

**Birzeit University**

**Faculty of Engineering and Technology**

**Department of Computer Systems Engineering**

**MACHINE LEARNING AND DATA SCIENCE**

**ENCS5341**

**EXP #5 Filters**

**Report #2**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Prepared by:**

Aws Shaheen 1212585

Ghassan Qandeel 1212397

**Instructor:** Dr. Yazan Abu Farha

**Section:** 1

**Date:** 28-11-2024

# Abstract

This assignment aims to build a series of regression models using a dataset, evaluate and compare their performance, and apply various techniques to improve model accuracy and prevent overfitting. It focuses on both linear and nonlinear registration model, feature selection methods and regularization techniques will be used also followed by hyperparameter tuning, to select the optimal model.

**Table of Contents**

[Abstract I](#_Toc173621055)

[1. Theory 1](#_Toc173621056)

[1.1. Passive Filters: 1](#_Toc173621057)

[1.1.1. First order Filters: 1](#_Toc173621058)

[1.1.2. Second order Filters: 2](#_Toc173621059)

[1.2. Active Filters: 2](#_Toc173621060)

[2. Procedure and data analysis: 3](#_Toc173621061)

[2.1. First order circuits: 3](#_Toc173621062)

[2.2. Second Order Filters: 4](#_Toc173621063)

[2.3. Active Filters: 6](#_Toc173621064)

[Conclusion 8](#_Toc173621065)

[References 9](#_Toc173621066)

[Appendix 10](#_Toc173621067)

**Table of Figures**

[Figure 1: Passive Filters[1] 1](#_Toc173621080)

[Figure 2:First Order Filter [2] 1](#_Toc173621081)

[Figure 3: second order filter[3] 2](#_Toc173621082)

[Figure 4: active filter[4] 2](#_Toc173621083)

[Figure 5:First order circuit 3](#_Toc173621084)

[Figure 6:input & output signals at fc 4](#_Toc173621085)

[Figure 7:second order filter 4](#_Toc173621086)

[Figure 8: input&output signals at fo 5](#_Toc173621087)

[Figure 9:input&output signals at fc1 6](#_Toc173621088)

[Figure 10:input&output signals at fc2 6](#_Toc173621089)

[Figure 11:Active filter circuit 6](#_Toc173621090)

[Figure 12:output voltage of the active filter circuit 7](#_Toc173621091)

**Table of Tables**

Table 1:first order filter output 3

Table 2:second order filter output values 5

# Data set description

The dataset refers to YallaMotors website [1], it includes around 6,750 rows and 9 columns, contains 9 features (car name, price, engine\_capacity, cylinder, horse\_power, top\_speed,

seats, brand, country). 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, the target variable is car price.

# Data preprocessing

first step we have standardized all currencies to USD for the target variable “Price”, then we cleaned the dataset by handling the missing values, encoding categorial features and standardizing numerical features.

To convert the price to USD we have used Mocki.io [2] to generate Json file contains the exchange price from each currency to USD [3].

For data cleaning we have cleaned each column separated each null values or wrong values that does not makes sense in the column it was replaced by Nan from the numpy library for the following features: seats, top\_speed, horse\_power, cylinder, engine\_capacity.

Then for the price column each row was checked if it has one of the currencies from the Json file [4] and for the wrong and null data, it was replaced with Nan from numpy library then the null values were taken as the test dataset.

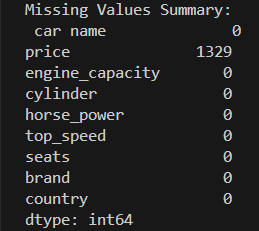
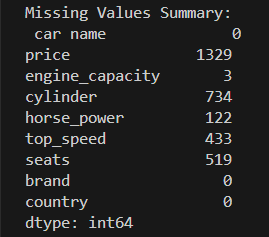
 

Figure :missing values summary before cleaning

Figure :missing values summary after cleaning

Then the data was split for three data sets, training, validation, test.

# Building Regression Models:

## Linear models:

### Linear Regression:

The linear regression was done using the built I function for the linear regression, then to test our model performance it was passed through three performance metrics and the results shown in the figure below:

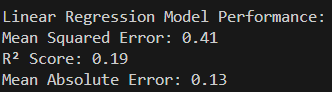
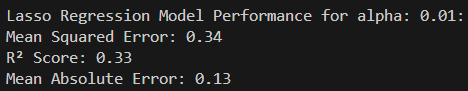
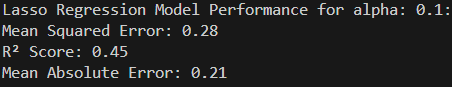


Figure :linear regression performance

### Lasso regression:

The Lasso regression was done using the built it function for Lasso for different alphas,”0.01,0.1,1,10,100”, and passed through three performance metrics and the results shown in the figures below:

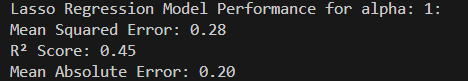


Figure :Lasso for alpha 0.1

Figure :Lasso for alpha 1

Figure :Lasso for alpha 0.01

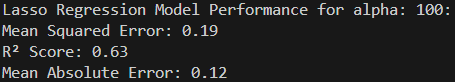
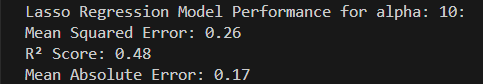


Figure :Lasso for alpha 100

Figure Lasso for alpha 10

### Ridge Regression:

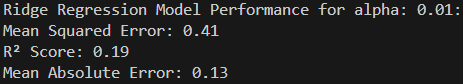
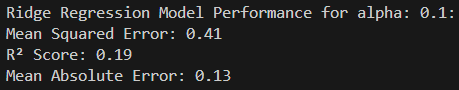
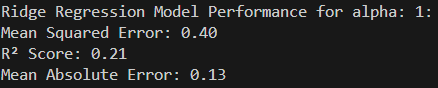
The ridge regression was done using the built it function for Lasso for different alphas,”0.01,0.1,1,10,100”, and passed through three performance metrics and the results shown in the figures below:

Figure :Ridge for alpha 1

Figure :Ridge for alpha 0.1

Figure :Ridge for alpha 0.01

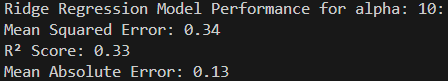
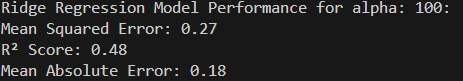


Figure :Ridge for alpha 100

Figure :Ridge for alpha 10

Then by using the belt in function for grid search to find the optimal alpha for the lasso and ridge regression models for lasso it was 0.01, for ridge it was 10 as shown the output in the figure below:



Figure :optimal alpha for lasso and ridge

## Closed Form Solution:

The closed form aims to find the weight vector W without using gradient descent, it was done using the closed form equation:

y

By using “np.linalg.pinv” from NumPy library to make sure of the numerical stability.

Then the error was evaluated using three performance metrics, mean square error, R2, mean Absolute Error the results in the figure below:

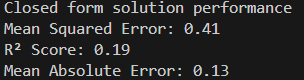


Figure :closed form solution performance

# References

[1]: <https://www.kaggle.com/datasets/ahmedwaelnasef/cars-dataset/data>

[2]: <https://mocki.io/>

[3]: <https://mocki.io/v1/6b55e2fc-4bfd-4e13-9589-30636717e6ce>

[4]: <https://mocki.io/v1/24326926-978b-4c04-a7f8-d79022e96d6f>