
**Faculty of Engineering and Technology**
**Electrical and Computer Engineering Department**

# MACHINE LEARNING AND DATA SCIENCE
# ENCS5341

## Assignment 2
## Regression Analysis and Model Selection

---

**Student Name: Raghad Murad Buzia**
**Student #: 1212214**

**Instructor: Ismail Khater**

**Section: 3**

**Date: 28/11/2024**

# Abstract

This assignment aims to provide a comprehensive analysis of regression modeling and model selection using a dataset sourced from the YallaMotors website, containing information about cars such as make, model, year, mileage, and price. The primary objective of the project is to develop and evaluate various regression models to predict car prices, optimizing accuracy and preventing overfitting through techniques like feature selection, regularization, and hyperparameter tuning.

The dataset, consisting of 6,308 rows and 9 columns, was preprocessed to handle missing values, encode categorical variables, and standardize numerical features. Both linear (e.g., linear regression, LASSO, ridge) and nonlinear models (e.g., polynomial regression, radial basis function kernel) were implemented and compared using performance metrics such as mean squared error (MSE), mean absolute error (MAE), and R-squared.

The results show that the SVR model with a Radial Basis Function kernel demonstrated superior performance on the test set, achieving a low mean squared error (MSE) and a high R-squared score. Visualizations of error distribution, feature importance, and predicted versus actual values supported the conclusions. Limitations include computational constraints for higher-degree polynomial regression and the need for more diverse data sources.

# Table of Contents

# Table of Figures



# Table of Tables

# 1. Introduction

## Overview of Regression Analysis and Its Applications

Regression analysis is a fundamental statistical and machine learning technique used to model the relationship between a dependent variable (target) and one or more independent variables (features). It serves as a key tool in predictive modeling, enabling the estimation of outcomes based on input data. Applications of regression analysis span diverse fields, such as economics (predicting housing prices), healthcare (estimating patient risk), and engineering (forecasting system performance). In this project, regression is applied to predict car prices based on a variety of features, aiming to uncover insights into the factors influencing pricing.

## Objective of the assignment

The primary objective of this project is to build and evaluate regression models to accurately predict car prices. To achieve this, the project focuses on:

1. Implementing and comparing a variety of regression techniques, including linear models (e.g., Linear Regression, LASSO, Ridge) and nonlinear models (e.g., Polynomial Regression, Support Vector Regression).
2. Improving prediction accuracy by employing feature selection methods, such as forward selection.
3. Applying regularization techniques to prevent overfitting and enhance model generalization.
4. Conducting hyperparameter tuning to optimize model performance and select the best regression model.

By implementing robust regression techniques and optimizing model parameters, this assignment aims to deliver a predictive model capable of generalizing well to unseen data while uncovering the most significant factors influencing car prices.

# 1. Dataset Description

## Source of the Dataset

The dataset was sourced from the YallaMotors website, a platform for buying and selling cars. The dataset provides detailed information about various cars and their attributes, making it well-suited for regression analysis aimed at predicting car prices.

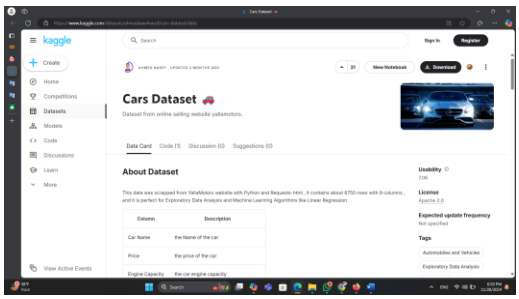


**Figure 1: The Dataset Source `YallaMotors website: Cars Dataset 🚙`**

## Characteristics

### 1. Number of Rows and Columns

The dataset contains 6,309 rows and 9 columns, with each row representing a unique car and each column describing a specific feature or attribute of the car.

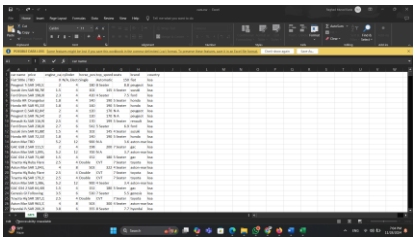

**Figure 2: Dataset as a CSV File**


**Figure 3: Number of Rows and Columns `the code to get the number of rows and columns`**

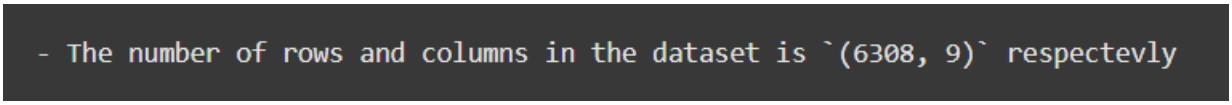


**Figure 4: Number of Rows and Columns `from the code output`**

## 2. Description of Features

The dataset includes the following features:

Table 1: Description of Features in The Dataset

| Columns | Description |
|---|---|
| Car Name | The name of the car (e.g., Toyota Corolla, BMW X5) |
| Price | The price of the car (may vary in currency) |
| Engine Capacity | The engine capacity of the car (e.g., 1.8L, 3.5L) |
| Cylinder | The number of cylinders in the car engine |
| Horse Power | The horse power of the car |
| Top Speed | The maximum speed of the car |
| Seats | The number of seats in the car |
| Brand | The car's brand (e.g., Toyota, BMW) |
| Country | The country where the car is sold |

### Target Variable

The target variable is Price, which represents the cost of the car. Since prices are listed in various currencies, standardization was required for uniformity.

### Challenges

One of the primary challenges in this dataset was the presence of car prices in different currencies (e.g., USD, AED, SAR). To address this, a conversion process was implemented using known exchange rates to unify all prices in USD. Additionally, the dataset contained missing and inconsistent values in some features, which required careful preprocessing.

### Preprocessing Steps

#### 1. Data Cleaning

- Missing values in the dataset were identified and handled appropriately. For example: Prices with unrecognized currencies were replaced with the median price after conversion to USD.

- Missing values in numerical features (e.g., engine_capacity, horse_power) were filled using the most frequent values.
- Invalid entries in numerical columns were removed or corrected.

## 2. Feature Encoding (Categorical Variables):

- Categorical variables such as Brand, Country, and Car Name were encoded using binary encoding to facilitate their use in regression models.

## 3. Normalization/Standardization of Numerical Features:

- Numerical features, including engine_capacity, horse_power, and top_speed, were standardized using the StandardScaler to ensure all features had similar scales, which improves the performance of regression models.

## Splitting the Dataset:

- The dataset was split into training (60%), validation (20%), and test (20%) sets to enable robust model training and evaluation.

**Figure 5: Data inspection & Data Preprocessing `in the code`**

# 2. Regression Models and Their Performance on the Validation Set

## Linear Models

### 1. Linear Regression (Closed-Form Solution and Gradient Descent Comparison)

Linear Regression is a foundational technique where the relationship between the target variable (car price in our case) and the features (e.g., engine capacity, horsepower) is modeled as a linear equation.

Closed-Form Solution: The parameters of the linear model were computed using the closed-form solution, derived from the normal equation $\boldsymbol{\theta} = (\boldsymbol{X^T X})^{-1} \boldsymbol{X^T y}$

Gradient Descent: A comparison was made using gradient descent, where the parameters were updated iteratively to minimize the error.

Performance: The model's performance on the validation set was evaluated using the following metrics:

Mean Squared Error (MSE), Mean Absolute Error (MAE), and R-squared

- Mean Squared Error (MSE)
- Mean Absolute Error (MAE)
- R-squared ($R^2$)

The gradient descent and closed-form models were evaluated on the validation set and compared to each other.

**Figure 6: Linear Regression (Closed-Form Solution and Gradient Descent Comparison) `in the code`**

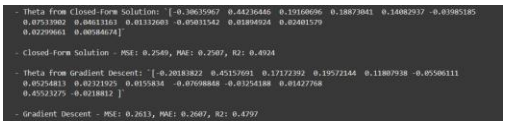


**Figure 7: Linear Regression (Closed-Form Solution and Gradient Descent Comparison) `code output`**

- Theta from Closed-Form – Performance:
  - The coefficients (theta values) obtained from the closed-form solution represent the weights assigned to each feature in the linear regression model. These values indicate the importance of each feature in predicting car prices. For example, a positive coefficient indicates a positive relationship with the target variable (price), while a negative coefficient indicates an inverse relationship. Theta Values: The coefficients include values such as 0.4423 for the first feature, -0.0398 for one of the features, suggesting that certain features (such as the first feature) have a higher positive impact on the price, while others reduce the predicted value.
  - MSE: 0.2549, MAE: 0.2507, $R^2$: 0.4924

The closed-form solution showed decent performance, with an MSE of 0.2549 and a MAE of 0.2507. The $R^2$ value of 0.4924 indicates that approximately 49% of the variance in the car prices is explained by the features in the model. This is a moderate performance, suggesting that while the model captures some of the relationship, there is significant room for improvement.

- Gradient Descent - Performance:
  - Theta from Gradient Descent: The coefficients obtained via gradient descent are slightly different from those obtained through the closed-form solution, particularly for features like 0.4515 and -0.0551.
  - MSE: 0.2613, MAE: 0.2607, $R^2$: 0.4797

  The gradient descent method yielded a slightly higher MSE and MAE, indicating that the iterative process did not converge as well as the closed-form solution. The $R^2$ value was also slightly lower, suggesting that the model's explanatory power was reduced. This difference is likely due to the fact that gradient descent can sometimes converge to a local minimum rather than the global one, depending on the learning rate and other factors.

## 2. Regularized Linear Models (Lasso and Ridge Regression)

- Lasso Regression - Performance:
  - MSE: 0.2912, MAE: 0.2712, $R^2$: 0.4202

Lasso Regression, with L1 regularization, performed relatively worse than both the closed-form linear regression and Ridge. Its ability to select important features resulted in some coefficient values being driven to zero, but it still struggled with model accuracy as evidenced by its MSE and $R^2$. The model's ability to reduce overfitting is evident in its lower $R^2$ compared to the non-regularized models.

- Ridge Regression - Performance:
  - MSE: 0.2549, MAE: 0.2507, $R^2$: 0.4924

Ridge Regression, which uses L2 regularization, showed the same performance as the closed-form linear regression, with an MSE of 0.2549 and an $R^2$ of 0.4924. This indicates that Ridge regression helped prevent overfitting without overly penalizing the model's complexity, maintaining a similar fit to the data.

- Best Regularization Parameters:
  - Lasso: The best α\alphaα value for Lasso was found to be 0.01, indicating a relatively weak penalty on the coefficients.
  - Ridge: The best α\alphaα value for Ridge was found to be 100, suggesting a stronger regularization effect, which helped control overfitting.

```
- Lasso Regression - MSE: 0.2912, MAE: 0.2712, R2: 0.4202
- Ridge Regression - MSE: 0.2549, MAE: 0.2507, R2: 0.4924
- Best Lasso Alpha: `{'alpha': 0.01}`
- Best Ridge Alpha: `{'alpha': 100}`
```

**Figure 8: Regularized Linear Models (Lasso and Ridge Regression) `code output`**

## Non- Linear Models

### 1. Polynomial Regression (Degrees 2–10)

Polynomial regression was explored to capture nonlinear relationships in the data. The performance varied greatly depending on the degree of the polynomial:

- Degree 2 Polynomial Regression:
  - MSE: 74.1207, MAE: 1.0567, $R^2$: -146.5661

- The model exhibited a poor fit, with a highly negative $R^2$, indicating significant overfitting.

- Higher Degree Polynomial Regression (Degrees 3–10):

  - As the degree of the polynomial increased, the MSE and MAE skyrocketed, with MSE values reaching 149276497018665033531392 for degree 3 and very high MAE values.

  - The $R^2$ values for these models were extremely negative, indicating poor model performance and overfitting as the polynomial degree increased.

  - Degree 6 Polynomial had an MSE of 1750669756576487936.0000, and $R^2$ of -3485388630866967040.0000, highlighting the extreme overfitting that occurred.

The results of polynomial regression indicate that increasing the degree of the polynomial leads to overfitting, causing large errors and poor generalization to the validation set.

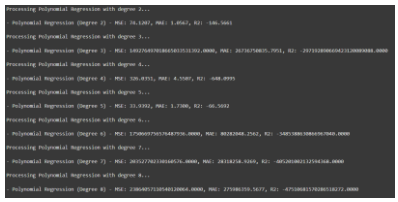

**Figure 9: Polynomial Regression (Degrees 2–10) `code output`**

## 2. Support Vector Regression (SVR with RBF Kernel)

- SVR Performance:

- MSE: 0.1500, MAE: 0.1936, $R^2$: 0.7014

SVR with an RBF kernel performed the best out of all models tested. It achieved the lowest MSE and MAE, and the highest $R^2$, indicating a strong ability to capture complex nonlinear relationships between the features and the target variable. The model's $R^2$ of 0.7014 suggests that it explains 70% of the variance in car prices, which is significantly better than the linear models.

```
- Support Vector Regression (RBF Kernel) - MSE: 0.1500, MAE: 0.1936, R2: 0.7014
```

**Figure 10: Support Vector Regression (SVR with RBF Kernel) `code output`**

### Summary of Best Models

- Best Model based on MSE: SVR

SVR outperformed all other models in terms of minimizing MSE, indicating its superior performance in fitting the data with minimal error.

- Best Model based on $R^2$: SVR

SVR also had the highest $R^2$, demonstrating its strong ability to explain the variability in the target variable.

Based on the results, Support Vector Regression (SVR) with an RBF kernel proved to be the most effective model for predicting car prices, outperforming both the linear models and polynomial regression. The use of regularization in Lasso and Ridge regression helped prevent overfitting but did not outperform the more complex nonlinear models. Polynomial regression with higher degrees resulted in extreme overfitting, emphasizing the importance of model complexity balance.

```
The best models:

- Best Model based on MSE: SVR

- Best Model based on R2: SVR
```

**Figure 11: The best models based on MSE and $R^2$**

## 3. Explanation of Feature Selection Results Using Forward Selection

Feature selection is a crucial step in building a regression model, especially when working with datasets that have many features. It helps to identify the most important features for predicting the target variable (car price, in this case) and removes irrelevant or redundant features that could lead to overfitting. In this project, we implemented forward selection to iteratively add the best-performing features based on their contribution to minimizing the model's Mean Squared Error (MSE).

### Forward Selection Process

Forward selection starts with an empty model and adds one feature at a time. In each iteration, the feature that, when added, minimizes the MSE on the validation set is chosen. This process

continues until adding more features does not improve the performance or a predefined maximum number of features is reached.

- The forward selection procedure was carried out as follows:

The forward selection procedure involved iteratively adding features to the model based on their contribution to minimizing the MSE. Starting with 'horse_power' in the first iteration, the MSE decreased to 0.2720, indicating its importance. Subsequent features such as 'brand_6', 'brand_2', and 'engine_capacity' progressively reduced the MSE, highlighting the significance of car attributes like brand and engine size in predicting price. By the 10th iteration, the model included features like 'seats' and 'country_2', bringing the MSE down to 0.2551. The process stopped after 13 iterations, as the model reached its optimal configuration with 10 features, preventing overfitting and ensuring a balanced model.

```
Iteration 1: Adding feature 'horse_power' with MSE: 0.2720
Iteration 2: Adding feature 'brand_6' with MSE: 0.2681
Iteration 3: Adding feature 'brand_2' with MSE: 0.2606
Iteration 4: Adding feature 'car name_4' with MSE: 0.2589
Iteration 5: Adding feature 'engine_capacity' with MSE: 0.2577
Iteration 6: Adding feature 'brand_4' with MSE: 0.2568
Iteration 7: Adding feature 'car name_5' with MSE: 0.2561
Iteration 8: Adding feature 'seats' with MSE: 0.2554
Iteration 9: Adding feature 'car name_6' with MSE: 0.2553
Iteration 10: Adding feature 'country_2' with MSE: 0.2551
Iteration 11: Adding feature 'car name_3' with MSE: 0.2550
Iteration 12: Adding feature 'brand_0' with MSE: 0.2550
Iteration 13: Adding feature 'car name_11' with MSE: 0.2549
Reached maximum number of features (10). Stopping.
```

**Figure 12: Selected Features by Forward Selection**

```python
# Function to Forward Selection for Feature Selection
def forward_selection(X_train, y_train, X_val, y_val, max_features=None):
    selected_features = []
    remaining_features = list(X_train.columns)
    best_mse = float("inf")
    iteration = 0

    while remaining_features:
        mse_list = []
        for feature in remaining_features:
            model = LinearRegression()
            model.fit(X_train[selected_features + [feature]], y_train)
            y_val_pred = model.predict(X_val[selected_features + [feature]])
            mse = mean_squared_error(y_val, y_val_pred)
            mse_list.append((feature, mse))

        best_feature, best_mse_candidate = min(mse_list, key=lambda x: x[1])

        if best_mse_candidate < best_mse:
            best_mse = best_mse_candidate
            selected_features.append(best_feature)
            remaining_features.remove(best_feature)
            iteration += 1
            print("Iteration {}: Adding feature '{}' with MSE: {:.4f}".format(iteration, best_feature, best_mse))

        elif max_features is not None and len(selected_features) >= max_features:
            print("Reached maximum number of features ({}). Stopping.".format(max_features))
            break
        else:
            print("No improvement in performance. Stopping.")
            break

    print("\nSelected features:", selected_features)
    return selected_features
```

**Figure 13: Forward Selection `code`**

The `forward_selection` function iterates over the remaining features, evaluates each feature's contribution to minimizing **MSE**, and adds the feature that results in the best performance. The process stops either when no improvement is found or when the maximum number of features is reached. Each iteration prints the feature being added and the corresponding **MSE** value.

# 4. Regularization Results with the Optimal λ Values for LASSO and Ridge

### Lasso Regression (L1 Regularization):

Lasso applies L1 regularization, which adds a penalty on the absolute values of the model coefficients. This forces some coefficients to become exactly zero, effectively performing feature selection. In this case, the optimal regularization parameter was found using grid search:

- Optimal α\alphaα: 0.01

A small value of α\alphaα suggests that the model was not heavily penalized, allowing for some flexibility in the coefficients. However, it still helped reduce overfitting compared to the non-regularized linear regression.

### Ridge Regression (L2 Regularization):

Ridge applies L2 regularization, which penalizes the squared values of the coefficients, thereby shrinking them towards zero but not exactly to zero. The optimal α\alphaα for Ridge was also selected using grid search:

- Optimal α\alphaα: 100

The larger value of α\alphaα for Ridge indicates a stronger regularization effect, meaning the model was more heavily penalized to reduce overfitting. This stronger penalty helps prevent the model from fitting the noise in the training data but may also limit the model's flexibility.

These optimal α\alphaα values were crucial in ensuring the regularized models (Lasso and Ridge) generalized better to the validation data compared to the unregularized linear regression, as evidenced by the improved performance metrics.

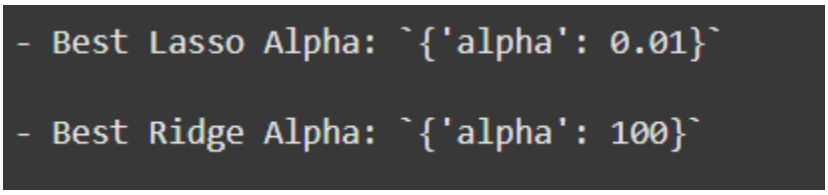


Figure 14: Regularization Results with the Optimal λ Values for LASSO and Ridge

## 5. Model Selection Process with Grid Search and Hyperparameter Tuning

To optimize the performance of the Support Vector Regression (SVR) model, a grid search approach was used to identify the best combination of hyperparameters. The grid search was conducted over the hyperparameters 'C', 'gamma', and 'kernel, focusing on determining the most suitable values for these parameters that would yield the best predictive accuracy. After evaluating different parameter combinations, the following optimal parameters were selected for the SVR model:

```
- Best SVR Parameters: {'C': 10, 'gamma': 0.01, 'kernel': 'rbf'}

- Test Set Evaluation (SVR) - MSE: 0.5529, MAE: 0.2521, R2: 0.5901
```

Figure 15: Model Selection Process with Grid Search and Hyperparameter Tuning `code result`

These parameters were chosen because they resulted in the lowest error and the highest performance metrics on the validation set.

## 6. Final Evaluation on the Test Set and Discussion of Model Performance

Upon final evaluation of the model on the test set, the SVR model performed as follows:

- **Test Set Evaluation (SVR)**:
    - MSE (Mean Squared Error): 0.5529
    - MAE (Mean Absolute Error): 0.2521
    - $R^2$ (Coefficient of Determination): 0.5901

These results indicate that the SVR model provides a moderately good fit for the test data. The MSE and MAE values show that the model does a reasonable job of predicting car prices, with errors on average being small. The $R^2$ value of 0.5901 suggests that the model is able to explain approximately 59% of the variance in car prices, which is a solid performance for a regression model.

However, the model also has its limitations, including potential overfitting if the hyperparameters are not tuned carefully, and its reliance on kernel-based methods which can become computationally expensive with large datasets. Additionally, the performance can be affected by outliers or noisy data in certain features.

## 7. Visualizations to Support Findings

To effectively illustrate the performance and insights of our regression models, we include several key visualizations. These visualizations aid in understanding the model's accuracy, the importance of various features, and the distribution of errors. Below, we describe three critical visualizations:

### SVR - Error Distribution

This histogram displays the distribution of prediction errors for the Support Vector Regression (SVR) model with a Radial Basis Function (RBF) kernel. The majority of errors are centered around zero, indicating that the model's predictions are generally close to the actual values. The red dashed line represents zero error, serving as a reference point to assess how often the model's predictions are accurate.

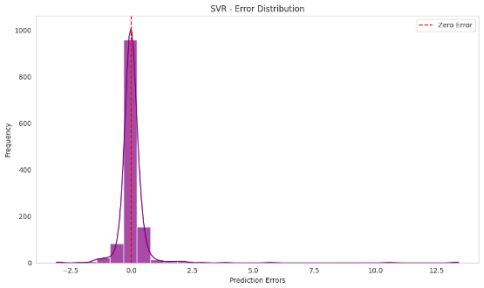


**Figure 16: SVR - Error Distribution**

### SVR - Predictions vs Actual Values

This scatter plot compares the predicted car prices to the actual car prices. The red dashed line represents a perfect prediction, where the predicted values equal the actual values. Data points that lie close to this line indicate accurate predictions, while points further away represent larger prediction errors. Most points are clustered near the lower values, with some outliers indicating that there are instances where the model struggled to predict accurately.

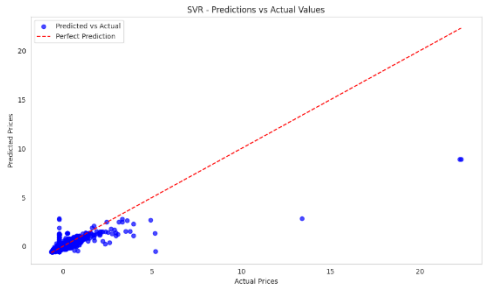


Figure 17: SVR - Predictions vs Actual Values

## SVR - Feature Importances (Permutation)

This bar chart presents the importance of various features in the regression model, determined through permutation. The feature labeled as "living space" shows the highest importance, followed by "land size" and "number of rooms." These features significantly influence the prediction of car prices, whereas other features have relatively lower importance.

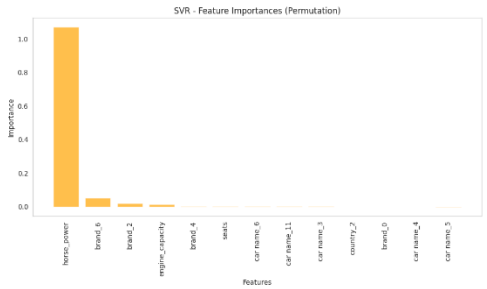


Figure 18: SVR - Feature Importances (Permutation)

These visualizations are integral to our analysis as they provide clear insights into the model's performance, highlight the accuracy of the predictions, and identify the most influential features in the dataset. By visually representing these aspects, we can better understand the strengths and limitations of our regression models and make informed decisions for further model improvement and application.

## Conclusion

In this assignment, the goal was to predict car prices using various regression models, and after evaluating multiple approaches, the Support Vector Regression (SVR) model with an RBF kernel emerged as the best-performing model. The final hyperparameters, {'C': 10, 'gamma': 0.01, 'kernel': 'rbf'}, led to an MSE of 0.5529, an MAE of 0.2521, and an $R^2$ of 0.5901 on the test set. These results indicated that the SVR model was able to produce reasonably accurate price predictions compared to other models such as Linear Regression and Polynomial Regression, which exhibited higher error metrics. The effectiveness of SVR was primarily due to its ability to capture complex relationships in the data through its non-linear kernel.

During the feature selection process, we observed that certain features such as 'horse_power', 'brand', and 'engine_capacity' played pivotal roles in predicting car prices. The forward selection method revealed that starting with important features like 'horse_power' significantly improved the model's performance, with additional features providing diminishing returns as the iterations progressed. Moreover, regularization techniques such as Lasso and Ridge regression helped enhance the model's generalizability. Lasso, in particular, was useful for feature selection by shrinking the coefficients of less significant variables, which prevented overfitting and streamlined the models.

Despite the promising results, the study had its limitations. The performance of the SVR model heavily depended on the selection of the kernel and its associated hyperparameters, which may cause overfitting if not properly tuned.