"بسم الله الرحمن الرحيم"

BIRZEIT UNIVERSITY

Faculty of Engineering and Technology.

Department of Electrical and Computer Engineering.

MACHINE LEARNING AND DATA SCIENCE - ENCS5341.

Assignment #2: Regression Analysis and Model Selection.

Students name (Prepared by): Razan Abdelrahman –1200531.
Hidaya Mustafa - 1201910.

Instructor: Dr. Ismail Khater.

Section: 2.

Date: 28th | November.

1. Dataset.

1.1 Dataset overview.

The dataset used in this assignment contains detailed information about cars, scraped from the YallaMotors website. It consists of approximately **6,750 rows** and **9 features**, making it a rich source for exploratory data analysis and predictive modeling. The primary target variable is *price*, which represents the cost of a car and is influenced by various attributes.

The feature include two types in the dataset:

- Categorical Features:

  Car name: The full name and description of the car.

  Brand: The manufacturer of the car (e.g., Toyota, Honda, Ford).

  Country: The country where the car is sold (e.g., KSA, UAE, USA).

- Numerical Features:

  Price: The target variable for prediction, representing the car's cost.

  Engine capacity: The size of the car's engine in liters.

  Cylinder: The number of engine cylinders or an indicator of electric engines.

  Horsepower: The engine’s horsepower, reflecting performance.

  Top speed: The car's maximum speed.

  Seats: The number of seats in the car.

1.2 Preprocessing Steps.

To prepare the dataset for regression modeling, several preprocessing steps were carried out to handle inconsistencies, missing values, and scaling issues. These steps ensured the data was clean, consistent, and suitable for training accurate models.

❖ Handling missing values.

1- The *seats* column contains missing or inconsistent values, including entries found in the *top_speed* column that contain seating information (e.g., "8 Seater"). To clean this data, seating information was extracted from the *top_speed* column where applicable. The "Seater" text was removed, and the cleaned values were converted into integers. Any remaining missing or invalid values were handled by coercing them into NaN, ensuring the seats column contains only valid numerical data.

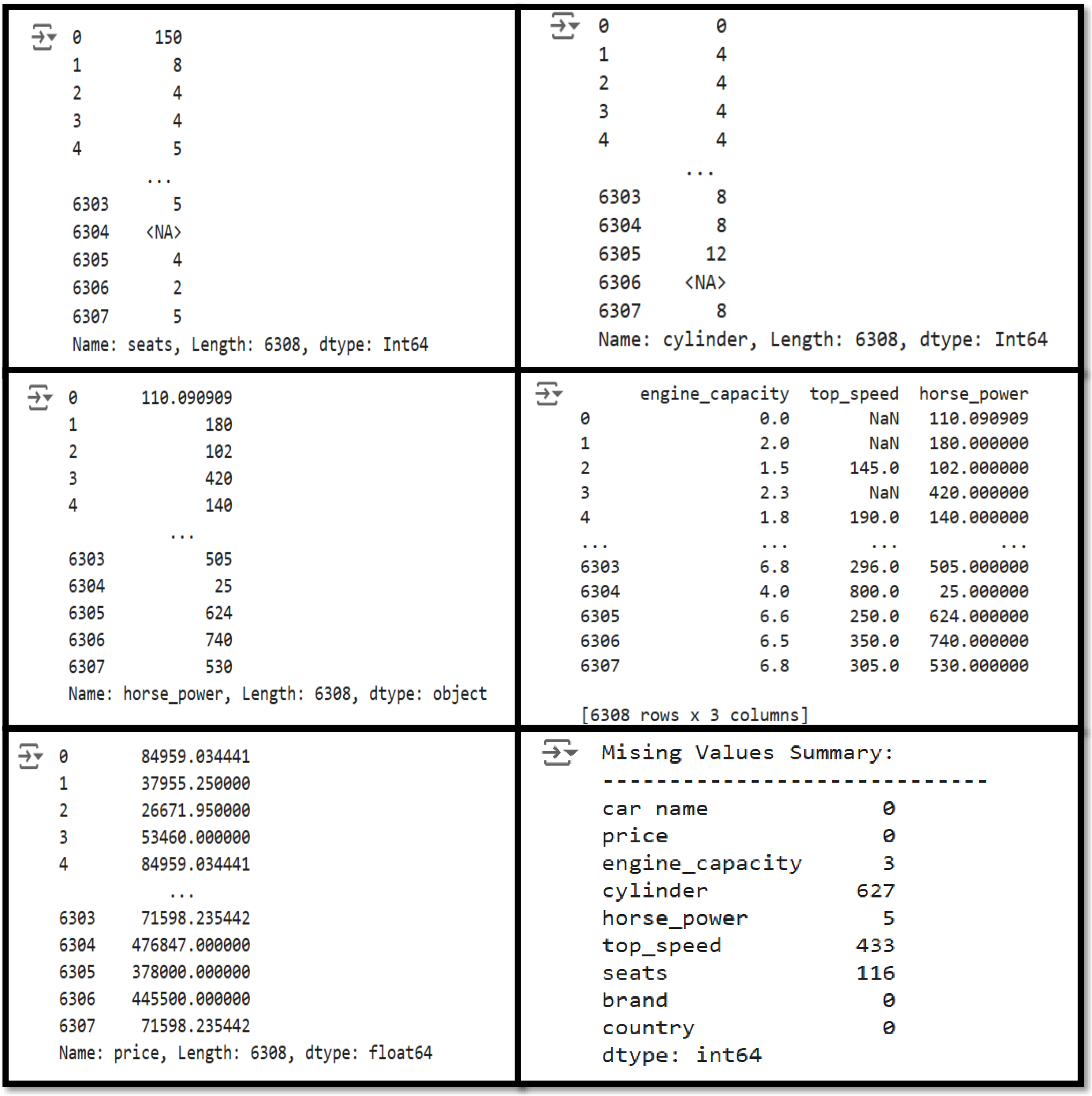
2- The *cylinder* column contains missing or inconsistent values, such as "N/A, Electric" for electric vehicles. To clean this column, the value "N/A, Electric" was replaced with 0, representing the absence of cylinders in electric vehicles. The column was then converted to numeric, coercing invalid entries into NaN. Finally, the data was cast as integers to ensure consistency and proper handling of valid numerical values.

3- The *horse_power* column contains non-numeric entries, such as "Single" and "Double," which represent undefined values. To clean this column, the non-numeric entries were replaced with the brand-specific average horsepower. The column was then converted to numeric values, ensuring consistency and usability for further analysis.

4- Some numerical columns, such as *engine_capacity* and *top_speed,* contained missing or invalid values. These were cleaned by coercing invalid entries into NaN. This ensured that all columns were numeric and consistent for analysis.

5- The *price* column, which is the target variable, was treated with extra care to ensure consistency and accuracy in the predictions. This column contained missing values and inconsistent formatting, with prices represented in multiple currencies. To clean this column: Numeric price values were extracted from the string representation while handling missing or invalid entries, then missing prices were filled using the mean price for the respective country to retain regional trends and variability, at the end all prices were converted to USD using predefined exchange rates to ensure consistency across the dataset.

6- The median was used to impute missing values in columns like *engine_capacity*, *cylinder*, *top_speed*, *seats*, and *horse_power* because it is robust to outliers and accurately represents the central tendency of the data. Unlike the mean, the median minimizes the impact of extreme values and works well for skewed distributions, ensuring the imputed values align with the typical range of each feature. This approach preserves the dataset's overall integrity and stability for further analysis and modeling.



```
0        150
1          8
2          4
3          4
4          5
        ...
6303       5
6304    <NA>
6305       4
6306       2
6307       5
Name: seats, Length: 6308, dtype: Int64
```

```
0          0
1          4
2          4
3          4
4          4
        ...
6303       8
6304       8
6305      12
6306    <NA>
6307       8
Name: cylinder, Length: 6308, dtype: Int64
```

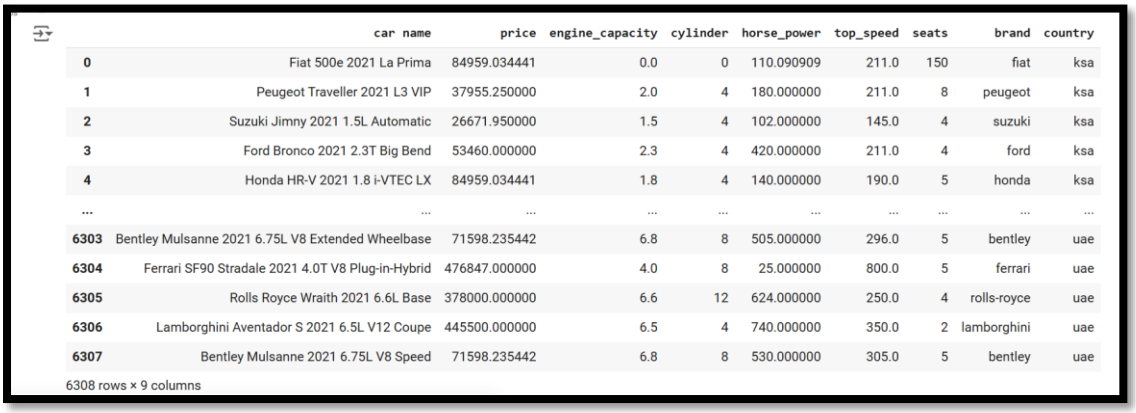
```
0       110.090909
1              180
2              102
3              420
4              140
           ...
6303           505
6304            25
6305           624
6306           740
6307           530
Name: horse_power, Length: 6308, dtype: object
```

```
      engine_capacity  top_speed  horse_power
0                 0.0        NaN   110.090909
1                 2.0        NaN   180.000000
2                 1.5      145.0   102.000000
3                 2.3        NaN   420.000000
4                 1.8      190.0   140.000000
...               ...        ...          ...
6303              6.8      296.0   505.000000
6304              4.0      800.0    25.000000
6305              6.6      250.0   624.000000
6306              6.5      350.0   740.000000
6307              6.8      305.0   530.000000

[6308 rows x 3 columns]
```

```
0        84959.034441
1        37955.250000
2        26671.950000
3        53460.000000
4        84959.034441
            ...
6303     71598.235442
6304    476847.000000
6305    378000.000000
6306    445500.000000
6307     71598.235442
Name: price, Length: 6308, dtype: float64
```

```
Missing Values Summary:
----------------------------
car name             0
price                0
engine_capacity      3
cylinder           627
horse_power          5
top_speed          433
seats              116
brand                0
country              0
dtype: int64
```

Figure 1 The steps of Handling the missing values in the dataset.



	car name	price	engine_capacity	cylinder	horse_power	top_speed	seats	brand	country
0	Fiat 500e 2021 La Prima	84959.034441	0.0	0	110.090909	211.0	150	fiat	ksa
1	Peugeot Traveller 2021 L3 VIP	37955.250000	2.0	4	180.000000	211.0	8	peugeot	ksa
2	Suzuki Jimny 2021 1.5L Automatic	26671.950000	1.5	4	102.000000	145.0	4	suzuki	ksa
3	Ford Bronco 2021 2.3T Big Bend	53460.000000	2.3	4	420.000000	211.0	4	ford	ksa
4	Honda HR-V 2021 1.8 i-VTEC LX	84959.034441	1.8	4	140.000000	190.0	5	honda	ksa
...	...	...	...	...	...	...	...	...	...
6303	Bentley Mulsanne 2021 6.75L V8 Extended Wheelbase	71598.235442	6.8	8	505.000000	296.0	5	bentley	uae
6304	Ferrari SF90 Stradale 2021 4.0T V8 Plug-in-Hybrid	476847.000000	4.0	8	25.000000	800.0	5	ferrari	uae
6305	Rolls Royce Wraith 2021 6.6L Base	378000.000000	6.6	12	624.000000	250.0	4	rolls-royce	uae
6306	Lamborghini Aventador S 2021 6.5L V12 Coupe	445500.000000	6.5	4	740.000000	350.0	2	lamborghini	uae
6307	Bentley Mulsanne 2021 6.75L V8 Speed	71598.235442	6.8	8	530.000000	305.0	5	bentley	uae

6308 rows × 9 columns

Figure 2 The dataset after completing the cleaning process.

❖ Outliers.

Outliers were detected and handled to ensure that extreme values do not distort the analysis or model predictions. Outliers in numerical columns can bias statistical measures and lead to overfitting in regression models.

Outliers in the dataset were handled to ensure they did not distort the analysis or model predictions. Initial boxplots identified outliers in numerical columns like *price*, *engine_capacity*, *cylinder*, and *horse_power*. Unrealistic values in the seats column (e.g., above 20) were replaced with random integers between 10 and 20 to maintain realistic variability. For other numerical columns, the Interquartile Range (IQR) method was used to clip outliers to the nearest valid boundary. Post-clipping boxplots confirmed that the outliers were effectively handled, ensuring the dataset was clean and ready for modeling.

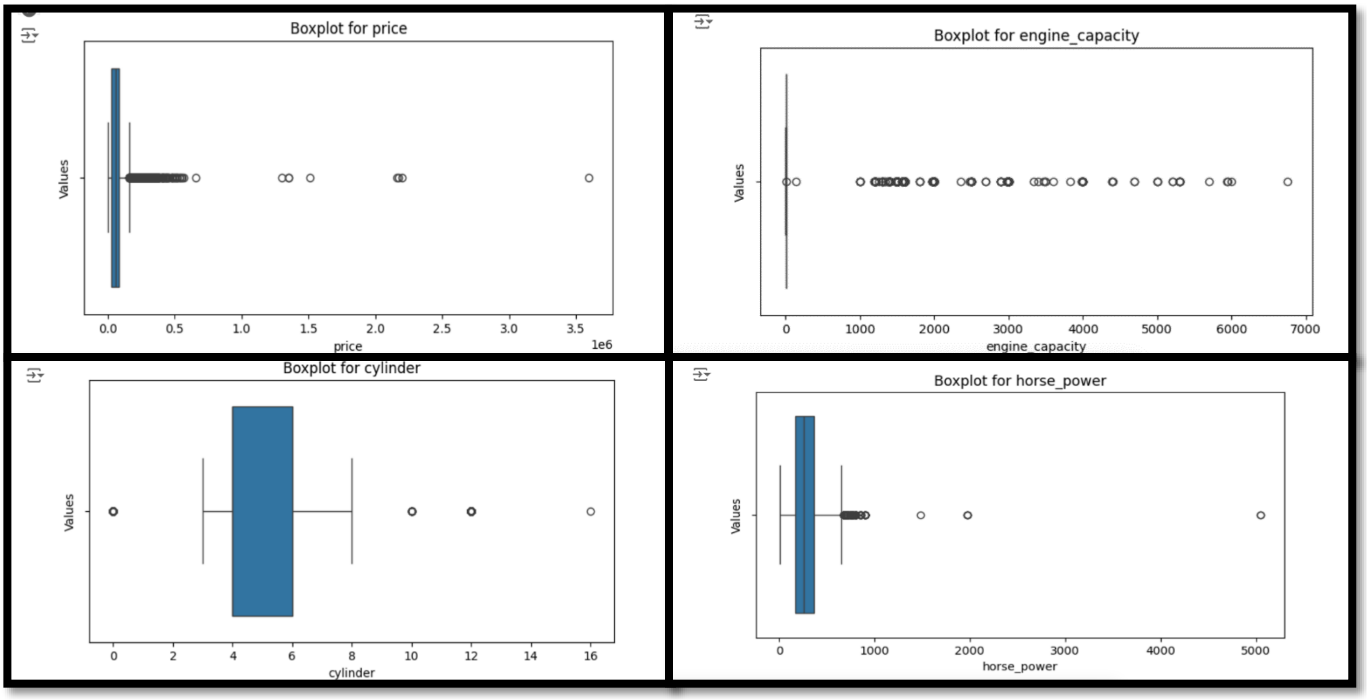


Figure 3 Boxplots of numerical features before handling outliers.

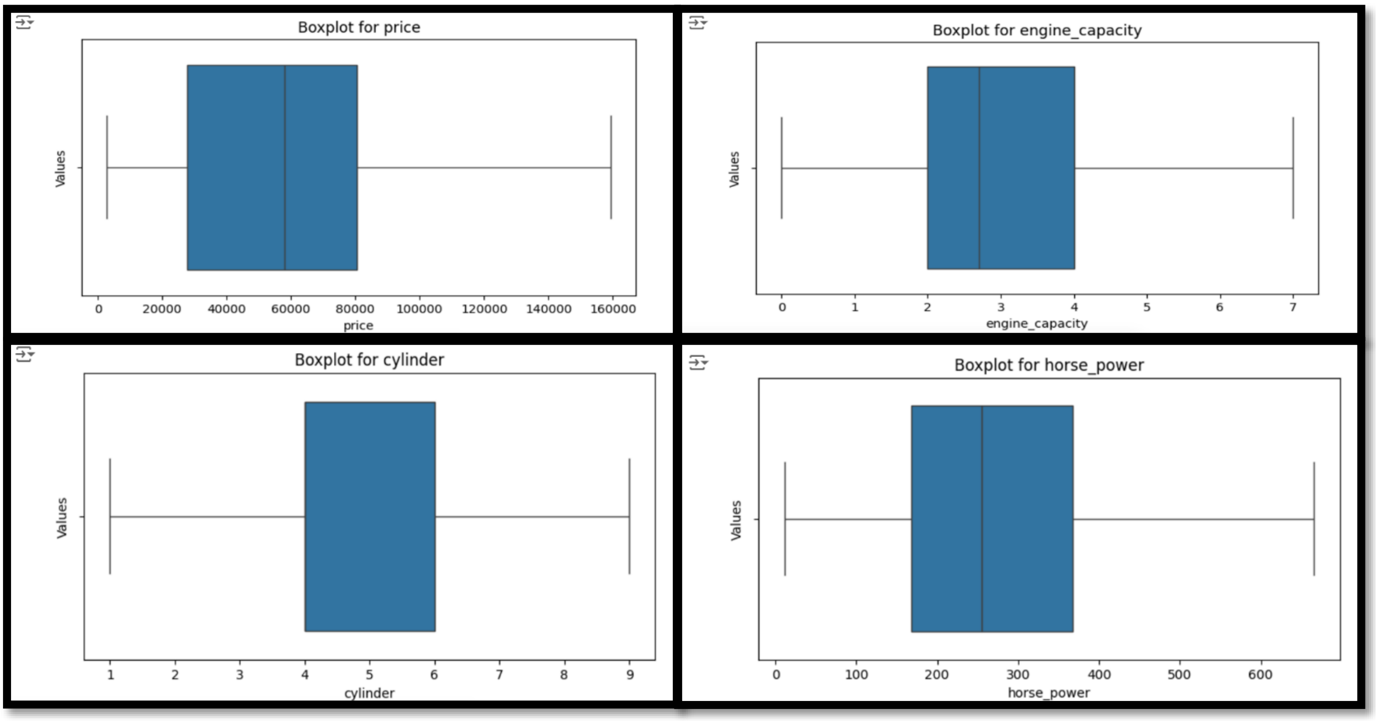


Figure 4 Boxplots of numerical features after handling outliers.

❖ Encoding categorical features.

Categorical features in the dataset, such as *country* and *brand*, were encoded to make them suitable for numerical analysis by machine learning models. This step ensured that the categorical data was represented in a format that the models could process while retaining the relationships between categories.

- One-hot encoding was applied to the country column, creating binary features for each country, which avoids introducing unnecessary relationships between the categories.
- Label encoding was used for the brand column, efficiently converting each brand into a unique integer label.

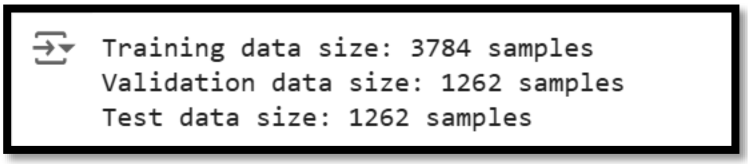
❖ Normalizing.

To ensure consistent scaling across features, numerical columns were normalized using z-score normalization via the *StandardScaler*. This step standardized the columns to have a mean of 0 and a standard deviation of 1, which helps prevent features with larger ranges (e.g., *price*) from dominating those with smaller ranges (e.g., *engine_capacity*) during model training.

❖ Splitting Data.

To build and evaluate regression models effectively, the dataset was split into three subsets: training, validation, and test sets. This ensures that:

- Training Set is used to fit the model.
- Validation Set is used to tune hyper parameters and select the best model.
- Test Set is used for final evaluation to assess the model's generalization performance.



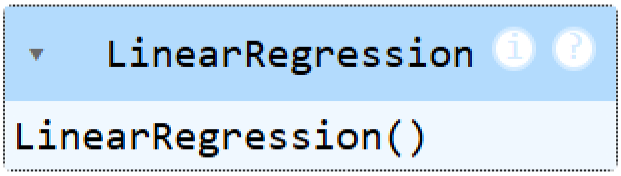
```
Training data size: 3784 samples
Validation data size: 1262 samples
Test data size: 1262 samples
```

2. Building Regression Models.

2.1 Linear Models.

2.1.1 Built-in Linear Regression.

Built-in linear regression was implemented using the *LinearRegression* class from *sklearn*. This model serves as a baseline to compare with custom implementations like the closed-form solution and gradient descent. The model was trained on the *X_train* and *y_train* datasets, learning coefficients to minimize the Mean Squared Error (MSE). Its performance will be evaluated on validation and test sets to assess accuracy and generalization.



```
LinearRegression
LinearRegression()
```

2.1.2 Build Linear Regression.

- ***The closed-form solution:*** for linear regression provides a direct way to compute the optimal weights for a regression model by minimizing the mean squared error. It uses the following equation:

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

The function calculates the weights for linear regression by first adding a bias term (a column of ones) to the feature matrix to account for the intercept. It then computes the product of the transposed feature matrix and the feature matrix, adjusts it with a small regularization term (λ\lambdaλ) to stabilize the inversion process, and ensures numerical stability to prevent overfitting. Finally, the regularized matrix is inverted to compute the optimal weights efficiently.

- ***The gradient descent method:*** is an iterative optimization technique used to train the linear regression model by minimizing the error between predicted and actual values. A bias term is incorporated into the features, and the initial weights are set to zero. In each iteration, the gradient of the cost function (error) is computed to determine the direction of optimization, and the weights are updated based on the learning rate. This method allows flexibility and scalability for datasets where direct computation of weights through matrix inversion (closed-form solution) may not be practical. By adjusting the learning rate and iteration count, this method can achieve robust predictions.

2.1.3 LASSO (L1 Regularization).

LASSO regression incorporates L1 regularization, which penalizes the magnitude of feature coefficients, effectively performing feature selection by shrinking some coefficients to zero. To optimize the regularization strength (alpha), a grid search approach is used with cross-validation, testing multiple values of alpha to minimize the mean squared error. This allows for identifying the most effective trade-off between model complexity and performance. The selected alpha value ensures the model is both accurate and efficient by focusing on the most influential predictors.

2.1.4 Ridge Regression (L2 Regularization).

Ridge regression applies L2 regularization, which penalizes the squared magnitude of coefficients to prevent overfitting and improve generalization. Unlike LASSO, Ridge retains all features by shrinking coefficients rather than setting them to zero. Grid search with cross-validation is used to optimize the regularization parameter (alpha) by testing a range of values to minimize mean squared error. This ensures the model balances predictive accuracy with feature coefficient stability.

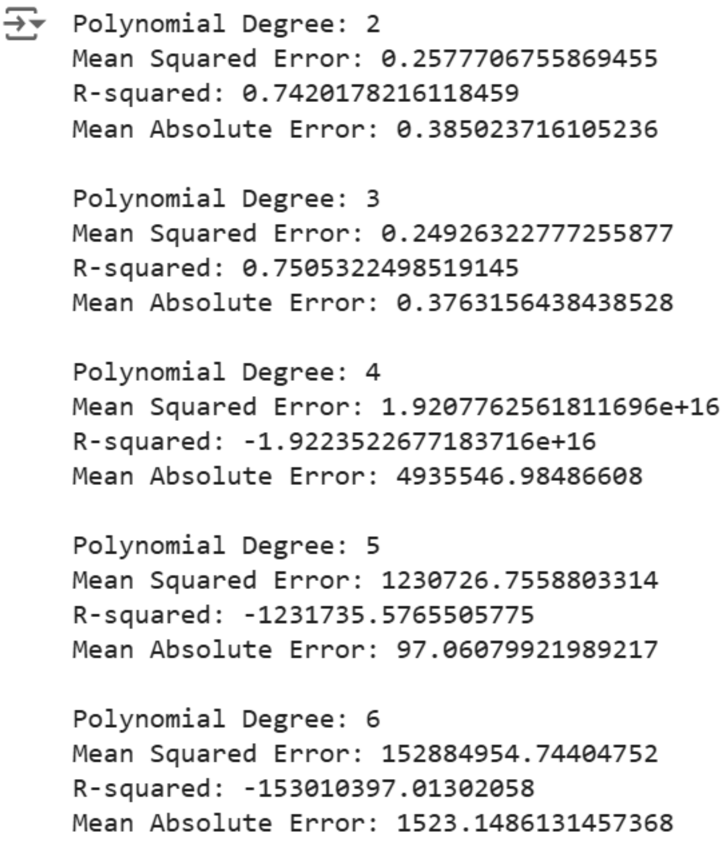
2.2 Non-Linear Models.

Polynomial regression introduces non-linearity to the model by transforming the features into polynomial terms of varying degrees. Degrees from 2 to 6 were tested to evaluate the model's ability to capture complex relationships in the data.

For each degree:

- The training features were expanded to include polynomial terms using the ***PolynomialFeatures*** class.
- The model was trained using these transformed features and validated on unseen data.

- Performance was assessed using Mean Squared Error (MSE), R-squared (R^2), and Mean Absolute Error (MAE).



```
Polynomial Degree: 2
Mean Squared Error: 0.2577706755869455
R-squared: 0.7420178216118459
Mean Absolute Error: 0.385023716105236

Polynomial Degree: 3
Mean Squared Error: 0.24926322777255877
R-squared: 0.7505322498519145
Mean Absolute Error: 0.3763156438438528

Polynomial Degree: 4
Mean Squared Error: 1.9207762561811696e+16
R-squared: -1.9223522677183716e+16
Mean Absolute Error: 4935546.98486608

Polynomial Degree: 5
Mean Squared Error: 1230726.7558803314
R-squared: -1231735.5765505775
Mean Absolute Error: 97.06079921989217

Polynomial Degree: 6
Mean Squared Error: 152884954.74404752
R-squared: -153010397.01302058
Mean Absolute Error: 1523.1486131457368
```

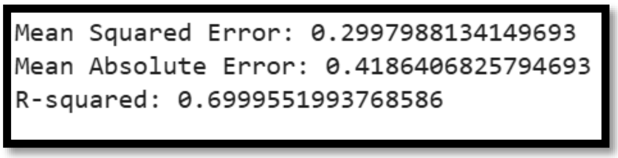
The results suggest that a degree 2 or 3 polynomial provides the best balance between accuracy and complexity. Higher-degree models over fit the data, leading to poor generalization and inflated errors. Polynomial regression should be used cautiously, as increasing complexity does not always lead to better performance.

3. Model Selection Using Validation Set.

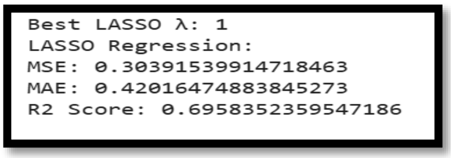
To select the best-performing regression model, each model was evaluated on the validation set using the following metrics:

- Mean Squared Error (MSE): Measures the average squared difference between predicted and actual values. Lower MSE indicates better accuracy.
- Mean Absolute Error (MAE): Represents the average magnitude of errors, with lower values reflecting better performance.

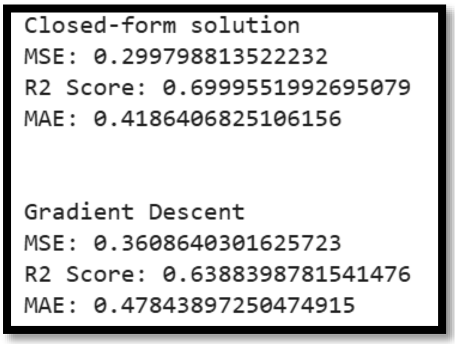
- R-squared (R^2): Indicates the proportion of variance explained by the model, with higher values showing a better fit.



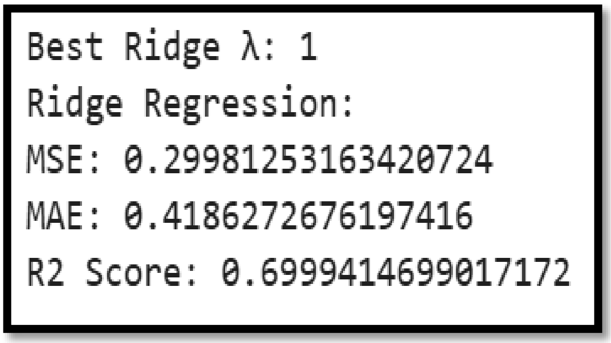
```
Mean Squared Error: 0.2997988134149693
Mean Absolute Error: 0.4186406825794693
R-squared: 0.6999551993768586
```



```
Best LASSO λ: 1
LASSO Regression:
MSE: 0.30391539914718463
MAE: 0.42016474883845273
R2 Score: 0.6958352359547186
```



```
Closed-form solution
MSE: 0.299798813522232
R2 Score: 0.6999551992695079
MAE: 0.4186406825106156


Gradient Descent
MSE: 0.3608640301625723
R2 Score: 0.6388398781541476
MAE: 0.47843897250474915
```



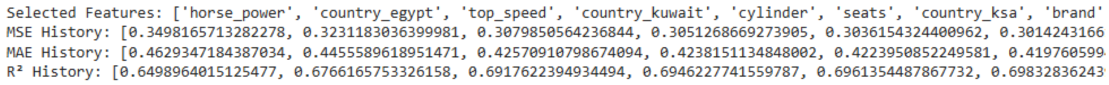
```
Best Ridge λ: 1
Ridge Regression:
MSE: 0.29981253163420724
MAE: 0.4186272676197416
R2 Score: 0.6999414699017172
```

The built-in linear regression, closed-form solution, and ridge regression models performed the best, with nearly identical results. Gradient descent showed slightly lower performance due to its iterative nature. LASSO regression offered competitive results while emphasizing feature selection through regularization. For this dataset, linear models with minimal regularization provided the most reliable predictions.

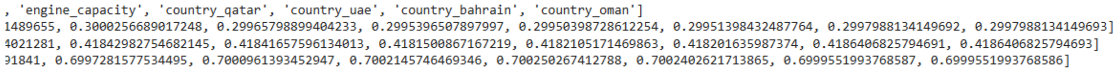
4. Feature Selection with Forward Selection.

Forward Feature Selection is a methodical process to identify the most important features for predictive modeling. Starting with an empty set of features, the method iteratively adds one feature at a time, evaluating the performance of the model with the inclusion of each feature. At every iteration, the feature that minimizes the Mean Squared Error (MSE) on the validation set is added to the model.

This process is repeated until adding more features no longer improves performance or until a predetermined number of features is selected. By monitoring the Mean Absolute Error (MAE), MSE, and R-squared (R^2) metrics, we can observe how the model's performance changes with the addition of features.



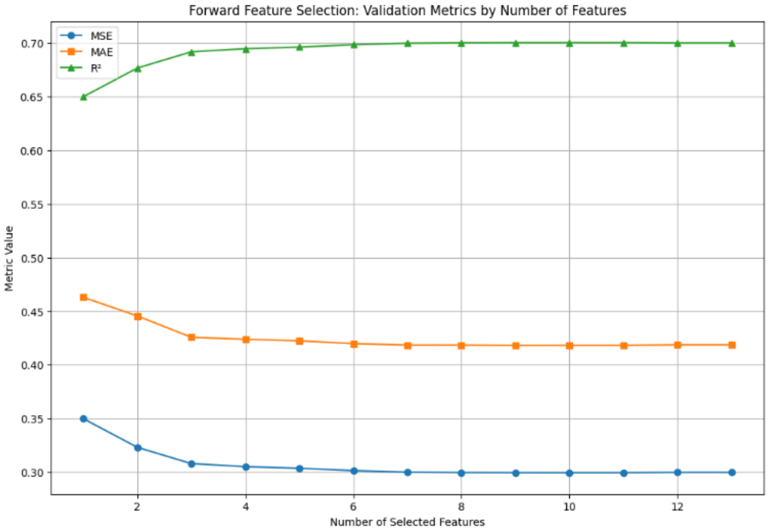
```
Selected Features: ['horse_power', 'country_egypt', 'top_speed', 'country_kuwait', 'cylinder', 'seats', 'country_ksa', 'brand'
MSE History: [0.3498165713282278, 0.3231183036399981, 0.3079850564236844, 0.3051268669273905, 0.3036154324400962, 0.3014243166
MAE History: [0.4629347184387034, 0.4455589618951471, 0.42570910798674094, 0.4238151134848002, 0.4223950852249581, 0.419760599
R² History: [0.6498964015125477, 0.6766165753326158, 0.6917622394934494, 0.6946227741559787, 0.6961354487867732, 0.69832836243
```



```
, 'engine_capacity', 'country_qatar', 'country_uae', 'country_bahrain', 'country_oman']
1489655, 0.3000256689017248, 0.29965798899404233, 0.2995396507897997, 0.29950398728612254, 0.29951398432487764, 0.2997988134149692, 0.2997988134149693]
4021281, 0.41842982754682145, 0.41841657596134013, 0.4181500867167219, 0.4182105171469863, 0.418201635987374, 0.4186406825794691, 0.4186406825794693]
91841, 0.6997281577534495, 0.7000961393452947, 0.7002145746469346, 0.700250267412788, 0.7002402621713865, 0.6999551993768587, 0.6999551993768586]
```

The results of the forward feature selection process highlight the step-by-step improvement in model performance as more features are added. The selected features, such as 'horse_power', 'top_speed', and country-specific indicators, showcase their importance in predicting car prices.

- MSE: The Mean Squared Error shows a significant reduction during the initial steps, indicating that early-selected features contribute the most to reducing prediction errors. However, the improvement plateaus as more features are added, suggesting diminishing returns.
- MAE: Similar to MSE, the Mean Absolute Error decreases consistently but stabilizes, confirming that additional features contribute marginally after a certain point.
- R²: The R-squared value steadily increases, approaching its maximum, demonstrating that the model explains a greater proportion of the variance in the target variable as more features are included.



The visualization effectively captures these trends. The rapid improvement in metrics during the initial iterations emphasizes the significance of the first few features.

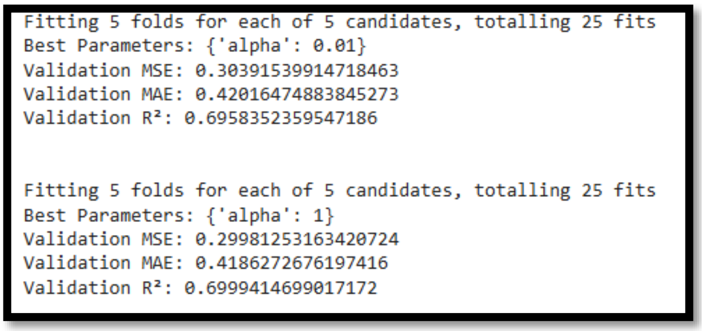
However, as the number of features increases, the curves flatten, reinforcing the notion of diminishing returns and suggesting an optimal feature subset. Overall, forward selection balances model complexity and performance, ensuring a robust feature set without overfitting.

5. Hyper parameter tuning with Grid Search.

In this section, hyper parameter tuning is conducted using grid search to optimize the models for better performance. This ensures that the models are neither under-regularized nor over-regularized, leading to balanced and interpretable predictions.

For Lasso and Ridge regression, the alpha parameter is tested over a range of values. Cross-validation ensures robust evaluation by splitting the data into multiple subsets for training and testing. The best model is selected based on the lowest Mean Squared Error (MSE) on the validation set.

For Polynomial Regression, a pipeline is created to integrate polynomial transformations with Lasso regression. By systematically varying the polynomial degree (from 2 to 10) and tuning the alpha parameter, the model identifies the optimal degree that minimizes validation MSE.

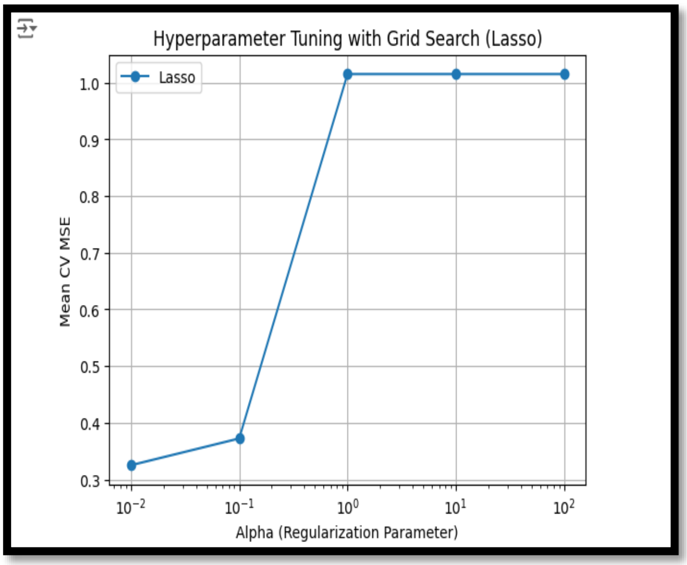


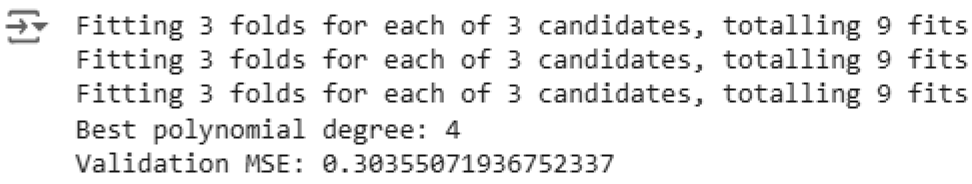
```
Fitting 5 folds for each of 5 candidates, totalling 25 fits
Best Parameters: {'alpha': 0.01}
Validation MSE: 0.30391539914718463
Validation MAE: 0.42016474883845273
Validation R²: 0.6958352359547186


Fitting 5 folds for each of 5 candidates, totalling 25 fits
Best Parameters: {'alpha': 1}
Validation MSE: 0.29981253163420724
Validation MAE: 0.4186272676197416
Validation R²: 0.6999414699017172
```

The results of the hyper parameter tuning for Lasso and Ridge regression show that both models were optimized using grid search to identify the best regularization parameter (α\alphaα).

The visualization displays the relationship between the regularization parameter (α\alphaα) and the Mean Cross-Validation MSE for the Lasso regression model during hyper parameter tuning.

The plot confirms that selecting α=0.01\alpha = 0.01α=0.01 ensures the Lasso model achieves optimal performance by minimizing validation MSE while avoiding overfitting.





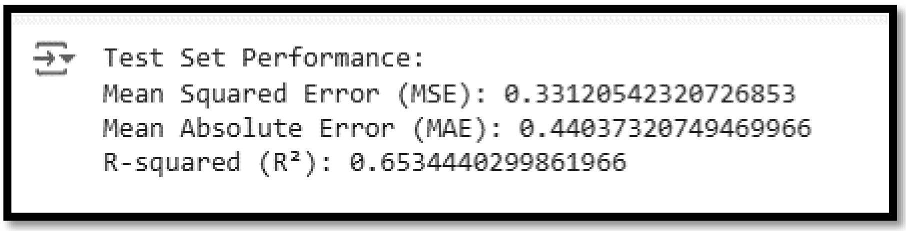
```
Fitting 3 folds for each of 3 candidates, totalling 9 fits
Fitting 3 folds for each of 3 candidates, totalling 9 fits
Fitting 3 folds for each of 3 candidates, totalling 9 fits
Best polynomial degree: 4
Validation MSE: 0.30355071936752337
```

The results indicate that the best polynomial degree for this dataset is 4, with a validation MSE of 0.3036. This degree strikes a balance between capturing the complexity of the data and avoiding overfitting. A smaller or larger polynomial degree did not yield better performance, likely due to under fitting or overfitting.

The reduced parameter grid for α ([0.1, 1, 10]) and the smaller range for polynomial degrees (2–4) helped improve computational efficiency while still allowing us to explore meaningful hyper parameter combinations. This ensures that the final model is both accurate and computationally efficient.

6. Model Evaluation on Test Set.

After determining the best model using the validation set, we evaluated its performance on the test set to gauge its generalization ability. Metrics such as Mean Squared Error (MSE), Mean Absolute Error (MAE), and R-squared (R^2) were calculated to summarize the model's predictive accuracy. The test set represents unseen data, ensuring that the model's evaluation is unbiased and indicative of its real-world performance.



```
Test Set Performance:
Mean Squared Error (MSE): 0.33120542320726853
Mean Absolute Error (MAE): 0.44037320749469966
R-squared (R²): 0.6534440299861966
```

The test set evaluation provides a final assessment of the selected model's ability to generalize to unseen data.

- Mean Squared Error (MSE):

  The MSE of 0.331 indicates the average squared difference between the predicted and actual values. While relatively low, it reflects the model's ability to minimize significant errors, but it is sensitive to outliers.

- Mean Absolute Error (MAE):

The MAE of 0.440 suggests that, on average, the model's predictions deviate by approximately 0.44 units from the actual values. This metric provides an intuitive sense of the model's error magnitude, as it is not influenced by large outliers.

- R-squared (R^2):
  Score of 0.653 shows that approximately 65.3% of the variance in the target variable (car prices) is explained by the model's predictions. While this indicates a moderate fit, there is room for improvement, suggesting that additional features or different model architectures could enhance predictive performance.

7. Collaboration and Contributions

This project was completed collaboratively, we are equally contributing to the development of the code and the preparation of the report. All steps were discussed and implemented during team meetings. These collaborative sessions ensured that both members actively participated in writing, debugging, and refining the code, as well as drafting and structuring the report.