Faculty of Engineering and Technology
Department of Electrical and Computer Engineering

Machine Learning and Data Science - ENCS5341

Assignment #2

Regression

Prepared by:

- Ali Shaikh Qasem ID: 1212171
- Abdelrahman Jaber ID: 1211769

Instructors: Dr. Ismail Khater, Dr. Yazan Abu Farha.

Sections: 1,3.

Date: 28/11/2024

Birzeit University

2024-205

Table of Contents

1 List of Tables

2 List of Figures

2. Description of the dataset, preprocessing steps, and features used

This data was scrapped from YallaMotors website with Python and Requests-html , it contains about 6750 rows with 9 columns , and it is perfect for Exploratory Data Analysis and Machine Learning Algorithms.

The main objective of this dataset is to predict car prices, making it ideal for developing regression models to understand the relationship between various features (e.g., car make, model, year, mileage, engine size, etc.) and the target variable (car price).

Table 2-1: Info of the dataset

Column	Non-Null Count	Dtype
car name	6308	object
price	6308	object
Engine capacity	6308	object
cylinder	5684	object
Horsepower	6308	object
Top speed	6308	object
seats	6308	object
brand	6308	object
country	6308	object

We have in the dataset 6308 entries, from 0 to 6307.

For Preprocessing we convert the value of the prices such that all valid prices are standardize into USD and we convert all numeric features to appropriate data type with converting any illegal value to "NaN".

For missing data we assume if the feature is numeric, then replace the missing value with **mean**, else replace it with **mode**.

We used **label encoding** to apply the label encoder to categorical data.

And for **normalization** we used MinMax normalization with the range 0 to 1.

We Also, did split the data to 60% training ,20% validation and 20% testing sets.

We select cylinder, top speed, engine capacity, country, brand and car name features using forward feature selection which will be cover later.

3 Details of each regression model and its performance on the validation set.

For each regression model we used the training set to build the model then we used the validation set to test each model.

3.1 Linear regression in closed form

We built this model using closed form solution.

And its performance on the validation set was as follows:

MSE is: 2979501283, MAE is: 28313 and R2 is: 0.503.

Model performance visualization:

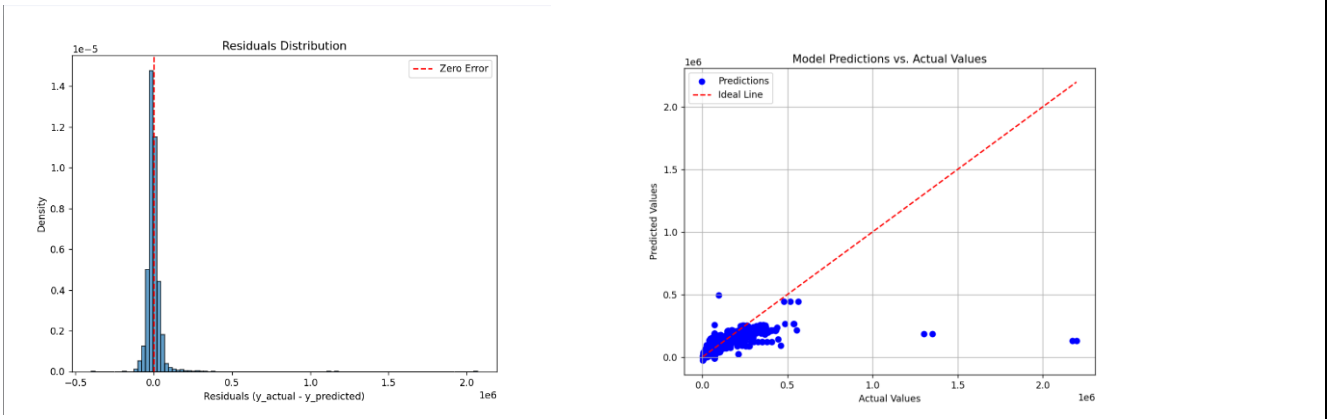


Figure 3-1: error distribution, predicted vs actual output for the model

3.2 Linear regression gradient descent

We built this model using gradient decent with 1000 iterations.

MSE is: 3003867620, MAE is: 28403, and R2 is: 0.499.

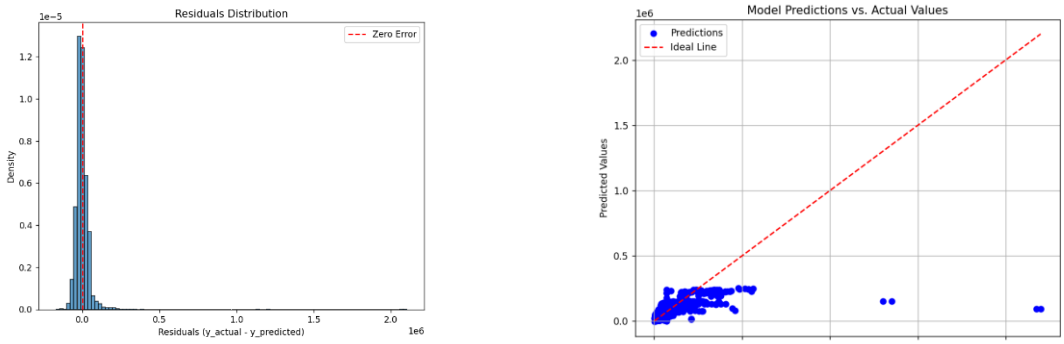


Figure 3-2: error distribution, predicted vs actual output for the model

3.3 LASSO regression

This regression type we used Lasso API because it doesn't have an equation.

But we did tune for alpha hyper parameter.

MSE is: 3975409571, MAE is: 43977, R2 is: 0.34.

The performance of the model is to be discussed in the next points.

3.4 Ridge regression

This regression model we did built it by using closed form solution.

And we made tuning for its alpha hyperparameter.

MSE is: 2981145014, MAE is: 28307 and R2 is: 0.503.

The performance of the model is to be discussed in the next points.

3.5 Polynomial regression

For the model we did use monomial basis function. And we did tune for the degree which gives us 9 for the best degree.

MSE is: 116193, MAE is: 247 and R2 is: 0.99998.

3.6 Gaussian regression

In this model we used radial basis Function from scratch.

And we tuned for sigma, Also we did use k-means to find cluster centers.

MSE is: 2993420932, MAE is: 28237, and R2 is: 0.504

4 Feature Selection

In this part, we applied feature selection using forward selection method.

The procedure used in the code is described as follows:

1. We start with an empty set of features.
2. in every iteration, we add the feature, that when added, improves the performance of the model.
3. Stop when the model performance is not improved.

The original data set contains the following features:

{car name, price, engine capacity, *cylinder,* horsepower, top speed, seats, brand, country}

The forward selection method is applied on the closed form linear regression to determine the best features that describes the model efficiently, the metric used to determine the performance is the R-squared metric.

The following table represents the feature selected at each step:

Table 4-1: feature selection details

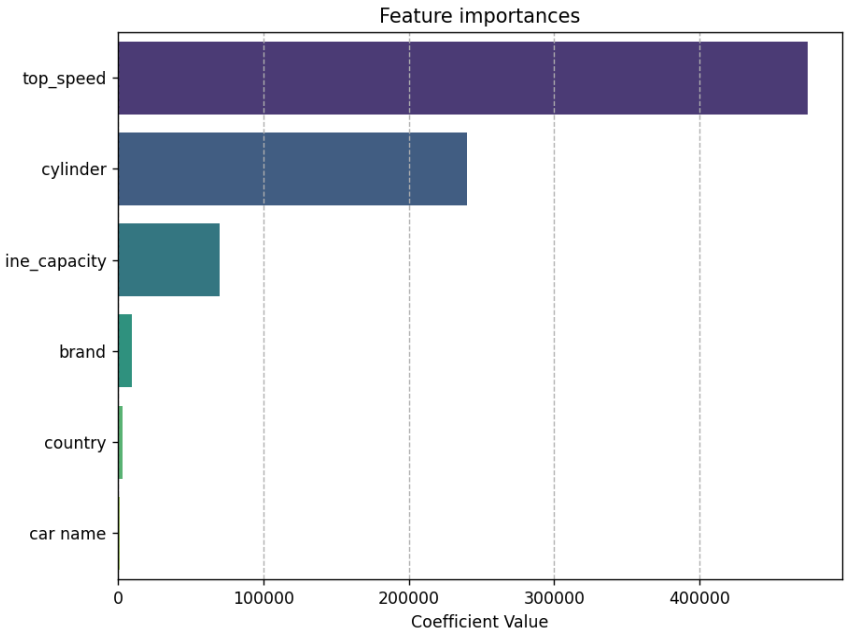
Iteration	**Chosen feature**	**R2 for the model**
1	Cylinder	0.3763
2	Top-speed	0.4561
3	Engine-capacity	0.4685
4	Country	0.4856
5	Brand	0.4884
6	Car name	0.4886

From the above results, we notice that the following features are chosen from the dataset:

{Cylinder, Top-speed, Engine-capacity, Country, Brand, Car name}

The features selected above can be explained using feature importance plot as shown below:

Figure 4-1: feature importances plot



5 Regularization results with the optimal λ values for LASSO and Ridge.

We used Lasso and ridge to control overfitting. So, we can use the model on other data.

The procedures used in the code is described as follows:

1. Saves values from 10^{-3} to 10^{3} with 50 different number.
2. Use R2 to evaluate the best value for the hyperparameter.
3. Use the validation set to train the hyperparameter.
4. Performing **k-fold cross-validation** for each alpha.
5. Use the Grid Search to find the best hyperparameter.

We got for Lasso alpha the value which equals 59.64 and for Ridge alpha the value which equals 0.123.

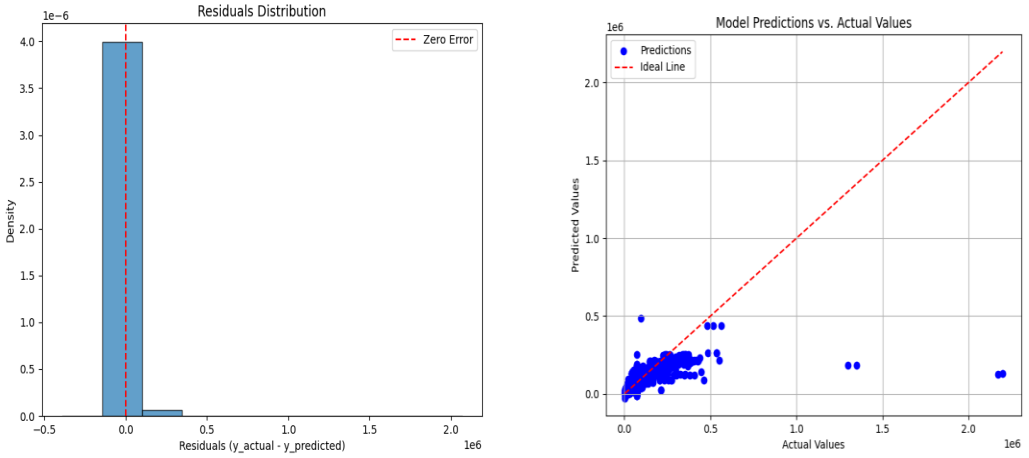


Figure 5-1:error distribution, predicted vs actual output for the Ridge model

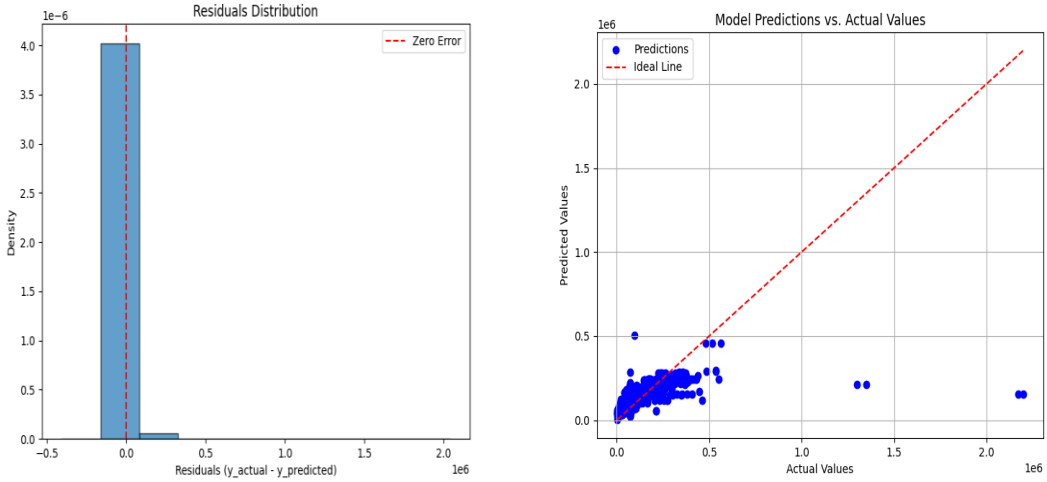


Figure 5-2: Figure 4:error distribution, predicted vs actual output for the Lasso model

6 Model selection process with grid search and hyperparameter tuning.

In this section we selected the best model after tuning the hyperparameter for each model.

We did tune hyperparameter for Ridge, Lasso, degree of polynomial, learning rate for gradient decent and sigma for Gaussian.

The best degree of polynomial was 9.

The value of learning rate was 0.452.

And the value of sigma was 7.9.

After that, to select the best model we evaluated the performance of all candidate models using the optimal hyperparameters on the validation dataset. For each model, we compared their results based on key performance metrics such as R^2, MSE, and MAE to determine which model generalized best to unseen data.

Polynomial regression with a degree of **9** delivered the highest R^2 but was prone to overfitting.

But we did test the polynomial on the testing set and it was excellent.

7 Evaluation on the test set and a discussion of the selected model's performance and limitations.

The models were evaluated on the testing set to assess their generalization performance. Below are the results and a discussion of the selected model's performance and limitations.

Table 2: Model Evaluation Results

Model	MSE	MAE	R2
Linear Regression (Closed Form)	14,735,032,947	30,743	0.2261
Linear Regression (Gradient Descent)	14,777,964,488	30,890	0.2238
LASSO Regression	15,451,762,091	46,999	0.1884
Ridge Regression	14,741,680,336	30,738	0.2257
Polynomial Regression (Degree 9)	7,793	52.90	0.9999996
Gaussian Regression	14,857,278,616	31,041	0.2196

Selected Model: Polynomial Regression

Polynomial regression delivered exceptional results on both the validation and test sets, with MSE and MAE orders of magnitude lower than those of other models. Its R2 value indicates it predicted nearly all variability in the data.

Performance:

- The model can accurately capture complex nonlinear relationships in the data.
- High performance in terms of prediction accuracy as indicated by evaluation metrics.

Limitations:

- The near-perfect fit suggests the model memorized the training data rather than generalizing to unseen data
- Further evaluation on out-of-sample data or cross-validation is required to confirm robustness.

Model performance on test set: