Final Project Report - Forest Fire Regression Analysis

Rahiya Rasheed

Contents

Introduction

Forest fires are a dangerous threat to our planet's environment, with impacts ranging from wildlife endangerment to economic losses. To minimize the damage caused by wildfires, it is important to be able to foresee and prepare for them. This report aims to predict the area burned by forest fires using a dataset from the UCI machine learning archive. The dataset provides information about forest fires in the Montesinho Natural Park, located in northeast Portugal. The original paper proposed by Paulo Cortez and Anbal Morais[1] solves this problem by using data mining and models such as multiple Regression, Neural Networks, and Support Vector Machines. The dataset has 13 variables, of which 12 are predictors: `X` (spatial x-coordinate ranging from 1 (West) to 9 (East)), `Y` (spatial y-coordinate ranging from 1 (South) to 9 (North)), `month`, `day`, `FFMC` (Fine Fuel Moisture Code - a metric for the amount of fuel moisture of forest under the shade of a forest canopy ranging from 0 to 101), `DMC` (Duff Moisture Code - which represents the fuel moisture of decomposed organic moisture underneath the litter), `DC` (Drought Code - which represents the dryness in the soil with a maximum value of 1000), `ISI` (Initial Spread Index - a numeric rating that estimates the initial spread of the fire under the fire weather index of the Canadian government), `temp` (temperature), `RH` (Relative Humidity, in %), `wind`, and `rain`. `area` is the response variable, and is measured in hectares.

Methodology

We divided this dataset into two: the so-called Normalized and Non-Normalized ones. These differed in the transformation applied to `area`, described below:

$$\begin{cases} \text{Normalized: } f_{\texttt{norm}}(x) = \sqrt{\texttt{orderNorm}(x)} \\ \text{Non-Normalized: } f_{\texttt{non-norm}}(x) = \log(1+x) \end{cases}$$

Our first model will be a traditional Multiple Linear Regression (MLR) model with cleaned data (removal of unusual points). We will examine the model's diagnostics, checking if it holds its assumptions of constant variance (Breusch Pagen Test), normality (Shapiro Wilks Test) and uncorrelated errors (Durbin Watson Test).

We will then perform three shrinkage methods: Principal Components Regression (PCR), Ridge Regression (RR), and Lasso Regression (LR). For these models, we classified outlier points as points more than 3 SD away from their mean (for continuous variables), and removed them prior to training the models.

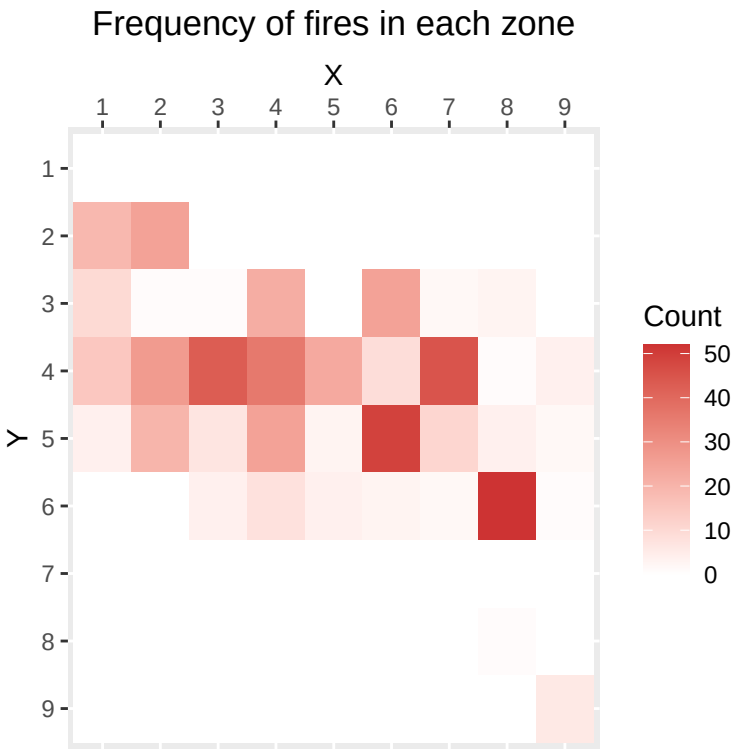
We will assess the models' performance by comparing their Root Mean Square Error (RMSE), for both the training and testing sets. We will also examine the RMSE for back-transformed (i.e. having the inverse of the corresponding transformation applied) predicted response variables. Note that the inverse functions are defined as follows:

$$\begin{cases} \text{Normalized: } f^{-1}_{\texttt{norm}}(x) = \texttt{orderNorm}^{-1}\left(x^2\right) \\ \text{Non-Normalized: } f^{-1}_{\texttt{non-norm}}(x) = e^x - 1 \end{cases}$$

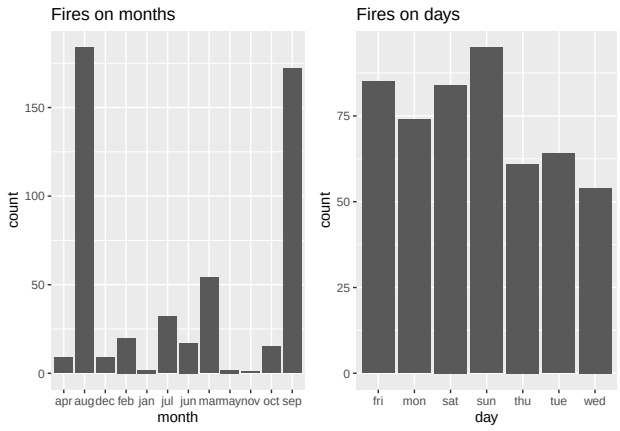
Note that when inverting the transformation applied to residuals for models trained on normalized data, we multiplied the residuals by -1 if they were originally negative, in order to mitigate some of the error arising from squaring a number which could be positive or negative. We recognize that this is not a mathematically sound way of dealing with the square root transform; however, given the scope of the project, we believe that this is a permissible error.

Data

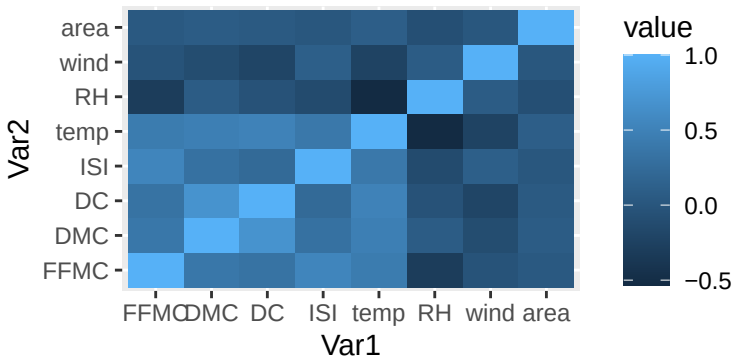
The data we will be using for this analysis is Meteorological with 13 variables, of which 4 are categorial: `X`, `Y`, `month`, and `day`.



`month` and `day` tell us on which day of the week and month of the year the fire occurred.

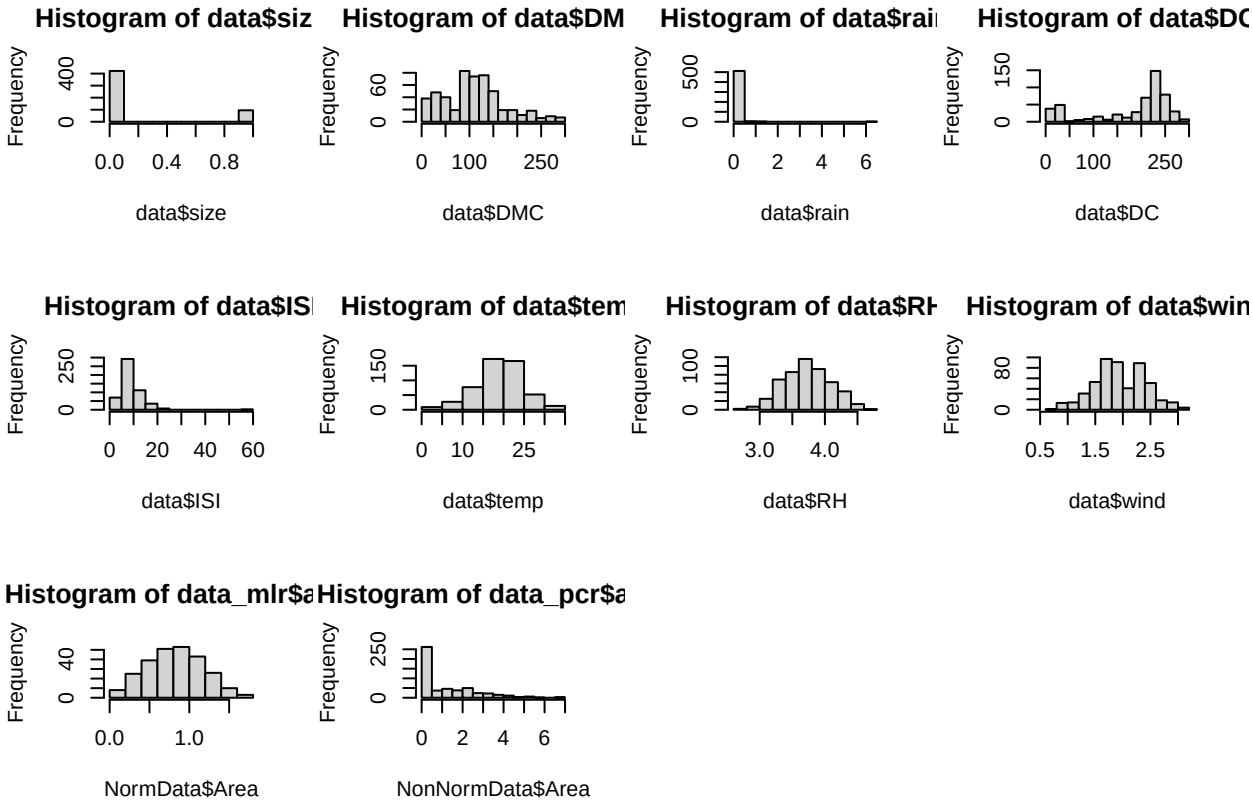


The plots above tell us that most fires occurred in August, and on Sunday. These variables do not tell us a lot about the size or occurence of forest fires, and are therefore not given much importance. Next, we examine the correlation of the dataset in the heatmap below.



From this, it can be seen that `temp` and `RH` have a very high negative correlation, whereas `DC` and `DMC` have a relatively high positive correlation.

This is believed to be a hard regression problem for a few reasons: the raw data is not normal, it does not appear to show any linearity with the regressors, and the independent variables do not follow a normal distribution. We tackle this problem by transforming the data into normal distributions using straightforward transformations on variables `DC`, `RH`, `wind`, and `ISI`, and applying the function `bestNormalize`[2] in R to find the best transformation for variables `area` and `FFMC`. The `bestNormalize` package was designed to find the optimal normalizing transformation for a vector. The function performs many transformations - such as Yeo-Johnson, `arcsinh`(x), and Ordered Quantile normalizing - and simplifies this process for more unevenly-distributed variables. We also add a new binary variable `size`, defined as 0 (`area` < 10) or 1 (otherwise).



The table below illustrates the transformation applied to each of the transformed variables in both the

normalized and non-normalized data sets.

Data Set	Variable	Transformation
Normalized	'area'	$\sqrt{\texttt{orderNorm(area)}}$
Normalized	'FFMC'	$\texttt{orderNorm(FFMC)}$
Non-Normalized	'area'	$\log(1 + \texttt{area})$
Both	'DC'	$\texttt{DC}^{\frac{1}{3}}$
Both	'RH'	$\log(\texttt{RH})$
Both	'wind'	$\sqrt{\texttt{wind}}$

Figure Table 3.1: Transformations

Implementation

Implementation on Non-Normalized Data

Multiple Linear Regression model

We started out by creating an $80 - 20$ training-testing split of the data and performing MLR with the following results:

Summary	Result
R^2	0.6887
Residual Standard Error	0.8123 on 476 df
F-Statistic	26.33 on 40 and 476 df
p-value	$<$ 2.2e−16

Figure Table 4.1.1: MLR Results

Looking into the model diagnostics, we removed any unusual points (defined as high leverage points, outliers, or influential points). Subsequently, we re-ran the MLR model, obtaining the following results:

Summary	Result
R^2	0.684
Residual Standard Error	0.7997 on 370 df
F-Statistic	22.28 on 36 and 370 df
p-value	$<$ 2.2e−16
RMSE on Training Set	0.7625175
RMSE on Test Set	0.8369324
RMSE on Training Set (Untransformed)	0.1441019
RMSE on Test Set (Untransformed)	71.97844

Figure Table 2.2: Cleaned Data MLR results

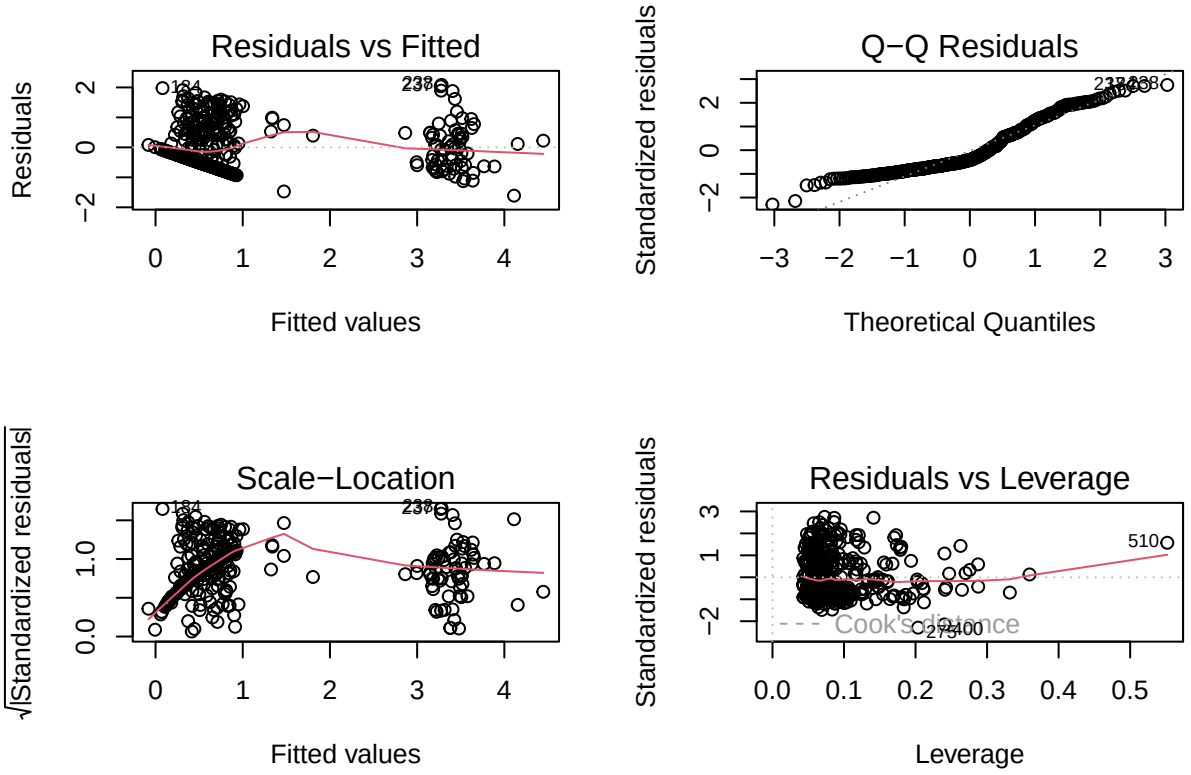
We can see that the R^2 slightly improved for this model, but not by a significant margin.

Next, we checking the model assumptions to see if the linear model passes the Breusch Pagen test (Homoscedasticity), Shapiro Wilks test (Normality), and Durbin Watson test (Correlation). The table and graphs show the results.

Test	Result
Breusch Pagen Test	0.08533
Shapiro Wilks Test	1.982×10^{-13}
Durbin Watson Test	5.487×10^{-13}

Figure Table 2.3: Cleaned MLR Model Diagnostics

As expected, it fails the Sharpiro Wilks test of normality as the data is not normalized and the errors are correlated. As for the Breusch Pagen test, the plot does not support this assumption as we can clearly see a linear trend in the `area` $= 0$ region below.



The residual vs fitted values plot show a pattern and therefore have non constant variance. The QQ plot is deviating from the line in the middle of the plot and therefore the model is not normal.

Shrinkage

We have observed that our data deviates from normality, and there are numerous predictors that lack significance. To address this issue effectively, we will employ dimensional reduction methods. These techniques involve transforming the data from its high-dimensional space to a new space with lower dimensionality, thereby reducing the number of predictors.

Dealing with high-dimensional data poses several challenges, and employing dimensionality reduction can yield several benefits. First and foremost, it enables more efficient computations. Additionally, by reducing

the complexity of the data, learning algorithms can achieve better generalization abilities. Moreover, dimensionality reduction helps in identifying meaningful structures within the data, making it more interpretable.

We will implement three shrinkage models:

1. Principal Components Regression
2. Ridge Regression
3. finally Lasso Regression.

Principal Components Regression (PCA) PCA is a dimensionality reduction technique used to convert high-dimensional data into a lower-dimensional space while preserving its main patterns. It identifies the principal components (PCs), which are orthogonal vectors capturing the most significant variations in the data. By selecting the top few PCs, PCA simplifies the data while retaining crucial information.

PCA does this by solving for the eigenvectors and eigenvalues of the covariance matrix of the standardized data. Given a dataset $\mathbf{X}$ with n samples and m features, PCA aims to find the matrix $\mathbf{V}$, containing the eigenvectors, and the vector $\vec{\lambda}$, containing the corresponding eigenvalues of the covariance matrix.

The covariance matrix $\mathbf{C}$ of the standardized data $\mathbf{X}$ is defined as:

$$\mathbf{C} = \frac{\mathbf{X}^T\mathbf{X}}{n-1}$$

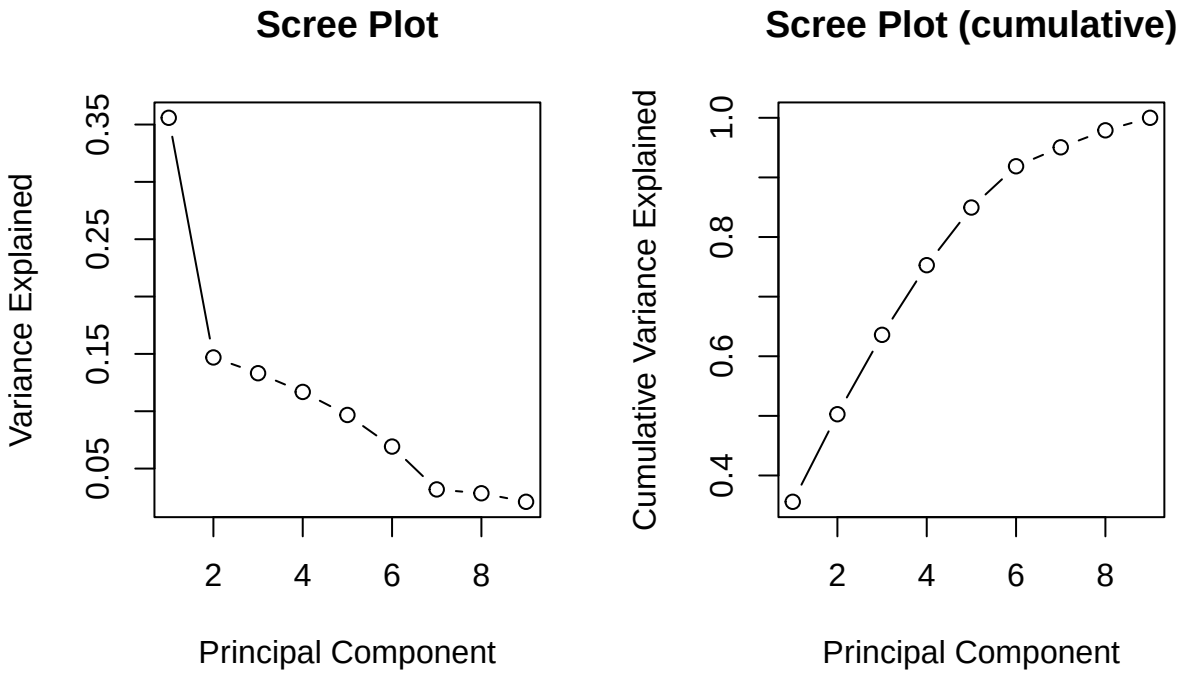
where $\mathbf{X}^T$ is the transpose of the data matrix X. The equation PCA solves is:

$$\mathbf{CV} = \mathbf{X}\vec{\lambda}$$

where $\mathbf{V}$ is the matrix of eigenvectors and $\vec{\lambda}$ is the vector of eigenvalues. The eigenvectors represent the PCs, and the eigenvalues indicate the amount of variance captured by each corresponding PC. PCA aims to compute these eigenvectors and eigenvalues to achieve dimensionality reduction.

Our first step was to recognize outliers in the continuous data and remove them, since PCA is very sensitive to outliers.

We than plotted the scree plot to determine how many PCs were necessary.



From the Scree plot above, there was no obvious elbow; therefore we selected the number of PCs by the amount of variance they explained cumulatively, setting the threshold to 90% of variance explained. This gave us 6 PCs to use.

We than proceeded to train a linear model on only the PCs computed from the continuous regressors, obtaining the following results:

Summary	**Result**
R^2	0.6557
Residual Standard Error	0.8181 on 394 df
F-Statistic	125.1 on 6 and 394 df
p-value	$< 2.2e{-}16$
RMSE on Training Set	0.810935
RMSE on Test Set	0.7497926
RMSE on Training Set (Untransformed)	1.609407
RMSE on Test Set (Untransformed)	16.60826

Figure Table Table 4.1.2.1a: PCA results

We also added in the categorical regressors which we had omitted (since PCA involves computing distances, which cannot easily be done for discrete variables). The table below illustrates the results:

We can see that adding categorical variables to the model data increased the R^2 of the model, allowing it to explain more of the variance in `area`.

Ridge Regression (RR) RR, also known as L2 regularization, is a linear regression technique that addresses the problem of multicollinearity (high correlation among predictors) and helps prevent overfitting in models with many predictors. In traditional Linear Regression, the objective is to minimize the sum

Summary	Result
R^2	0.6925
Residual Standard Error	0.8055 on 363 df
F-Statistic	22.09 on 37 and 363 df
p-value	$< 2.2e{-}16$
RMSE on Training Set	0.6777014
RMSE on Test Set	0.7724247
RMSE on Training Set (Untransformed)	1.468539
RMSE on Test Set (Untransformed)	17.54927

Figure Table 4.1.2.1b: PCA with categorial variables results

of squared errors between the predicted values and the actual target values. RR introduces an additional penalty term proportional to the square of the magnitude of the regression coefficients to the objective function.

Mathematically, in Ridge Regression, the objective function becomes:

$$\min_{\vec{\beta}} \left\{ \sum_i \left(y_i - \vec{\beta} x_i \right)^2 + \lambda \sum_i \beta_i^2 \right\}$$

where:

- y_i is the target value for the ith observation.
- x_i is the vector of predictor values for the ith observation.
- β_i represents the regression coefficients.
- λ is the regularization parameter (also known as the ridge parameter or penalty term). It is a hyper-parameter that determines the strength of regularization.

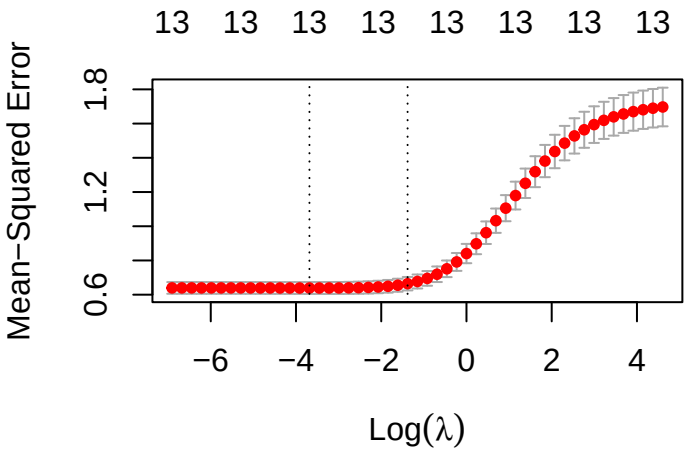
The regularization term $\sum_i \beta_i^2$ penalizes large coefficient values, encouraging them to be small. This has the effect of shrinking the regression coefficients towards zero, which helps in reducing the impact of less relevant predictors and controlling the model's complexity.

Ridge Regression strikes a balance between fitting the data well and keeping the model simple, which can improve its performance on unseen data and reduce the risk of overfitting. The optimal value of the regularization parameter λ is typically chosen through cross-validation or other model selection techniques.

The function `glmnet` from r was used to perform ridge and lasso regression where $\alpha = 0$ was for ridge regression and $\alpha = 1$ was for lasso regression.

The same testing and training sets were used for ridge regression.

The plot is the plot of all the lambda values found after using cross validation.



The results below are the values from Ridge Regression on the testing and training set:

Summary	Result
lambda	0.03981072
RMSE on Training Set	0.7553566
RMSE on Test Set	0.09413435
RMSE on Training Set (Untransformed)	10.07874
RMSE on Test Set (Untransformed)	8.825959

Figure Table 4.1.2.2: Ridge Regression results

Lasso Regression (LR) Lasso (short for Least Absolute Shrinkage and Selection Operator) Regression is a linear regression technique that performs both regularization and feature selection. It addresses the same issues as RR, such as multicollinearity and overfitting, but it achieves sparsity in the model by adding an L1 regularization term to the objective function.

Mathematically, in Lasso Regression, the objective function becomes:

$$\min_{\beta} \left\{ \sum_{i} \left(y_i - \vec{\beta} x_i \right)^2 + \lambda \sum |\beta_i| \right\}$$

where:

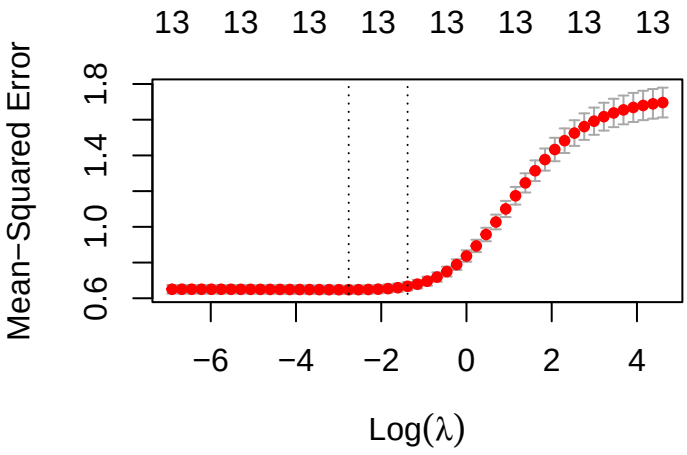
- y_i is the target value for the ith observation.
- x_i is the vector of predictor values for the ith observation.
- β_i represents the regression coefficients.
- λ is the regularization parameter (also known as the lasso parameter or penalty term). It controls the strength of regularization.

The key difference from Ridge Regression lies in the regularization term $\sum_i |\beta_i|$. This term enforces sparsity by encouraging some regression coefficients to be exactly zero. As a result, Lasso Regression can effectively perform feature selection by excluding less relevant predictors from the model, essentially shrinking them to zero.

The L1 regularization in Lasso Regression leads to a sparse model, where only a subset of the predictors has non-zero coefficients. This property can be beneficial when dealing with high-dimensional data, as it provides a more interpretable and compact model.

As with Ridge Regression, the optimal value of the regularization parameter λ is typically determined through cross-validation or other model selection techniques. Lasso Regression is particularly useful when there are many predictors in the data, and we want to identify the most important ones while keeping the model's complexity under control.

The same testing and training sets were used for ridge regression.



The results below are the values from Lasso Regression on the testing and training set:

Summary	Result
lambda	0.02511886
RMSE on Training Set	0.7647896
RMSE on Test Set	0.09701367
RMSE on Training Set (Untransformed)	9.127104
RMSE on Test Set (Untransformed)	9.093667

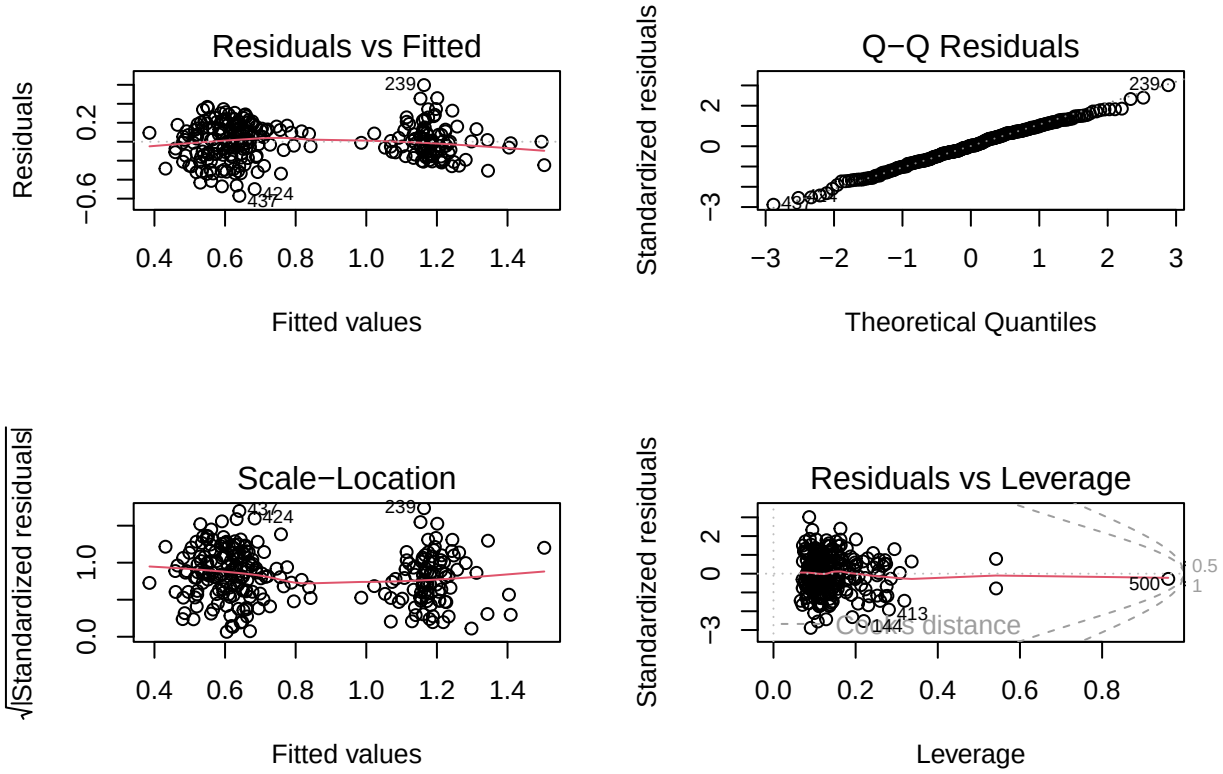
Figure Table 4.1.2.3: Lasso Regression results

Implementation on Normalized Data

Multiple Linear Regression Model

We ran an MLR model on the full dataset without taking out any outliers or unusual points. However, this was not a great model; we decided to examine and remove any unusual points in the data, and subsequently re-fit the model.

Model Statistics



Multiple Linear Regression (without outlier/influential points) The table below summarizes the diagnostics for this model:

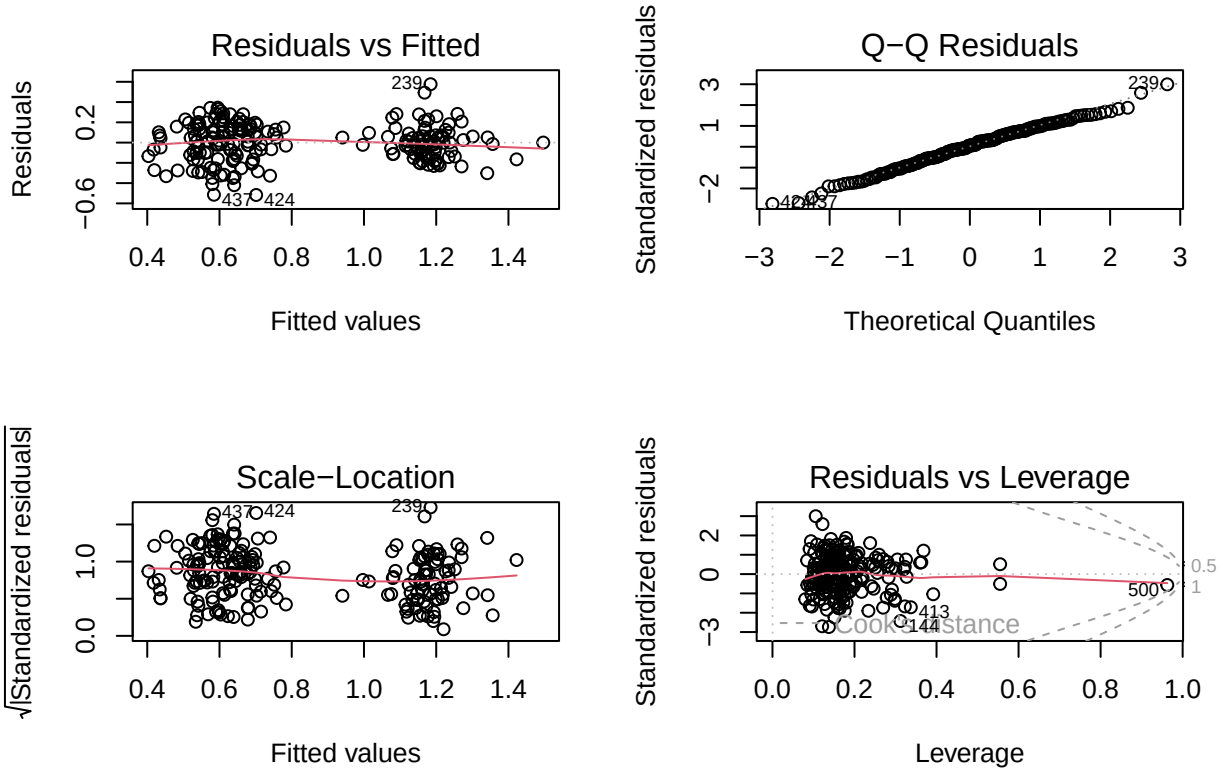
Summary	Result
R^2	0.6995
Residual Standard Error	0.2141 on 167 df
F-Statistic	10.23 on 38 and 167 df
p-value	$< 2.2e{-}16$
RMSE on Training Set	0.192803
RMSE on Test Set	0.2022793
RMSE on Training Set (Untransformed)	0.636758
RMSE on Test Set (Untransformed)	27.26443

Figure Table 4.2.1: Cleaned Data MLR results

Model diagnostics We can see that the model passed both normality and constant variance tests, and can therefore be considered valid. The plots of the model diagnostics support this statement.

Test	Result
Breusch Pagen Test	0.2232
Shapiro Wilks Test	0.8473
Durbin Watson Test	8.641×10^{-8}

Figure Table 4.2.2: Cleaned MLR Model Diagnostics



Results

The tables below compiles the RMSE values from each model executed above, for ease of comparison.

Model	Transformed Data		Untransformed Data	
	RMSE (Training)	**RMSE (Test)**	**RMSE (Training)**	**RMSE (Test)**
MLR	0.7625175	0.8369324	0.1441019	71.97844
PCR	0.6777014	0.7724247	1.468539	17.54927
Ridge	0.7553566	0.09413435	10.07874	8.825959
Lasso	0.7647896	0.09701367	9.127104	9.093667

Figure Table 5.1: RMSE Results

We see that Lasso has the smallest RMSE value among the models. This is potentially due to the fact that Lasso tends to make some coefficients zero, whereas Ridge sets coefficients close, but not equal, to zero. We

also believe that PCA has a higher RMSE than the other shrinkage methods because PCA does not perform variable selection like the other two. Next, it is incorrect to comment or believe in the RMSE values for MLR, since the data does not conform to the model's assumptions. Lastly, although the RMSE for the MLR model on the normalized data is low for the training set, it is still very high for the test set.

We believe, from these results, that Lasso regression is the most appropriate model to predict `area` from the given data.

Future Discussion

More work needs to be done on the data collection of forest fires. The data set used in this analysis was imbalanced and did not account for many other factors that are more likely to cause forest fires, such as lightening strikes[3], unattended campfires[4], human related causes, and the natural flora of the land. Furthermore, even with this data, we believe that it is possible that an ensemble method - which first classifies points into zero and nonzero `area` and subsequently applies a regression technique like one discussed above to predict `area` for the nonzero bin - might work better than MLR, and perhaps comparably to RR and LR, due to the fact that a large number of data points had `area` $= 0$ and so the additional classification error could potentially be offset by the improved accuracy of regressing conditional on $`area` \neq 0$. This was, however, outside of the scope of this course, and hence the results - while performed in the code submitted - were not reported on.

References

1. https://repositorium.sdum.uminho.pt/bitstream/1822/8039/1/fires.pdf
2. https://www.rdocumentation.org/packages/bestNormalize/versions/1.9.0
3. https://www.cbsnews.com/sacramento/video/researchers-90-of-us-wildfires-human-caused/
4. https://wfca.com/articles/what-causes-wildfires/