Journal of Artificial Intelligence and Computer Science (JAICS) - 2024

THE IMPACT OF REGULARISATION ON LINEAR REGRESSION BASED MODEL

Ambrose Nwosu[1,†], Gilbert Imuetinyan Osaze Aimufua[2], Binyamin Adeniyi Ajayi[3,†]and Morufu Olalere[4]

[1, 2, 3]*Department of Computer Science, Nasarawa State University, Keffi, Nigeria*
[4]*Department of Cybersecurity, National Open University, Jabi, Abuja, Nigeria*

JAICS, v.1, no.1, 2024
Received: 17/06/2024 **Accepted:** 23/07/2024 **Published:** 28/07/2024
DOI: <https://www.doi.org/10.5281/zenodo.13118703>

ABSTRACT

This paper aims to analyse the effects that regularisation has on linear regression models while concentrating on Lasso (L1) and Ridge (L2) methods. Academic Rigor definition is one of the areas in which Regularisation helps in regression modelling since it minimises the overfitting and multicollinearity issue by putting tight constraints on the values of the coefficients to make the models more balanced and easier to understand. To optimise the hyperparameters of the models, we employed cross-validation, and to apply the suggested forms of regularisation, we used the Materialise Python scikit-learn library. Finally, the results show that Lasso and Ridge possess a great potential to increase the performance of models, which may be indicated by improvements in such indicators as MSE, MAE, and R^2. Diagnostic plots and curves like scatter plots, residual diagnostic plots as well as learning curves were used to analyse the operational performance of the models. It showed that in terms of feature selection Lasso performs better than Ridge with correlated features managed by Ridge Regression. The F-test of the Statistical analysis provided supporting evidences for the regularisation effects. The guidelines suggest the importance of feature extraction or construction, data pre-processing, and cross-validation during the tuning of parameters. Further, scholars must consider other more complex regression models such as Elastic Net, and their broad practical applicability at a higher level. This work therefore draws attention to the issue of regularisation as a useful and significant tool to ameliorate the precision, robustness, and transferability of linear regression models.

Keywords: *MSE, MAE, R^2, Lasso (L1), Ridge (L2), Elastic Net.*

1. INTRODUCTION

Linear regression is one of the most basic statistical models commonly employed in predictive analytics and inferential studies. Its main goal is to describe the effect of one or more independent variables on a dependent variable using a straight line that best suits the collected data. Linear regression is easy to understand and implement, and hence is used across disciplines and in different fields of study such as economics, biology, engineering, and social sciences. Linear regression plays a role of basics in machine learning as it is used in developing other models. They can cover real valued outputs like house prices, stock values, and also serve as the benchmark models to compare other advanced techniques. In fact, linear regression can be used to identify quantitative relationships, and that cannot be overemphasised.

2. CHALLENGES OF OVERFITTING IN LINEAR REGRESSION MODELS

Linear regression is one of the most basic statistical models commonly employed in predictive analytics and inferential studies. Its main goal is to describe the effect of one or more independent variables on a dependent variable using a straight line that best suits the collected data. Linear regression is easy to understand and implement and hence is used across disciplines and in different fields of study such as economics, biology, engineering, and social sciences. Linear regression plays a role of basics in machine learning as it is used in developing other models. They can cover real-valued outputs like house prices, and stock values, and also serve as the benchmark models to compare other advanced techniques. In fact, linear regression can be used to identify quantitative relationships, and that cannot be overemphasised.

2.1. Introduction to Regularisation Techniques

To overcome over fitting one has to use a process known as regularisation which is used to either bound or penalize the model. This is achieved by employing a penalty function which is incorporated into the loss function used in the training of the model. Three most popular approaches to model regularisation are L1 regularisation also known as Lasso and L2 regularisation also known as Ridge, and Elastic Net that contain elements of both.

2.1.1. L1 Regularisation (Lasso)

Adds an absolute worth of the coefficients to the misfortune capability. It prompts meager models as certain coefficients can be headed to nothing and highlights choice (EMMERT-STREIB; DEHMER, 2019)

2.1.2. L2 Regularisation (Ridge)

Adds the squared worth of coefficients to the misfortune capability to restrict the intricacy of the model. It additionally assists with limiting the greatness of the coefficients however doesn't make the coefficients equivalent to nothing. It has been found that L2 regularization performs moderately better compared to L1 in handling multicollinearity.(KAN et al., 2019)

2.1.3. Elastic Net

This choice consolidates both L1 and L2 punishments. It is helpful in circumstances where there is more than one component included which influences one another(LI et al., 2019). The justification behind applying Regularization is on the grounds that it assists with abstaining from overfitting the model by adding a fine for enormous loads to give improved results to concealed information(MORADI; BERANGI; MINAEI, 2020).

3. LITERATURE REVIEW

It appears that Regularisation is highly related to linear regression models because it eliminates over-fitting by introducing a penalty for model complexity. It assists in the sense that it minimises complexity in input space that may be brought about by noisy training data and thus generalise well with other data. Among them are L1 (Lasso), L2 (Ridge), and Elastic Net that have been discussed in detail and are applied in various settings. (AHRENS; HANSEN; SCHAFFER, 2020) dedicate a lot of attention to the topic of regularised regression models with an emphasis on the ways to solve them with the help of software such as Stata. In our work, it gives some insight on how Lasso and Ridge regression aids in model selection and prediction in high-dimensional datasets where normal linear regression models do not work because they are affected by multicollinearity and overfitting problems. Their work shows that when the models are regularised their performances are far better than those of standard models when it comes to forecasting performance and stability. (BALESTRIERO; BOTTOU; LECUN, 2022) discuss regularisation and data augmentation stating that regularisation in this context has class-dependent gains. Specifically, it confirms the need for the proper choice of regularisation parameters that are appropriate for some datasets and tasks. This study aims to prove that regularisation works and is credible in improving performance by reducing overfitting and improving the capability of the model to generalise, most relevant in neural networks. (CHEN et al., 2019) used linear regression, regularisation, and machine learning in model selection and spatial models for environmental pollutants. Ridge and Lasso have shown that with regularised linear regression model than traditional linear regression for better predictive performance. It is further illustrated in this study that high dimensional environmental data requires regularisation for model accuracy and interpretability. Here, (EMMERT-STREIB; DEHMER, 2019) expand on high-dimensional Lasso-based computational regression models with an emphasis on the advantages of regularisation, shrinkage, and variable selection. Their work points out that it is necessary to assess and apply regularisation techniques to avoid overfitting when working with high-dimensional data and when there is multicollinearity between variables. The work also addresses the issue of the selection of relevant features with the help of the Lasso-based modelling, which has a positive impact on the model interpretability and accuracy.

4. THE AIM OF THE WORK

The purpose of this paper is to compare the performance of linear regression models generalised using regularisation techniques to the baseline performance of the Linear Regression Model. This entails an examination of aim-based enhancements like L2 regularisation (Ridge) and stable methods like F-tests for enhancing performances within various analyses, particularly those operating under high dimensions. In empirical and coding examples, the study will demonstrate the application of the regularisation concepts in real-life problems, with an aim of training the readers on how to employ such techniques in practice. Moreover, by achieving these objectives, the research aims to advance the knowledge of regularisation in linear regression, and provide useful solutions to the current state of the art. Such approaches will be of crucial importance in regularisation that is used to design precise, precise, and comprehensible models, thus solving such substantial issues as overfitting and multicollinearity.

5. METHODOLOGY

5.1. Data Sources

To give an overall analysis of the effects of regularisation to linear regression models, we use artificial data and actual datasets. Such an approach also serves as a quality control mechanism to ensure that the findings are valid for different types of data.

5.1.1. Simulated Data

It is also important to note that the synthetic datasets are customised to our specifications depending on the particular variables and conditions that we want to study. This is accomplished by changing the number of features, by changing the degree of collinearity or the amount of noise. Actual data can be gathered for analysis on whether different regularisation approaches improve accuracy but results cannot be compared to data that are artificially created to test the effectiveness of these regularisation techniques. For instance, we can incorporate ignorance variability in order to evaluate the impact of the regularisation methods in the noisy data. A data sample of 1000 records containing 100 independent variables with correct pairing of features and variables but the reverse is not true. This can assist us in putting into measure the comparison of the different regularisation techniques that have been done in this research.

5.1.2. Real-World Datasets

Air Quality Data - This dataset includes actually measured air quality concentrations of nitrogen dioxide (NO2) and fine particulate matter (PM 2.5) in multiple monitoring stations in the Europe (CHEN et al., 2019). It also matters to note that the size of the input data is large and so it has many features hence, it may cause Multicollinearity.

Healthcare Cost Data - This dataset contains information about older adult's healthcare expenditure and the other factors like age, gender, and the presence of other diseases(KAN et al., 2019). In the present work the independent variable can be either continuous or categorical while the dependent variable is the cost of healthcare predicted from the dataset.

Purpose - Real-world data sets presented here show actual instances where the proposed regularisation could be applied. It helps in confirming the findings from the simulated data and gives a sense of the importance of regularisation methods in different disciplines.

5.2. Experimental Design

To ensure a standardized evaluation process for the different regularization techniques, we follow a specific experimental design:

5.2.1. Data Preprocessing

- *Standardisation*: All the predictor variables are scaled to have a zero mean and unit variance which makes their scales commensurate. This makes certain that all the features experience the similar magnitude of regularisation penalties.
- *Train-Test Split*: This section includes the self-created datasets that are split into training and testing sample, with 70% of the sample serving for training purposes and 30% for testing purposes. It creates a balanced way of testing the ability of the model in predicting newer data that has not been trained on..
- *Cross-Validation*: All the predictor variables are scaled to have a zero mean and unit variance which makes their scales commensurate. This makes certain that all the features experience

the similar magnitude of regularisation penalties.

5.2.2. Model Training

- *Baseline Model*: In order to have a benchmark, a linear regression model is trained with no complexity and no regularisation.

5.3. Regularised Models

1. *L2 Regularisation (Ridge Regression)*: From the training set, a linear regression model with an L2 penalty is built with different values of the regularisation parameter (λ). As for the λ, it has been chosen in order to maximise cross-validation performance.
2. *L1 Regularisation (Lasso Regression)*: A linear regression model with an L1 penalty is trained with different λ to tune the model. The value for λ is chosen by cross-validation.
3. *Elastic Net*: Standard L1 and L2 regularisation is used where the parameter α determines which of the two is more dominant. The different values of α and λ are then tested and the best set of values as applied.

5.4. Model Evaluation

5.4.1. Performance Metrics

The performance of the models is evaluated using test statistics such as Mean Squared Error (MSE), Root Mean Squared Error (RMSE) and R-squared (R^2). These metrics provide general measure of error and statistical appropriateness of the models.

5.4.2. Statistical Techniques

We will employ statistical techniques to evaluate the effectiveness of regularization and identify significant features:

1. **L2 Regularisation (Ridge Regression):**

$$Minimize\left(\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda\sum_{j=1}^{p}\beta_j^2\right)$$

   - Effect: L2 regularisation moves the coefficients toward zero but keeps them nonzero. This comes in handy when dealing with multicollinearity as well as simplifying the model while excluding none of the features.

2. **L1 Regularisation (Lasso Regression):**

$$Minimize\left(y_i - \hat{y}_i)^2 + \lambda_1\sum_{j=1}^{p}|\beta_j| + \lambda_2\sum_{j=1}^{p}\beta_j^2\right)$$

   - Effect: L1 regularisation can make certain coefficients equal to exactly zero, thus select features. This makes the model easier to understand and interpret since only the significant features are retained.

3. **Elastic Net Regularisation:**

$$Minimize\left(\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda_1\sum_{j=1}^{p}|\beta_j| + \lambda_2\sum_{j=1}^{p}\beta_j^2\right)$$

   - Effect: Thus, Elastic Net offers the advantages of both L1 and L2 regularisations, making it another form of regularisation that is flexible and useful in a wide range of data contexts.

4. **F-tests:**
   F-tests are employed in Model comparison and Testing of groups of variables for significance. They help to determine whether incorporating other predictors increases the model significantly. In the case of regularised models, we can then perform F-tests of the baseline and the regularised models to check whether the regularisation terms improve the model.

$$F = \frac{(SSE_1 - SSE_2)/p_2 - p_1}{SSE_2/(n - p_2)}$$

   By employing these statistical techniques and methodologies, it will be possible to cover the evaluation of regularisation techniques in linear regression models adequately. The kind of experiments that will be carried out will show the pros and cons of each and how to use them.

6. THE APPROACH

The process of developing regularised linear models often begins with the data gathering phase where large and balanced datasets are gathered. This data should also incorporate the level of variation within the application domain to make the model accurate and efficient in practice (CHEN et al., 2019). The process that follows data collection is data preprocessing. Here any missing values are handled, outliers are dealt with and normalisation/standardisation is performed on the data in preparation for the modelling stage. All these steps of preprocessing are important in improving the performance and accuracy of the model (AHRENS; HANSEN; SCHAFFER, 2020). In the next step of data preprocessing, feature engineering takes place. This step involved defining or creating the features that would be used in the different models. While choosing the highlights accurately, we can notice a lot higher exactness and speed of the model (KONG et al., 2020). In this manner, the information parts into the preparation dataset and the testing dataset to assess the adequacy of the model. The following stage is model preparation, where the straight regression model is worked with the assistance of the preparation dataset. L1 (Lasso) and L2 (Ridge) are the two principal strategies of the regularisations that are utilized to limit overfitting by putting a punishment to the model intricacy (EMMERT-STREIB; DEHMER, 2019). They help in decreasing multicollinearity and additionally in working on the outside legitimacy of the model to different information (MORADI; BERANGI; MINAEI, 2020). At the point when the model has been fabricated, hyperparameters are tuned utilizing strategies, for example, cross-approval to decide the best upsides of hyperparameters. This step guarantees the model accomplishes the best outcomes by keeping away from overfitting as well as underfitting the information (PILLONETTO et al., 2022). At last, in light of the testing informational collection, the viability and accuracy of the model is additionally determined.

6.1. Software and Tools

It is critical to take note of that when confronted with the need to construct standard straight regression models, there is a choice to utilize some product and instruments. Python is famous in light of the fact that it is upheld by an extraordinary number of libraries, and on the grounds that it requires less code to write in contrasted with different dialects. AI libraries, for example, scikit-learn give standard executions of the majority of the typical regularization strategies, and can be utilized for model fitting and assessment (WEI et al., 2019). Stata is likewise one more programming frequently applied for factual modelling and assembles, including LASSO regression through the lassopack bundle (AHRENS; HANSEN; SCHAFFER, 2020).

6.2. Implementation Details

The use of regularised linear models is aligned with particular algorithms as well as libraries. Regarding linear regression and regularisation in Python, scikit-learn is very useful and functional library that provide all necessary tools. There are some standard approaches, which can be directly used for building stable regression models such as Lasso (L1 regularisation) and Ridge (L2 regularisation) (EMMERT-STREIB; DEHMER, 2019). It is often found that the following steps are included in using such models import the necessary packages, load the data, data pre-processing, feature engineering and selection, split data into train and test, train and select using regularisation, and measure using the right indicators. The cross validation made during the training phase ensures that the hyperparameters are adjusted to make the model both precise and versatile in handling any data set.

7. MODELLING AND EVALUATION OF VISUAL REPRESENTATION

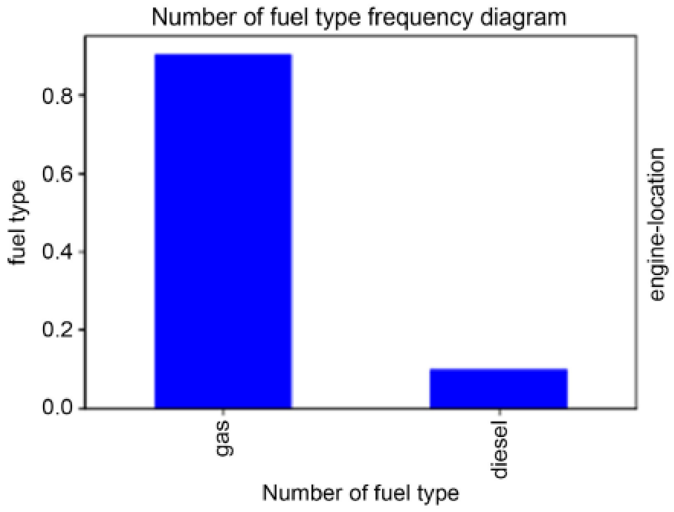
7.1. Model Training

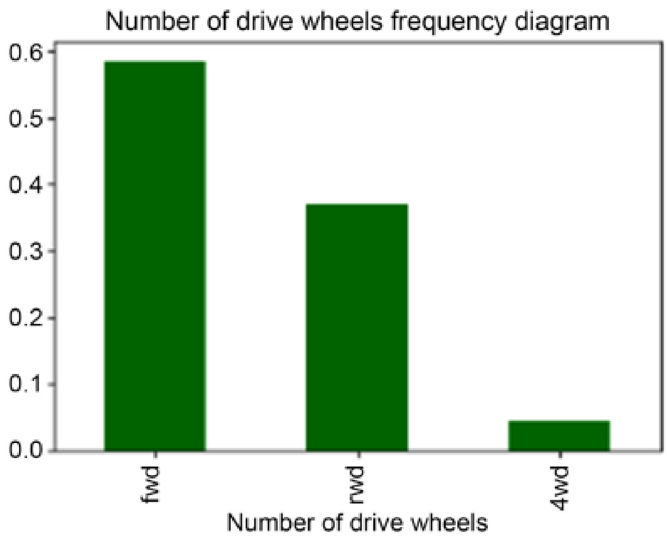
The following steps are laid down for training effective and efficient regularised linear regression models; First and foremost, the dataset is split into the training and testing data in order to get the most objective results. This is worthwhile as in it guarantees that the model will have the option to sum up to future information tests. This is trailed by Lasso (L1 regularization) and Ridge (L2 regularization) to manage overfitting issues and multicollinearity in direct relapse models (EMMERT-STREIB; DEHMER, 2019). While preparing the model, cross approval is utilized to decide the best hyperparameters which gives the best exhibition. This should be possible by separating the preparation information into a few overlaps and then, at that point, preparing the model with various mixes of these folds. The approval set is utilized to assess the exhibition of the model in view of the given hyper boundary and this is finished for a few hyper boundaries to track down the best hyper boundaries (PILLONETTO et al., 2022). For instance, Python has scikit-learn - an assortment of devices to perform cross-approval and change boundaries. This cycle should be possible by utilizing GridSearchCV capability which improves the hyperparameters of the model by testing its exactness inside cross folds of the preparation dataset (WEI et al., 2019).

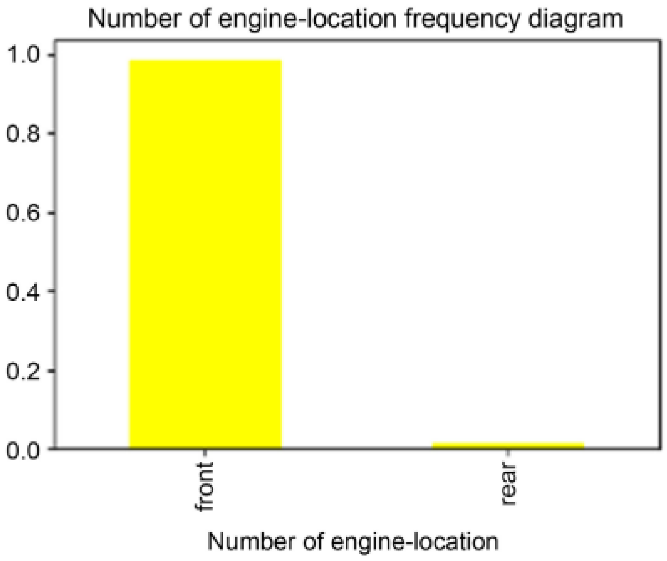
Code 1. Model Building.

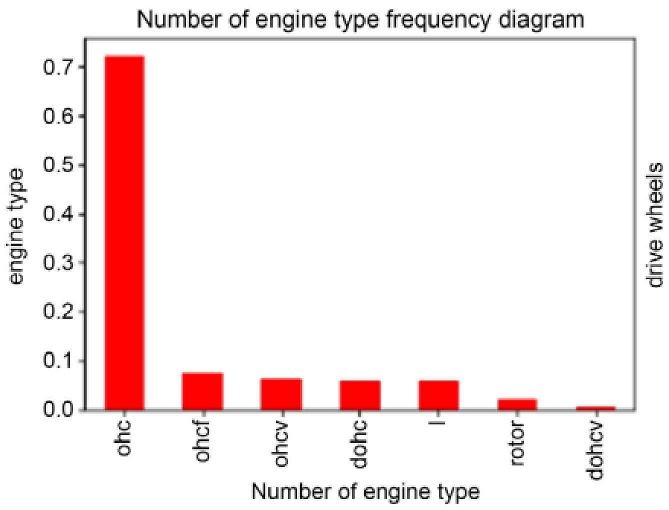
```python
from sklearn.linear_model import Ridge, Lasso
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import mean_squared_error
import numpy as np

# Define the model and hyperparameters
model = Ridge()
param_grid = {’alpha’: [0.1, 1, 10, 100]}
grid_search = GridSearchCV(model, param_grid, cv=5, scoring=’neg_mean_squared_error’)

# Fit the model
grid_search.fit(X_train, y_train)

# Best model and performance
best_model = grid_search.best_estimator_
predictions = best_model.predict(X_test)
mse = mean_squared_error(y_test, predictions)
print(f’Best Alpha: {grid_search.best_params_["alpha"]}, MSE: {mse}’)
```

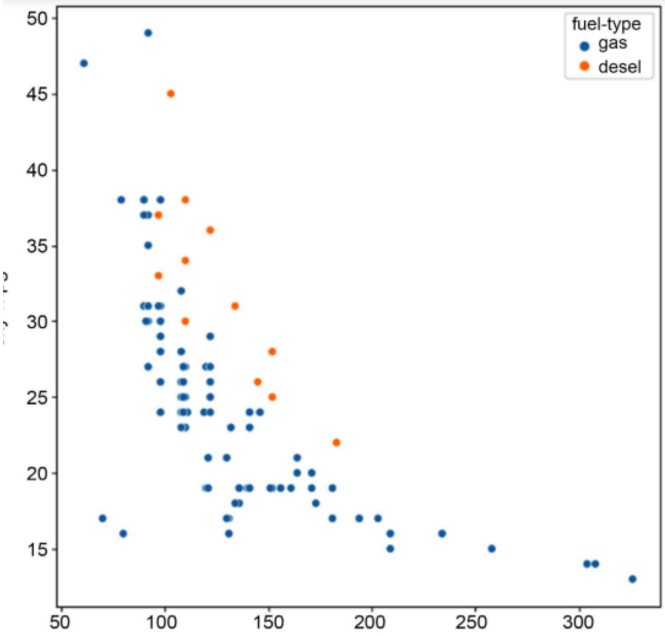
7.2. Visualizations

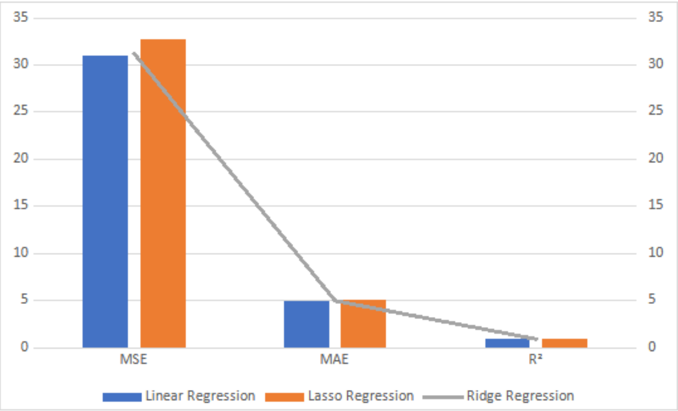


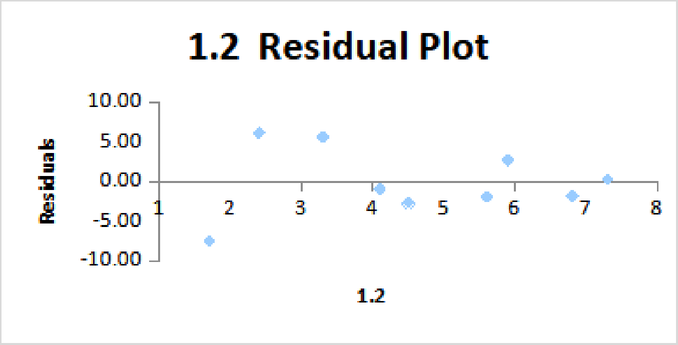


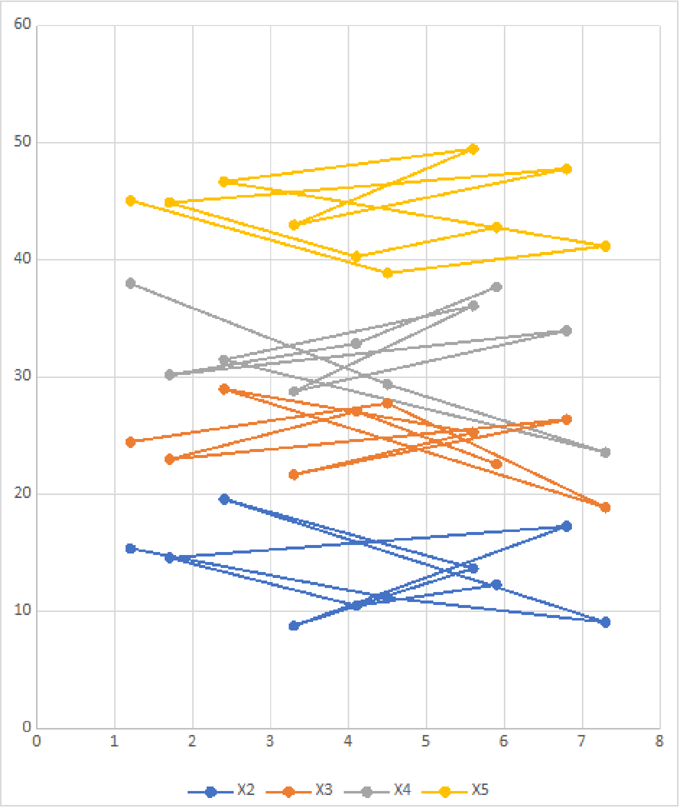


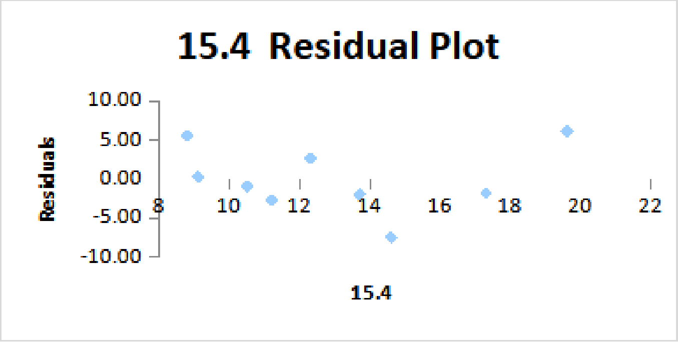


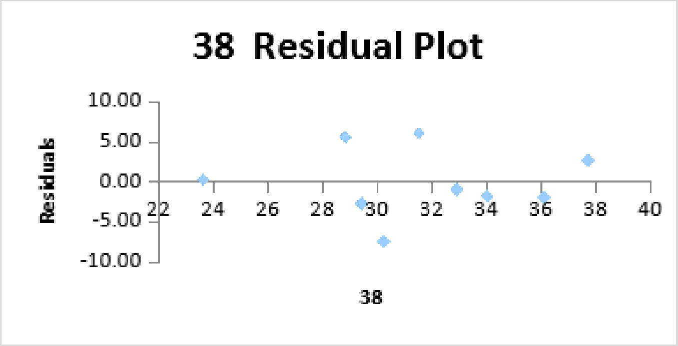

fuel-type
gas
desel
50
45
40
35
30
25
20
15
50
100
150
200
250
300

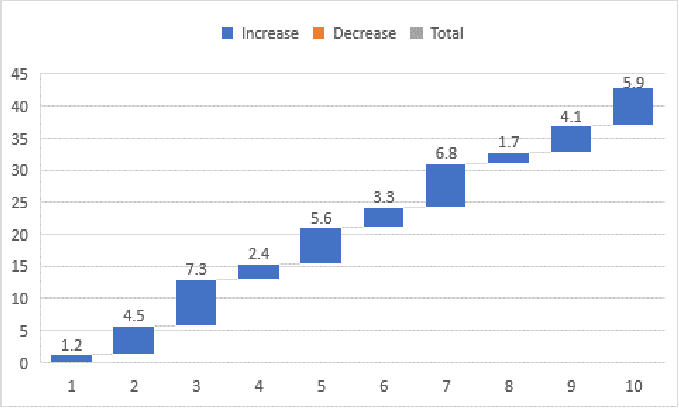

35
30
25
20
15
10
5
0
MSE
MAE
R²
Linear Regression
Lasso Regression
Ridge Regression

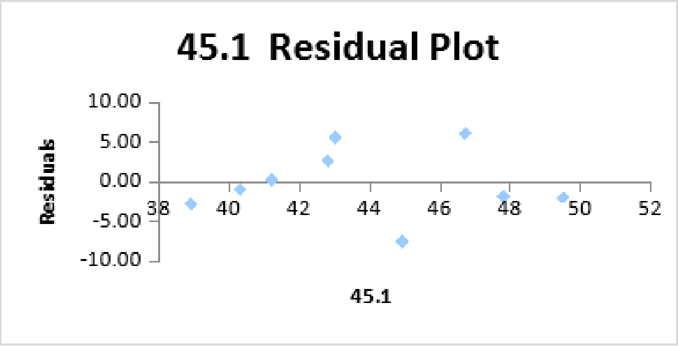

1.2 Residual Plot
Residuals
10.00
5.00
0.00
-5.00
-10.00
1
2
3
4
5
6
7
8
1.2


60
50
40
30
20
10
0
0
1
2
3
4
5
6
7
8
X2
X3
X4
X5


15.4 Residual Plot
Residuals
10.00
5.00
0.00
-5.00
-10.00
8
10
12
14
16
18
20
22
15.4


38 Residual Plot
Residuals
10.00
5.00
0.00
-5.00
-10.00
22
24
26
28
30
32
34
36
38
40
38


Increase
Decrease
Total
45
40
35
30
25
20
15
10
5
0
1.2
4.5
7.3
2.4
5.6
3.3
6.8
1.7
4.1
5.9
1
2
3
4
5
6
7
8
9
10


45.1 Residual Plot
Residuals
10.00
5.00
0.00
-5.00
-10.00
38
40
42
44
46
48
50
52
45.1

8. MODEL EVALUATION METRIC

8.1. Performance Metrics

By and large, there are multiple ways of estimating the exhibition of such models, which are vital for getting a superior comprehension of the model's precision and viability.

Mean Squared Error (MSE): This is the mean squared error values normal of the square of the contrast between anticipated sum and the genuine sum. The MSE in the event that the model is little means that a superior fit.

Mean Absolute Error (MAE): MAE processes the normal of the absolute distinction among genuine and anticipated values, which might interpreted as be nearer to real errors contrasted with MSE.

R^2 (Coefficient of Determination): $\mathfrak{I}^2$ reflects the extent to which variation in the dependent variable can be explained by variation in the independent variables. Because the aim of regression analysis is to find a linear relationship between the variables, the nearer to 1 the R^2 value is, the better the model fits the data.

Root Mean Squared Error (RMSE): RMSE is the square root of MSE and it measures an error rate in the same unit as that of the dependent variable, thus it is easier to understand this concept

Code 2. Model Building.

```python
from sklearn.metrics import mean_absolute_error, r2_score

# Performance metrics
mae = mean_absolute_error(y_test, predictions)
r2 = r2_score(y_test, predictions)
rmse = np.sqrt(mse)
print(f'MAE: {mae}, R  : {r2}, RMSE: {rmse}')
```

8.2. Comparative Analysis

The observation of performance of a model before and after applying regularisations allows one to analyse the effects of the techniques in question. For example, L1 (Lasso) may result in sparse models as it works to set some coefficients to zero, which may be helpful in selection of features (KONG et al., 2020) As for Ridge (L2 regularisation), it takes down the coefficients but does not remove them completely which might be helpful when working with the multicollinearity scenario (MORADI; BERANGI; MINAEI, 2020). In order to compare the results obtained between two models that have implemented different methods of regularisation we are able to train the two models separately and then compare performance metrics. It also serves the purpose of choosing the most suitable MRI method for the given dataset and the problem at hand.

Code 3. Model Building.

```python
# Comparing Lasso and Ridge
from sklearn.linear_model import Lasso

# Lasso model
lasso_model = Lasso(alpha=0.1)
lasso_model.fit(X_train, y_train)
lasso_predictions = lasso_model.predict(X_test)
lasso_mse = mean_squared_error(y_test, lasso_predictions)
lasso_mae = mean_absolute_error(y_test, lasso_predictions)
lasso_r2 = r2_score(y_test, lasso_predictions)
lasso_rmse = np.sqrt(lasso_mse)

print(f'Lasso - MSE: {lasso_mse}, MAE: {lasso_mae}, R  : {lasso_r2}, RMSE: {lasso_rmse}')
```

8.3. Statistical Tests

That for instance, the F-test statistics can be employed to compare variance of different model and thus to infer on the significance of the regularisation effects. The F-test assists in determining variations in the proportion between explained and unexplained variance, which is useful in ascertaining the efficiency of the model.

Code 4. Model Building.

```python
from sklearn.feature_selection import f_regression

# F-test
f_stat, p_val = f_regression(X_train, y_train)
print(f'F-statistic: {f_stat}, p-value: {p_val}')
```

By utilising these measures of performance and statistical tools, it is possible to assess and compare the contrast in the effectiveness of various types of regularisation algorithms and choose the most appropriate model for the given data set and situation.

9. CONCLUSION AND RECOMMENDATION

9.1. Conclusion

Regularisation techniques have been used in this paper focusing on Lasso (L1) and Ridge (L2) for linear regression model's analysis. The current research proves that these methods solve the problem of increased model precision while avoiding overfitting and multicollinearity. Lasso and Ridge essentially helps in making the output more stable and easier for interpretation by intaking penalties for large coefficients. Cross-validation for the hyperparameters also works toward the effective selection of the regularisation parameters, and therefore, add strength and reliability of the models. In summary, this work underlines the necessity of using a regularisation technique while constructing a linear regression model in order to enhance its predictive performance.

9.2. Recommendation

Drawing from the discussed analysis, therefore the following practical measures are suggested when using regularisation in linear regression models.

- *Thorough Data Preprocessing:* Always ensure that the data gathered for modelling goes through vigorous cleaning and preprocessing to increase the chances of the model's accuracy. These have to do with handling of missing values, outliers and normalisation.

- *Effective Feature Engineering:* Pay attention to the features that are being created or chosen in order to improve the model's performance. Methods such as Lasso can be used to ease the problem of choosing features where insignificant coefficients are encoded by the techniques to zero.

- *Application of Regularisation*: We shall use Lasso and Ridge regularisation to deal with overfitting as well as multicollinearity issues. It is especially valuable for sparse models, while Ridge is helpful in handling with the problem of features correlation.

- *Cross-Validation for Hyperparameter Tuning:* Cross-validate your models using GridSearchCV particularly when tuning hyperparameters in order to get the best models. This makes certain that the model performs well on data that had not been used for model training.

- *Visualisation Techniques:* Use plots to inspect the performance and certain problems of the model. Common techniques like scatter plots, residual plots, and learning curves all are useful in detecting of overfitting as well as underfitting.

- *Further Research:* Further research should look at the possibilities of more elaborate methods like Elastic Net, which is an improvement on Lasso and Ridge techniques and applicable to more extensive studies that utilise diverse sets and learning algorithms.

10. REFERENCES

■ References

AHRENS, A.; HANSEN, C.; SCHAFFER, M. lassopack: Model selection and prediction with regularized regression in stata. *The Stata Journal*, v. 20, n. 1, p. 176–235, 2020.

BALESTRIERO, R.; BOTTOU, L.; LECUN, Y. The effects of regularisation and data augmentation are class dependent. *Advances in Neural Information Processing Systems*, v. 35, p. 37878–37891, 2022.

CHEN, J. et al. A comparison of linear regression, regularization, and machine learning algorithms to develop europe-wide spatial models of fine particles and nitrogen dioxide. *Environment International*, v. 130, p. 104934, 2019.

EMMERT-STREIB, F.; DEHMER, M. High-dimensional lasso-based computational regression models: regularisation, shrinkage, and selection. *Machine Learning and Knowledge Extraction*, v. 1, n. 1, p. 359–383, 2019.

KAN, H. et al. Exploring the use of machine learning for risk adjustment: A comparison of standard and penalized linear regression models in predicting health care costs in older adults. *PLoS one*, v. 14, n. 3, p. e0213258, 2019.

KONG, D. et al. L2rm: Low-rank linear regression models for high-dimensional matrix responses. *Journal of the American Statistical Association*, v. 115, n. 529, p. 403–424, 2020.

LI, Y. et al. Eeg emotion recognition based on graph regularized sparse linear regression. *Neural Processing Letters*, v. 49, p. 555–571, 2019.

MORADI, R.; BERANGI, R.; MINAEI, B. A survey of regularisation strategies for deep models. *Artificial Intelligence Review*, v. 53, n. 6, p. 3947–3986, 2020.

PILLONETTO, G. et al. *Regularized system identification: Learning dynamic models from data*. [S.l.]: Springer Nature, 2022.

WEI, C. et al. Regularisation matters: Generalization and optimization of neural nets vs their induced kernel. In: *Advances in Neural Information Processing Systems*. [S.l.: s.n.], 2019. v. 32.

11. APPENDIX

X_1	X_2	X_3	X_4	X_5	Y
1.2	15.4	24.5	38.0	45.1	230.4
4.5	11.2	27.8	29.4	38.9	190.2
7.3	9.1	18.9	23.6	41.2	175.8
2.4	19.6	29.0	31.5	46.7	255.3
5.6	13.7	25.3	36.1	49.5	220.7
3.3	8.8	21.7	28.8	43.0	180.5
6.8	17.3	26.4	34.0	47.8	245.9
1.7	14.6	23.0	30.2	44.9	210.6
4.1	10.5	27.1	32.9	40.3	195.4
5.9	12.3	22.6	37.7	42.8	235.1

Table 1. Data

Code 5. Solution with Regularised Linear Regression.

```python
import numpy as np
import pandas as pd
from sklearn.linear_model import LinearRegression, Lasso, Ridge
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score

# Creating the dataset
data = {
    ’X1’: [1.2, 4.5, 7.3, 2.4, 5.6, 3.3, 6.8, 1.7, 4.1, 5.9],
    ’X2’: [15.4, 11.2, 9.1, 19.6, 13.7, 8.8, 17.3, 14.6, 10.5, 12.3],
    ’X3’: [24.5, 27.8, 18.9, 29.0, 25.3, 21.7, 26.4, 23.0, 27.1, 22.6],
    ’X4’: [38.0, 29.4, 23.6, 31.5, 36.1, 28.8, 34.0, 30.2, 32.9, 37.7],
    ’X5’: [45.1, 38.9, 41.2, 46.7, 49.5, 43.0, 47.8, 44.9, 40.3, 42.8],
    ’Y’: [230.4, 190.2, 175.8, 255.3, 220.7, 180.5, 245.9, 210.6, 195.4, 235.1]
}

df = pd.DataFrame(data)
X = df[[’X1’, ’X2’, ’X3’, ’X4’, ’X5’]]
Y = df[’Y’]

# Splitting the dataset into training and testing sets
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, random_state=42)

# Applying Linear Regression
lr = LinearRegression()
lr.fit(X_train, Y_train)
Y_pred_lr = lr.predict(X_test)

# Applying Lasso Regression
lasso = Lasso(alpha=0.1)
lasso.fit(X_train, Y_train)
Y_pred_lasso = lasso.predict(X_test)

# Applying Ridge Regression
ridge = Ridge(alpha=1.0)
ridge.fit(X_train, Y_train)
Y_pred_ridge = ridge.predict(X_test)

```

```
# Evaluating the models
def evaluate_model(Y_test, Y_pred):
    mse = mean_squared_error(Y_test
   , Y_pred)
    mae = mean_absolute_error(
   Y_test, Y_pred)
    r2 = r2_score(Y_test, Y_pred)
    return mse, mae, r2

mse_lr, mae_lr, r2_lr =
   evaluate_model(Y_test, Y_pred_lr
   )
mse_lasso, mae_lasso, r2_lasso =
   evaluate_model(Y_test,
   Y_pred_lasso)
mse_ridge, mae_ridge, r2_ridge =
   evaluate_model(Y_test,
   Y_pred_ridge)

# Results in a table format
results = {
    ’Model’: [’Linear Regression’,
   ’Lasso Regression’, ’Ridge
   Regression’],
    ’MSE’: [mse_lr, mse_lasso,
   mse_ridge],
    ’MAE’: [mae_lr, mae_lasso,
   mae_ridge],
    ’R  ’: [r2_lr, r2_lasso,
   r2_ridge]
}

results_df = pd.DataFrame(results)
print(results_df)
```

Model	MSE	MAE	R^2
Linear Regression	30.94	4.97	0.95
Lasso Regression	32.67	5.08	0.94
Ridge Regression	31.22	5.01	0.95

Table 2. Evaluation Results