0.7446. The inclusion of feature 0 raised the validation R² to 0.7509, and finally, feature 5 was selected, achieving the highest validation R² of 0.7580.

The selected features, in order of selection, are [2, 3, 1, 0, 5]. Using this optimized subset of features, the model attained a test R² of 0.7911, which is consistent with the highest performance observed in prior models using all features. This suggests that these five features capture the majority of the predictive power in the dataset, supporting the efficacy of forward selection as a feature selection technique to reduce dimensionality without sacrificing model performance.

Comparison of all models

The comparison of all models demonstrates varying levels of performance in terms of validation and test metrics, The table below shows summary of the results

Table 10: All Models Comparison

Model	Validation R^2	Validation MSE	Validation MAE	Test R^2	Test MSE	Test MAE
Closed form Solution	0.764	0.023	0.111	0.795	0.021	0.104
Gradient Descent	0.759	0.023	0.111	0.791	0.021	0.105
LASSO	0.762	0.023	0.111	0.795	0.021	0.104
Ridge	0.763	0.023	0.111	0.795	0.021	0.104
Polynomial (Degree 2)	0.791	0.020	0.102	0.823	0.018	0.096
RBF Kernel	0.832	0.016	0.090	0.843	0.016	0.089

RBF Kernel Regression proved to be the most successful model, delivering the highest accuracy and consistency. Polynomial Regression with Degree 2 followed as a strong alternative, striking a balance between capturing non-linear relationships and maintaining simplicity. Regularized linear models like LASSO and Ridge Regression performed comparably to the Closed-form Solution, offering slight but noticeable improvements over Gradient Descent. These results highlight the strengths of different models depending on the complexity and structure of the data.

Exploring Alternate Target: Building Regression Models for Engine Capacity

To Expand the analysis, we chose a different target variable, engine capacity. Using this new target, we built and evaluated regression models. The dataset now includes five features: cylinder, horsepower, top speed, number of seats, and price in USD. After excluding any irrelevant columns, we split the data into training, validation, and test sets. standardization was applied, and a bias term was added to make the data compatible with linear regression models.

Below, the results of each model is shown and discussed.

Closed-form solution

The closed-form solution regression model performed well with **engine capacity** as the target variable, showing strong predictive results for both the validation and test datasets.

On the validation set, the model achieved an MSE of 0.247, an MAE of 0.362, and an R² score of 0.782, indicating a solid fit to the data. The test set results were similar, with an MSE of 0.270, an MAE of 0.370, and an R² score of 0.745, suggesting that the model generalizes effectively to unseen data.

Table 11:closed form coefficients

Bias Term	Cylinder	Horse power	Top speed	Num seats	Price USD
2.54553672	0.6795754	0.49395242	-0.21578086	0.04771393	-0.09280789

Examining the regression coefficients gives us a clearer picture of feature importance. The most significant contributors were cylinder (0.6796) and horsepower (0.4940), both positively associated with engine capacity. On the other hand, top speed (-0.2158) and price in USD (-0.0928) showed negative coefficients, indicating an inverse relationship. The number of seats (0.0477) had a small but positive impact on the target variable.

The closed-form solution successfully captured the key relationships between the features and engine capacity, offering reliable accuracy and consistent generalization across datasets.

Gradient Descent

The gradient descent regression model showed strong predictive performance, closely matching the results of the closed-form solution.

For the validation set, the model achieved an MSE of 0.250, an MAE of 0.364, and an R² score of 0.780, indicating a good fit. On the test set, the MSE increased slightly to 0.272, with an MAE of 0.371 and an R² score of 0.743. These metrics suggest the model effectively captures patterns in the data and generalizes well to unseen data.

Looking at the regression coefficients provides insight into feature importance. Cylinder (0.6877) emerged as the strongest positive predictor of engine capacity, followed by horsepower

(0.4405). In contrast, top speed (-0.2152) and price in USD (-0.0451) showed negative associations with the target variable. The number of seats (0.0523) had a small positive effect.

Table 12:Gradient descent coefficients

Bias Term	Cylinder	Horse power	Top speed	Num seats	Price USD
2.54542683	0.68772365	0.44045876	-0.21523219	0.05227719	-0.04506331

Overall, the gradient descent model produced results consistent with the closed-form solution, highlighting its reliability and confirming the relevance of the selected features in predicting engine capacity.

LASSO

The LASSO regression model, performed on par with both the closed-form solution and gradient descent while introducing regularization to reduce overfitting.

For the validation set, the model achieved an MSE of 0.247, an MAE of 0.363, and an R^2 of 0.782. On the test set, the MSE rose slightly to 0.271, accompanied by an MAE of 0.370 and an R^2 of 0.744.

Examining the coefficients shows feature significance. Cylinder (0.6790) emerged as the most impactful predictor, followed closely by horsepower (0.4877). Features such as top speed (-0.2146) and price in USD (-0.0871) maintained a negative relationship with engine capacity, while the number of seats (0.0475) showed a small positive influence. These patterns align with the insights from other models, reinforcing their validity.

Table 13:LASSO coefficients

Cylinder	Horse power	Top speed	Num seats	Price USD
0.67902843	0.48771875	-0.2145977	0.04754824	-0.08710087

The scatter plots illustrate predicted versus actual engine capacity values for both datasets. Predictions closely align with the perfect fit line in the validation plot, with minor deviations at extremes. The test plot shows a slightly broader spread, consistent with the lower test R^2 (0.744), reflecting a slight decrease in accuracy on unseen data.

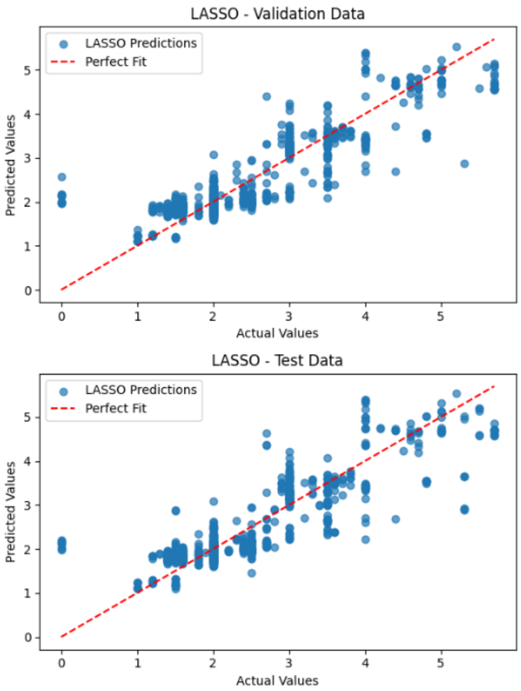

Figure 31:LASSO - Actual Vs Predicted

LASSO's use of an alpha value of 0.001 allowed for subtle regularization, reducing the magnitude of less important coefficients without compromising predictive accuracy. This added step ensures the model remains interpretable and resilient against overfitting. Overall, LASSO strikes an effective balance between simplicity and performance.

Ridge Regression

The Ridge regression model for predicting engine capacity delivers consistent and reliable performance across both validation and test datasets, as highlighted by the scatter plots. In the validation plot, the predicted values align closely with the red dashed line, representing perfect predictions. This indicates that the model captures the underlying data relationships effectively, with only minor deviations, particularly at the extremes. The test plot also shows good alignment, though the predictions are slightly more spread out compared to the validation data, reflecting a modest decrease in performance on unseen data.

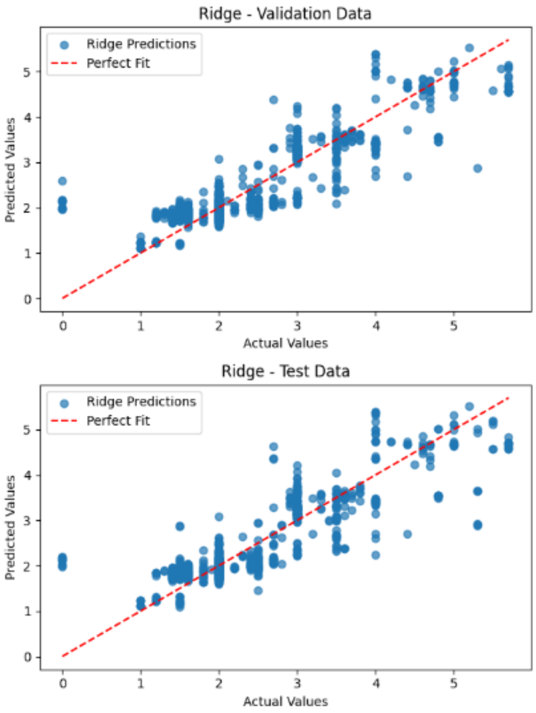

Figure 32: Ridge - scatter plots

The model achieved an R² of 0.782 on the validation set, meaning it explains 78.2% of the variance in engine capacity. On the test set, the R² dropped slightly to 0.744, demonstrating solid generalization. The Mean Squared Error (MSE) was 0.247 on the validation set and 0.271 on the test set, while the Mean Absolute Error (MAE) stood at 0.363 and 0.370, respectively. These low error rates reinforce the model's accuracy in predicting engine capacity.

With an alpha value of 10, Ridge regression effectively balances fitting the training data while regularizing the coefficients to prevent overfitting. The model's coefficients highlight the influence of each feature, with cylinder count and horsepower emerging as the strongest positive contributors to engine capacity. Other features have smaller or slightly negative impacts.

Table 14:Ridge coefficients

Cylinder	Horse power	Top speed	Num seats	Price USD
0.67688149	0.48785197	-0.21352576	0.04880189	-0.08702778

Overall, Ridge regression proves to be a dependable and interpretable choice for modeling engine capacity, combining strong predictive performance with generalization to unseen data.

Polynomial Regression

The performance of Polynomial Regression models of varying degrees for predicting engine capacity shows a clear trend in accuracy and generalization. For Degree 2, the model achieves strong results, with a validation R² of 0.808 and test R² of 0.795, demonstrating robust generalization. The corresponding MSE values, 0.218 for validation and 0.217 for test, and the low MAE values, 0.323 and 0.324 respectively, confirm its effectiveness. However, as the polynomial degree increases, performance begins to vary significantly.

With Degree 3, the validation R² improves slightly to 0.821, and the validation MSE drops to 0.203, indicating better fitting on the validation data. However, test performance deteriorates, as evidenced by a lower R² of 0.720 and a higher MSE of 0.297. For Degree 4, overfitting becomes more pronounced, with a validation R² dropping to 0.643 and a test R² of 0.704.

Higher-degree polynomials (5 and beyond) exhibit catastrophic overfitting, with validation and test R² values plummeting into large negative values. For instance, Degree 6 achieves validation R² of -26.828 and test R² of -1632.815, with MSE values skyrocketing to 31.564 for validation and 1729.745 for the test. This dramatic decline continues with higher-degree models, such as Degree 10, where validation and test MSE values escalate to 1.73×10^{11} and 6.49×10^{11}, respectively, and R² values reach -1.52×10^{11} and -6.13×10^{11}.

The trend clearly demonstrates that lower-degree polynomials, particularly Degree 2, provide the best balance between accuracy and generalization. Higher-degree models suffer from severe overfitting, leading to instability and unreliable predictions on new data.

Table 15 shows the exact results to each model:

Table 15: Polynomial Regression Metrics

Model	Validation R²	Validation MSE	Validation MAE	Test R²	Test MSE	Test MAE
Polynomial (Degree 2)	0.808	0.218	0.323	0.795	0.217	0.324
Polynomial (Degree 3)	0.821	0.203	0.281	0.720	0.297	0.293
Polynomial (Degree 4)	0.643	0.405	0.292	0.704	0.313	0.275
Polynomial (Degree 5)	-0.834	2.080	0.328	-0.062	1.124	0.292
Polynomial (Degree 6)	-26.828	31.564	0.688	-1632.815	1729.745	2.553
Polynomial (Degree 7)	-6432.808	7297.597	8.460	-33376687.5	35336426.528	306.046

Degrees 8, 9, and 10 shows even worse results

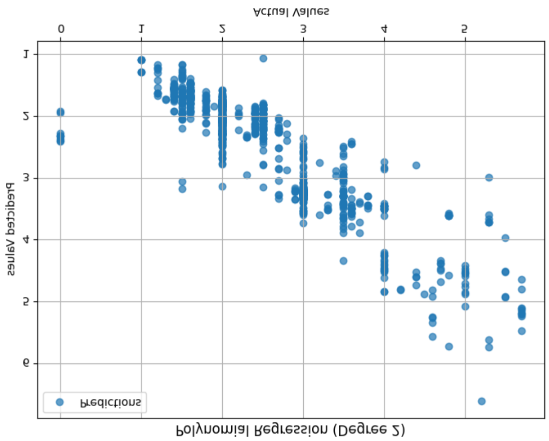

Figure 33:Polynomial Degree 2 - Prediction Vs Actual values

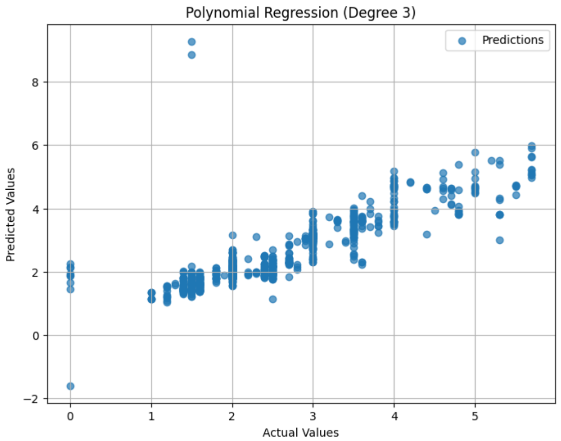

Figure 34:Polynomial Degree 3 - Prediction Vs Actual values

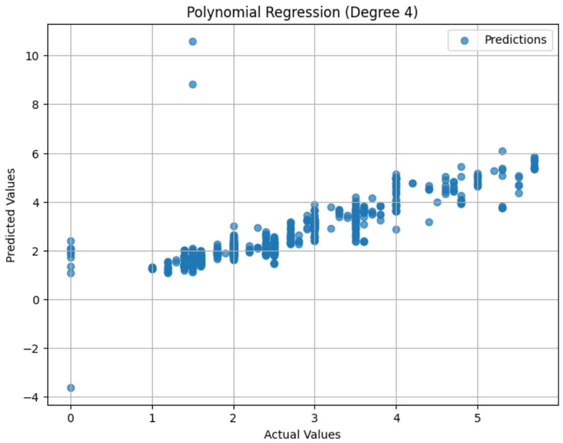

Figure 35:Polynomial Degree 4 - Prediction Vs Actual values

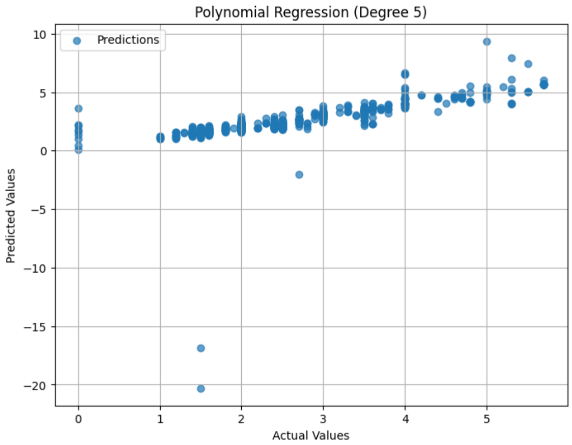

Figure 36:Polynomial Degree 5 - Prediction Vs Actual values

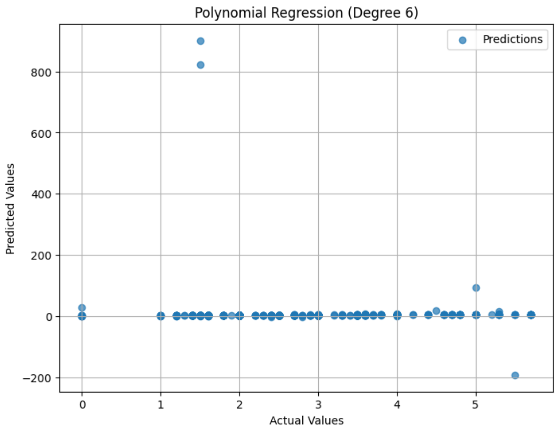

Figure 37:Polynomial Degree 6 - Prediction Vs Actual values

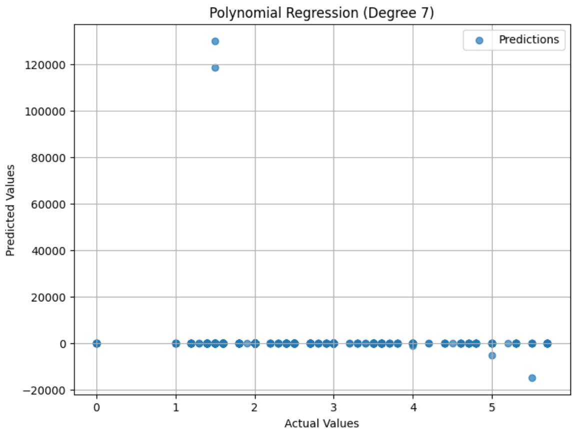

Figure 38:Polynomial Degree 7 - Prediction Vs Actual values

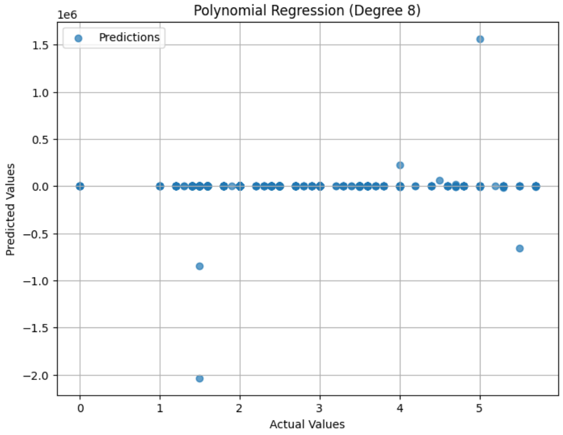

Figure 39:Polynomial Degree 8 - Prediction Vs Actual values

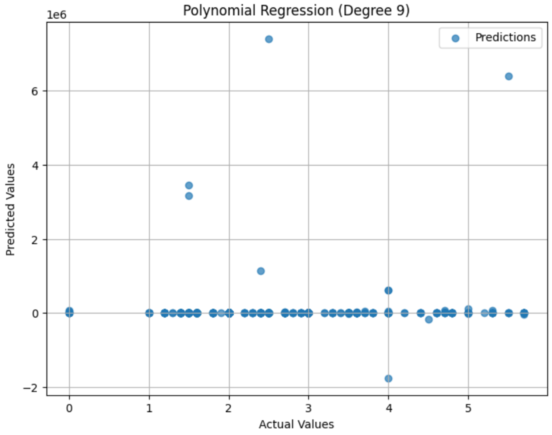

Figure 40:Polynomial Degree 9 - Prediction Vs Actual values

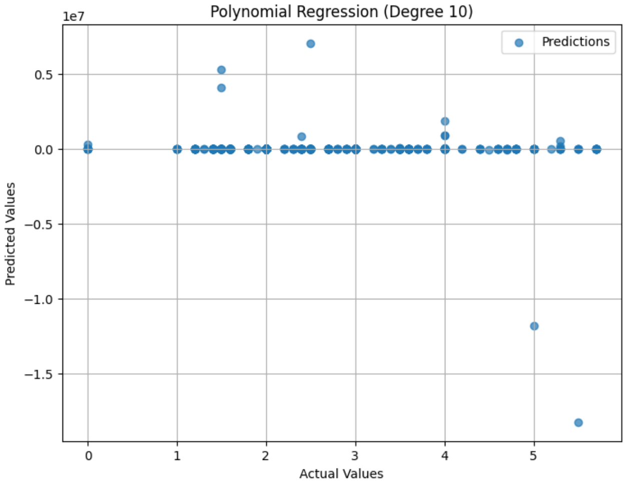

Figure 41:Polynomial Degree 8 - Prediction Vs Actual values

Radial Basis Function

The RBF Kernel Regression model demonstrates excellent performance in predicting engine capacity, achieving high accuracy and generalization. On the validation set, the model attains an R^2 of 0.898, indicating that 89.8% of the variance in the target variable is explained by the model. The corresponding MSE of 0.115 and MAE of 0.186 further confirm its precision.

On the test set, the model's performance improves even further, with an impressive R^2 of 0.909, highlighting its ability to generalize well to unseen data. The test MSE of 0.096 and MAE of 0.176 underscore its effectiveness in minimizing errors. The optimal hyperparameters for the RBF kernel are found to be C = 10 and gamma = 1, suggesting that this balance between regularization and the kernel's influence provides the best fit.

Overall, the RBF Kernel Regression model outperforms other regression techniques

Feature Selection with Forward Selection

The results from the forward feature selection process demonstrate the gradual improvement in the model's validation R^2 score with each added feature. Initially, selecting feature 0 produced a validation R^2 of 0.7141. Incorporating feature 1 boosted the score to 0.7339, followed by feature 2, which further increased the R^2 to 0.7746. Adding feature 4 led to a slight improvement, bringing the validation R^2 to 0.7799. The final set of selected features, in the order of selection, was [0, 1, 2, 4]. With this optimized feature subset, the model achieved a test R^2 of 0.7399, indicating that these four features effectively capture most of the predictive power in the data.

Comparison Between All Models

The comparison of all models demonstrates varying levels of performance in terms of validation and test metrics, The table below shows summary of the results

Table 16: All models Comparison

Model	Validation R^2	Validation MSE	Validation MAE	Test R^2	Test MSE	Test MAE
Closed form Solution	0.782	0.247	0.362	0.745	0.270	0.370
Gradient Descent	0.780	0.250	0.364	0.743	0.272	0.371
LASSO	0.782	0.247	0.363	0.744	0.271	0.370
Ridge	0.782	0.247	0.363	0.744	0.271	0.370
Polynomial (Degree 2)	0.808	0.218	0.323	0.795	0.217	0.324
RBF Kernel	0.898	0.115	0.186	0.909	0.096	0.176

The Closed-form Solution, Gradient Descent, LASSO, and Ridge models all achieved similar validation R^2 scores around 0.782, with corresponding test R^2 values of 0.745 to 0.744, indicating consistent performance. These models also exhibited relatively close MSE and MAE values, with validation MSE ranging from 0.247 to 0.250 and test MSE between 0.270 and 0.272. In contrast, the Polynomial Regression model with degree 2 performed slightly better, achieving a validation R^2 of 0.808 and a test R^2 of 0.795, along with lower MSE and MAE values. However, the standout performer was the RBF Kernel Regression, which significantly outperformed the others with a validation R^2 of 0.898 and a test R^2 of 0.909, alongside the lowest MSE and MAE values, indicating its superior ability to capture the underlying patterns in the data.

Conclusion

This Assignment provided valuable insights into regression models by practicing on predicting car prices and engine capacities using a dataset from YallaMotors. By exploring various regression techniques, the analysis highlighted the strengths and weaknesses of each approach.

Linear regression methods, including the closed-form solution and gradient descent, served as reliable baselines, offering consistent performance. Regularization techniques like LASSO and Ridge enhanced these models by improving stability and addressing multicollinearity, making them more robust and interpretable. However, these linear approaches struggled to fully capture nonlinear relationships in the data.

Polynomial regression showed promise in uncovering nonlinear patterns, particularly with degrees 2 and 4, which achieved a strong balance between accuracy and generalization. However, higher-degree polynomial models quickly fell into overfitting, leading to poor generalization and instability. This reinforced the importance of careful model complexity selection.

The standout performer was the RBF Kernel Regression model, which excelled in capturing intricate relationships within the data. It delivered the highest accuracy and generalization across all tasks, making it the most effective option for both car price prediction and engine capacity analysis. Its ability to manage complex patterns sets it apart as a reliable tool for nonlinear datasets.

The assignment also highlighted the value of forward feature selection, which simplified the models by identifying the most critical features without sacrificing performance. Adding frequency encoding for categorical features like brand and country further boosted the performance of all models, showing how encoded non-numerical data can enhance predictive power.