A Data Analytic Approach to Predict Forest Fires

Malsha Ranawaka
School of Computer, Data and Mathematical Sciences
Western Sydney University
Victoria Rd, Rydalmere NSW 2116
20321967@student.westernsydney.edu.au

!"#$%&'$– **Forest fires are a major environmental hazard that effects the lives of both humans and animals, while causing ecological and economical destruction. Effective prediction of forest fires is vital in preparing an action plan to avoid or minimize the damage caused by fires, thereby saving wildlife and their habitats. This paper discusses a data analytic approach taken to predict the occurrence of fires and the burned area in the Montesinho Natural Park in Portugal.**

()*+,%-# . ',/0,1)1$23 4,%)#$345%)30%)-5'$5,123#%,16%)#
-)'5#5,13%))#23'78#$)%3516

I. Introduction

Forest fires are among the major natural catastrophes that endanger humans and wildlife and cause severe forest destruction and environmental degradation. Accurate and timely prediction of forest fires is instrumental in preserving the natural territories and minimizing the damage of fires. Various research has been carried out to model the probability of fire and its causes, which have been used to predict the occurrence of fires and quantify the risk of fires. These techniques range from geospatial statistical analysis (Tariq, 2020) and machine learning ensembles (Tehrany, 2019) to neural network models (Zhang, 2019).

In this study, data analytic techniques are used to explore the data from forest fires that occurred from January 2000 to December 2003 in the Montesinho Natural Park in Portugal. The location of the park is shown in Figure 1. The park includes an area of 74,230 hectares and is the home to 240 species of animals (Visit Porto, 2009). The annual temperature of the park varies from 8 to 12ºC, though the temperature in summer could reach up to 40ºC. The data collection procedure is detailed by Cortez et al. (2007) in the paper following their research study.

Fig. 1. Location of Montesinho Natural Park. *(Kerdprasop, 2018)*

II. Data Description and Preprocessing

A. Dataset

The dataset was obtained from the UCI Machine Learning Repository (UCI, 2008), and contains 517 entries described using 13 attribute variables and the target variable, amount of area burnt by fire. The dataset attributes are summarized in the Table 1 below.

TABLE I. Dataset Attributes

Attribute	Description	Unit/Range
X	x-coordinate of the fire location	1 to 9
Y	y-coordinate of the fire location	1 to 9
month	Month the fire occurred	Jan to Dec
day	Day of the week the fire occurred	Mon to Sun
FFMC	Fine Fuel Moisture Code	-
DMC	Duff Moisture Code	-
DC	Drought Code	-
ISI	Initial Spread Index	-
temp	Temperature	Celsius (ºC)
RH	Relative humidity	%
wind	Speed of the wind	km/h
rain	Amount of rain	mm/m^2
area	Total burnt area	Hectares (ha)

FFMC, DMC, DC and ISI are indexes and codes which signify the fire danger based on weather conditions. These codes are defined in the Fire Weather Index (FWI), and higher values of these indexes indicate increase of burning conditions. FFMC denotes the fuel moisture content in the forest litter, while DMC indicates the fuel moisture of decomposed organic material. DC determines long-term moisture conditions, and ISI measures the speed of fire spread (Cortez, 2007).

B. Target Variable Transformation

The distribution of the target variable of the dataset ‘area’, is plotted in the first histogram of Figure 2 below. It can be seen that more than 47% of the entries in the dataset have zero

burnt area. This is due to the fact that fires with area less than 0.01 *ha* were recorded as zero values during data collection. In order to reduce the skewness, the logarithmic value of the target variable was taken. As area should be positive, the logarithmic transformation *log(area+1)* was applied as shown in the third graph (Cortez, 2007). The resulting variable is taken as the target variable for this study.

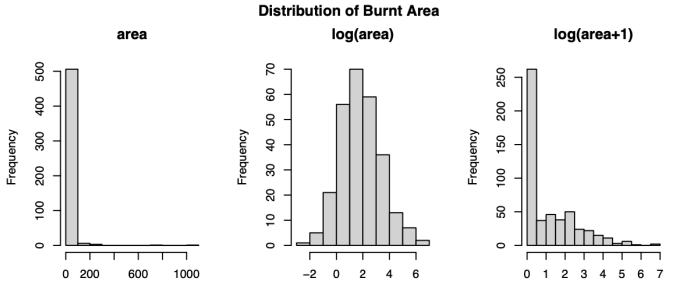


Fig. 2. Distribution of target variable before and after transformation.

C. *Linear Relationship and Correlation*

The plots below in Figure 3 show the distribution of target variable against the independent variables of the dataset. In the first 4 graphs, Area does not have an obvious linear relationship with the variables x-coordinate, y-coordinate, Month and Day of the fire. In graphs 5 and 8, the fire indexes FFMC and ISI seem to have a skewed relationship with the target variable, while in graphs 6 and 7, a clear relationship is not visible. Temperature seems to have a slightly positive relationship, while RH shows a slightly negative relationship with the target variable. In the last two graphs, Wind and Rain do not show evident relationships with the burnt area.

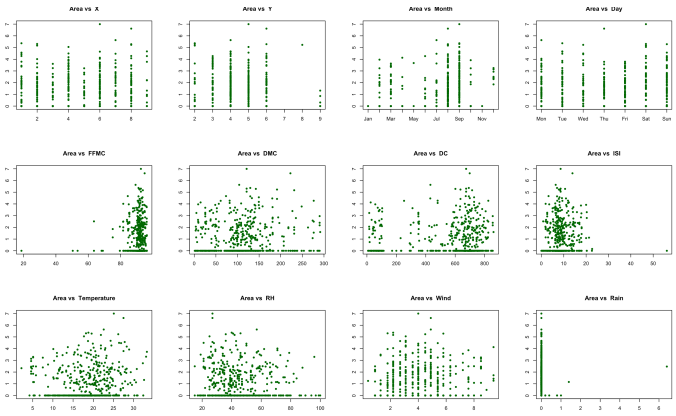


Fig. 3. Scatter plots of target variable against independent variables.

Burnt area has the highest correlation with Month, which is 0.114. It has positive correlations of close to 0.06 with the variables DMC, Wind, DC and x-coordinate. Temperature, FFMC, y-coordinate and Rain show a somewhat positive relationship with correlation coefficients ranging from 0.05 to 0.02. Day of the fire has an almost zero correlation with the area, while ISI and RH show a negative correlation.

D. *Relationship between Categorical Variables and Target*

The categorical variables in the dataset: x-coordinate, y-coordinate, Month, and Day, can be associated with geographical and environmental factors, which can be used to further capture the information contained in the dataset.

Initially, the x- and y-coordinate data were used to generate a heatmap to identify the areas that were most prone to fires. The heatmap is shown in Figure 4, where the green area corresponds to outside the park, and the colored area correspond to areas within the park. As shown in the map, it can be seen that most of the heavy fires were contained in the edges of the park, whereas the middle of the park has had to face less severe fires. However, by examining the less severe and highly burnt areas close to each other in the leftmost edge of the park, it can be seen that the fires have not spread rapidly to other areas. This may be attributed to the fact that the fires were controlled as soon as possible, therefore limiting the burnt area to a specific block.

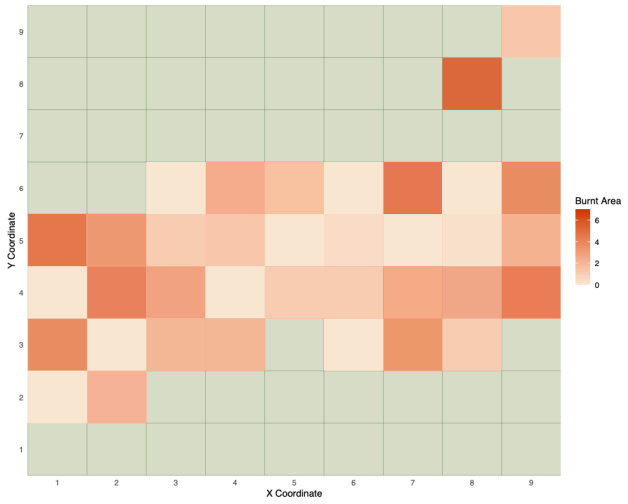


Fig. 4. Heatmap of burnt areas in the park by location coordinates.

Furthermore, the monthly data in the dataset were used to explore the seasonal variation of fires. By grouping the months by seasons and plotting the Burnt Area for each season, the graph shown in Figure 5 was generated.

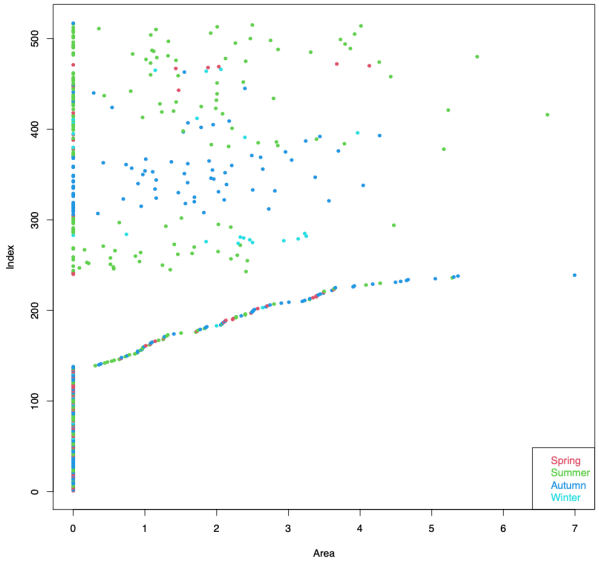


Fig. 5. Variation of Burnt Area by Seasons.

It can be seen that there has been less fires during Spring and Winter, as expected, and there have been more fires during Summer and Autumn. However, a clear association between the amount of burnt area and the season the fire occurred cannot be identified through the graph. There had been many fires corresponding to zero values during Summer, while there

had been fires corresponding to values from 2 to 4 during Winter. Therefore, an evident relationship cannot be identified using the seasonal variations for the dataset.

III. Regression Tree Modelling

As the linear relationship between Area and the independent variables is low, decision tree modelling was first applied to further investigate the dataset.

A regression tree model was created using the complete dataset to explore the behavior of the variables. The resulting tree contained 6 terminal nodes, and was constructed using only the Month, Temperature, DMC and DC variables. Then the dataset was split into train and test sets to build and test the tree model. This tree gave a Mean Squared Error (MSE) of 2.79 for the test set. In order to find a better model for the tree, cross validation was applied to this model. The resulting plot of deviation explained by the tree against the size of the tree is shown in Figure 6.

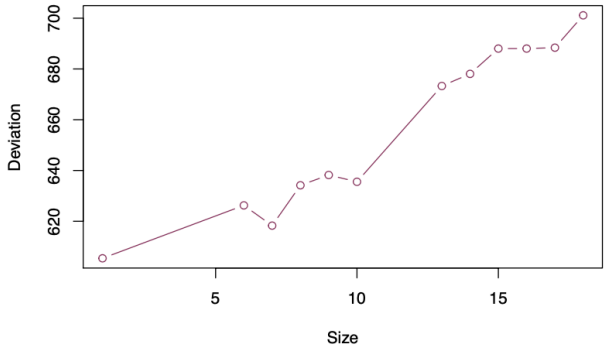


Fig. 6. Variation of deviation of the tree against the size of the tree.

As seen in the plot, the deviation is lowest when the tree has only one terminal node and increases as more terminal nodes are added to the tree. When the number of terminal nodes is 7, however, there is a drop in the deviation, which corresponds to the lowest deviation in the plot. Therefore, the tree is pruned at 7 terminal nodes, which gave an MSE of 2.36. Pruning the tree increased the regression accuracy, while reducing the complexity of the tree. The pruned tree shown in the Figure 7 below has summarized the dataset using only 5 variables.

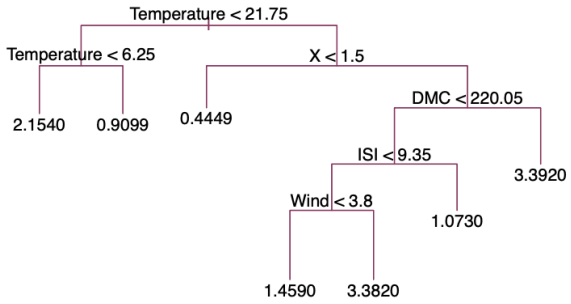


Fig. 7. Pruned regression tree.

Regression tree model improvement is summarized in the Table II below. Among the factors considered in comparing the tree models, the MSE signifies the number of unseen instances predicted correctly by the model, the number of terminal nodes denote the complexity of the model and the number of variables used in the model increases the interpretability of the tree. For this regression tree, although using the complete dataset gave a lower MSE, it has used only 4 variables in the model. In contrast, the pruned tree explains the data using 5 variables and has 7 terminal nodes and has shown an increase in the testing MSE compared to the unpruned tree.

TABLE II. Regression Tree Models

	Whole Dataset	Before Pruning	After Pruning
Mean Squared Error	1.778	2.791	2.364
Terminal Nodes	6	18	7
No of Variables	4	8	5

The plot in Figure 8 shows the distribution of actual target values of the dataset and the values predicted using the pruned regression tree. The straight line is plotted along $y = x$, denoting the equal predicted and actual values. It can be seen that while actual values range from 0 to 5, predicted values have a smaller range of from 0.5 to 3.5. This is due to the fact that the predicted values are based on the mean values at each terminal node of the tree. The predicted axis consists of values that belong to several straight lines, which correspond to the terminal nodes. Therefore, even though the MSE value is low for the model, most values are not predicted exactly.

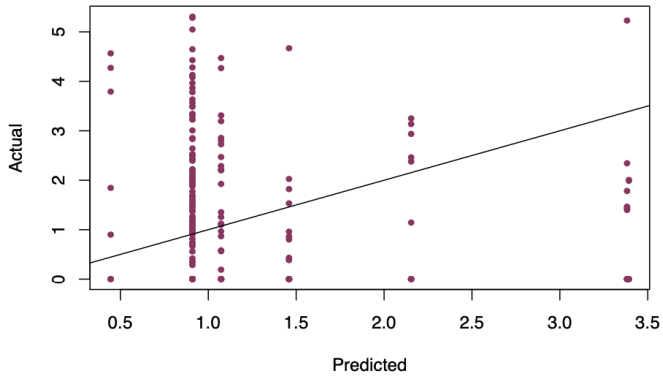


Fig. 8. Similarity between predicted and actual values.

IV. Multiple Linear Regression

For Linear regression, only the numeric variables of the dataset were selected to be fit into the model, leaving out the x-coordinate, y-coordinate, Month and Day attributes. The resulting dataset was normalized to bring the attributes to the same scale between 0 and 1. In order to explore the effect of linear regression models on the dataset, the models were built using the complete dataset.

A. Model 1 – Using All Numerical Variables

The initial regression model was built for all the numerical variables in the dataset as follows:

```
Area = FFMC + DMC + DC + ISI + Temperature
          + RH + Wind + Rain
```

The model summary showed that only Wind parameter had the highest significance among predictor variables with a P-value of 0.039 and a one-star (*) rating. ISI and RH had a negative relationship with the target variable but showed a high significance with P-values of 0.158 and 0.319 respectively. Both DMC and DC parameters had a similar low significance with a P-value of 0.4, while FFMC and Rain showed higher P-

values. Temperature had the lowest significance among the predictors, with a P-value of 0.887. Furthermore, the distribution of residuals had a maximum value of 0.8165 while its minimum was -0.2173, which indicated that the residuals are not normally distributed. The model had an R^2 of 0.01988 which explains the proportion of the variance captured by the model. As the R^2 value is rather small, further combinations of predictor variables were considered for regression.

The diagnostic plots for the initial model are shown in Figure 9. The diagnostic plots of a model expose how well the data is represented by the model. In graph 1, the residuals are not equally spread around a horizontal line, which indicates there is a non-linear relationship that is not captured by the model. Graph 2 shows if residuals are normally distributed. Here, the residuals deviate from a straight line, which indicates that residuals are not normally distributed. The third graph explains if residuals are equally spread along the ranges of predictors. In graph 3, the residuals appear to have very small and very large values, which disrupts the equal spread of values, and the line shows a slight angle as the x values increase. Graph 4 enables finding influential or extreme values, which may alter the results of regression if they were excluded from the analysis. Such points are identified by having a high Cook's distance and appear outside the dashed line that shows Cook's distance. Here it is seen that the 500th observation is an influential value, indicated by a high Cook's distance score.

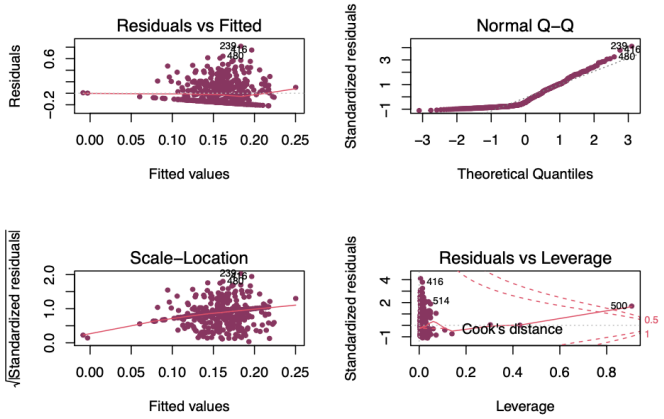


Fig. 9. Diagnostic plots for Model 1 of Multiple Linear Regression.

Model 1 Improvement – Excluding Influential Data Points

As identified using the first model, the 500th observation was removed from the data set to observe the effect of the data point. The model gave an R^2value of 0.02361, which is higher when compared to the 0.01988 from the previous model. Diagnostic plots from the resulting model are shown in Figure 10.

The 4th graph shows that the point 510 is closer to the Cook's distance line. To explore the effect, it was further excluded from the study. This resulted in a model with an increased R^2 value of 0.02562. The model still recorded Wind as the most significant variable with a P-value of 0.0241, and the least significant variable was Temperature, with a P-value of 0.749, indicating that its significance had increased. The diagnostic plots for the improved Model 1 excluding both 500 and 510 data points is shown in the Figure 11.

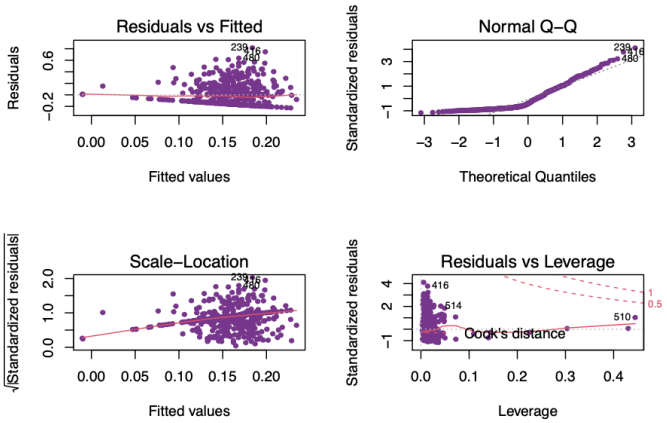


Fig. 10. Diagnostic plots after excluding 500th data point.

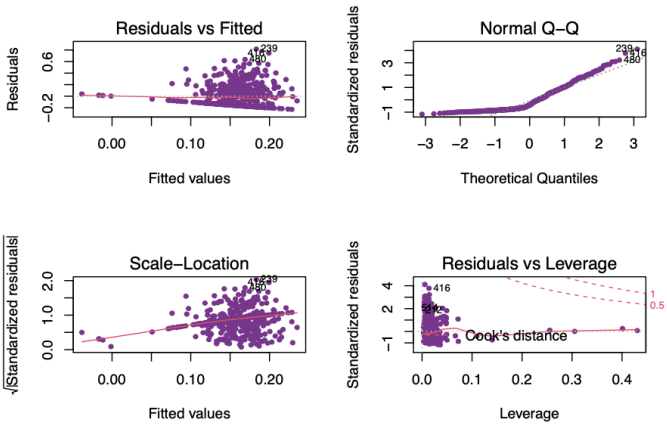


Fig. 11. Diagnostic plots after excluding 500 and 510 data points.

The first, second and third diagnostic plots do not show much difference as only two data points were excluded. The fourth graph, however, shows that the remaining data points are farther away from the Cook's distance lines, indicating that extreme data points are not present in the dataset.

The R^2 and adjusted R^2 squared values were found to be farther from each other for this model, with an R^2 of 0.02562 and an adjusted R^2 of 0.01022. The adjusted R^2 value incorporates the number of variables used in the model to adjust R^2 for the complexity of the model. A close R^2 and adjusted R^2 indicates that the model is sufficiently generalized, and not overfit to the dataset used. As the values are far from each other, the model may not perform well with new data points, as it is not sufficiently generalized.

B. Model 2 – Using Step-wise Feature Selection

As Model 1 only contained one significant variable, the next model was created by performing stepwise feature selection to choose the most significant variables. The resulting model contained DMC, Wind and Rain variables as follows:

```
Area = DMC + Wind + Rain
```

The model had high significance for all the variables used: with DMC and Wind having a one-dot (.) level of significance and Rain having the highest significance with a P-value of 0.0415, indicated by one star. The model gave an R^2 value of 0.01774 and an adjusted R^2 value of 0.01198. It can be seen

that the R^2 and adjusted R^2 values are close to each other, indicating that the model will perform well for unseen data.

The diagnostic plots for Model 2 are shown in Figure 12. The first graph shows a near linear line, but the data points are not equally spread around it, indicating that a non-linear relationship exists. The second graph is similar to that of the first, model, showing that the residuals are not normally distributed. The slanted straight line in graph 3 denotes that the residuals are not equally spread, violating the assumption of homoscedasticity. Graph 4 does not point to any extreme data points, although some points are seen to be farther away from the rest of the data points.

Model 2 with coefficient estimates can be presented as follows:

```
Area = 0.074*DMC + 0.084*Wind - 1.662*Rain
```

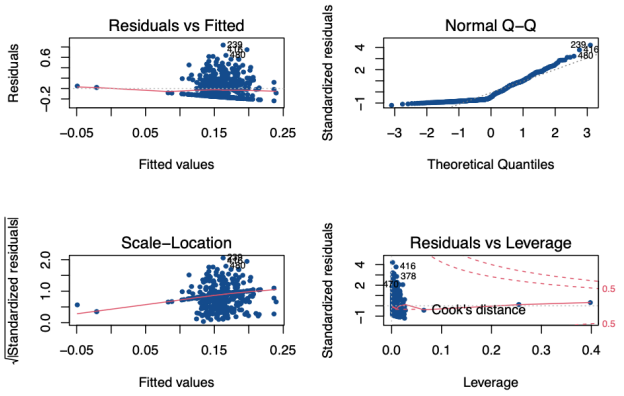


Fig. 12. Diagnostic plots for Model 2 of Multiple Linear Regression.

V. Nonlinear Regression

Since the linear regression models did not capture the relationship to the target variable effectively, its nonlinear relationship was explored using the pair plots shown below.

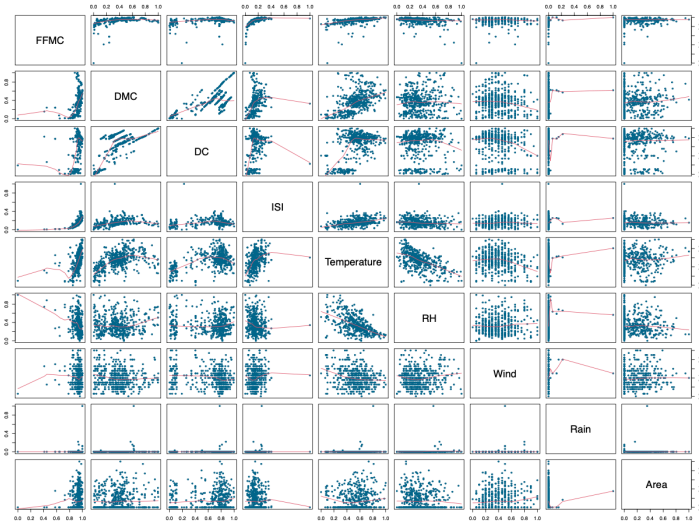


Fig. 13. Pair plots for the dataset.

The last row of the above pair plot shows the variation of Area against each predictor variable. It can be seen that Temperature has an evident curve in its graph, indicating a possible non-linear relationship. Furthermore, if the outlier points in the graphs of FFMC and ISI were disregarded, it can be seen that majority of data points have a curved distribution with the target variable. These variables were added as polynomial terms to the model with an order of 2.

To add interaction terms, the relationship among predictor variables was considered. If two predictor variables did not show any evident relationship among each other, they were added as interaction terms to the model. For instance, Temperature and Wind variables show an equally spread set of points, which does not signify a clear connection. Therefore, the combination of Temperature and Wind was added to the model as an interaction term. Another instance is the plot of Wind against ISI, which indicates that ISI has a curved relationship with Wind. Therefore, an interaction term containing a polynomial term was added to the model as ISI^2*Wind. After training the model, variables that had a low significance (high P-value) such as ISI and RH were removed to improve the model.

A. Model 1 – Using Polynomial and Interaction Terms

The resulting model containing polynomial and interaction terms was as follows:

```
Area = FFMC + DMC + DC + Temperature +
Wind + Rain + Temperature^2 + ISI^2 + FFMC^2 +
DMC^2 + Wind^2 + DC*ISI + FFMC*ISI +
RH*Wind*Rain + Temperature*Wind +
Temperature*DMC + ISI^2*Wind
```

The model had an R^2 value of 0.05345, which is higher than those given by the linear models. Among the predictor variables, `Temperature`^{2} had the highest significance, indicated by a dot (.) rating and a P-value of 0.0687. `Temperature`, `Wind`, `ISI`^{2}, `Temperature*Wind` and `ISI`^{2}`*Wind` had lower P-values indicating high significance, while `DMC`, `Rain`, `Wind`^{2}, `FFMC*ISI` had high P-values, indicating lower significance.

The diagnostic plots for the model shown in Figure 14 were examined to identify further information from the model.

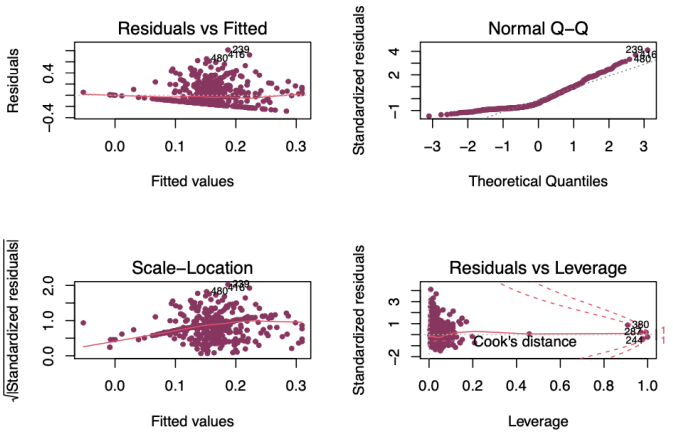


Fig. 14. Diagnostic plots for Model 1 of Non Linear Regression.

The Residual graph indicates an almost horizontal line, however more points are spread above the line than below, indicating the presence of a behavior not captured in the model. The Normal Q-Q graph is similar to that of the linear models and shows that the residuals are not normally distributed. The Scale-Location graph consists of a straight line with a slant, indicating that residuals are not equally distributed among predictor variable ranges. The Leverage graph shows that few points are on the margin of Cook's distance, indicating that they could be extreme cases with regard to the model.

Although this model had a high R^2 value of 0.05345 compared to the non-linear models, the adjusted R^2 of the model was 0.01513, a value much lower than R^2. This indicates that although the model is capturing more information from the dataset causing R^2 to increase, it is not well generalized and may overfit to the dataset. This is reflected by the low significance of the majority of variables in the model.

B. *Model 2 – Using Most Significant Terms*

The above model was further simplified by removing variables with high P-values (low significance), in order to improve the significance of the model. The resulting model was as follows:

```
Area = FFMC + DC + Temperature +
Temperature² + FFMC² + Wind² + DC*ISI +
RH*Rain + ISI²*Wind
```

The model gave an R^2 value of 0.04602 which was lower than the first non-linear model which had an R^2 of 0.05345. However, this model had more significant terms, of which `Temperature²` had the highest significance with three stars and a P-value of 0.000907. `Temperature` and `DC` had high significances with two- and one-star ratings respectively. The interaction term `RH*Rain` had a significance rating of one dot (.), with a P-value of 0.063176.

The diagnostic plots for the simplified model are shown in Figure 15. As seen in the graphs, the residual distribution shown in first and second graphs does not show much difference from the previous non-linear model. The fourth graph shows that some points are close to the Cook's distance and could represent extreme data points in the model.

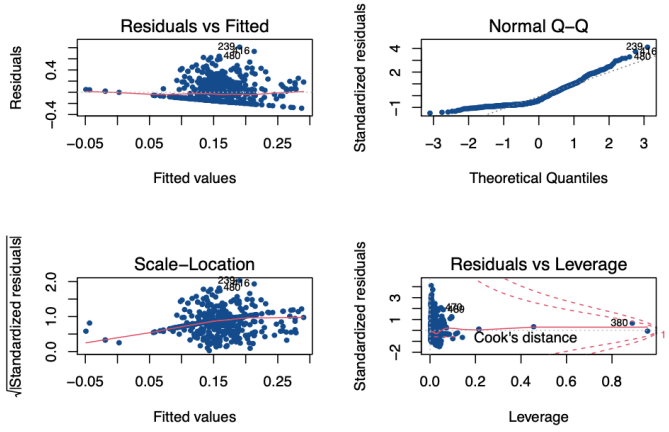


Fig. 15. Diagnostic plots for Model 2 of Non Linear Regression.

The R^2 and adjusted R^2 of the model were 0.04602 and 0.02709 respectively. Although the R^2 value has decreased compared to the first non-linear model, the adjusted R^2 value has increased from 0.01513, indicating that the model is more generalized to the dataset. The decrease of R^2 occurs due to the fact that as the variables are removed to simplify the model, less variance is captured in the model itself. However, as the model becomes less complex, the significance of the variables increases, reducing the model overfit to the given dataset. This results in increased adjusted R^2, and reduced difference between R^2 and adjusted R^2.

VI. Clustering

Clustering, which is an unsupervised technique, was applied to the dataset to explore the possible clusters or groupings, which could expose further insight into the dataset.

A. *K-means Clustering*

K-means clustering was used on the numerical attributes of the dataset, excluding the x-coordinate, y-coordinate, Month and Day variables. As it is an unsupervised technique, the target variable Area was removed as well. As K-means require the number of clusters to be supplied to build the model, initially two clusters were built using the algorithm. The resulting clusters against each predictor and target variable is shown in Figure 16.

When k = 2, it can be seen that distinct two clusters can be seen in the variables FFMC, DMC, DC and Temperature. RH shows a slight separation, while RH, Wind, Rain and the target variable Area, do not show any distinct two clusters.

The cluster distribution when k = 3, is shown in Figure 17. In this model, it can be seen that DMC, DC and Temperature variables show distinct clusters, while in other variables, the behavior is not evident. Furthermore, it shows that the target variable, Area, does not show any distinct grouping based on K-means clustering.

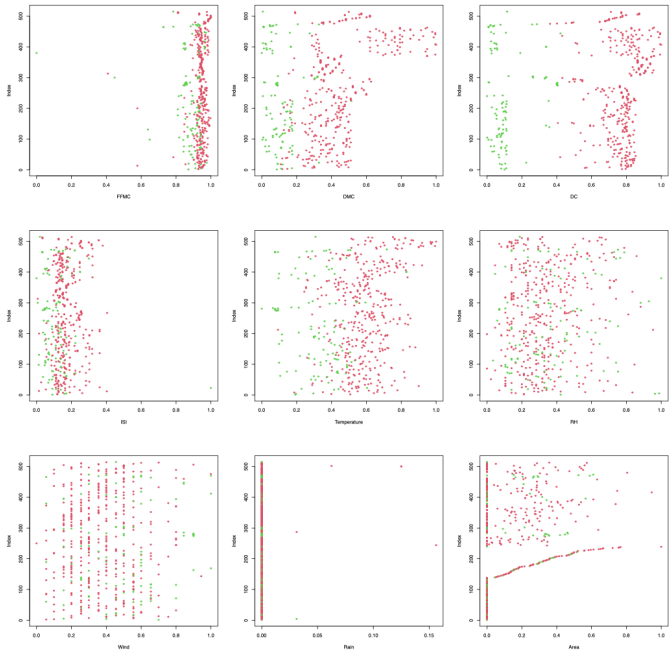


Fig. 16. K-means clusters for *k = 2*.

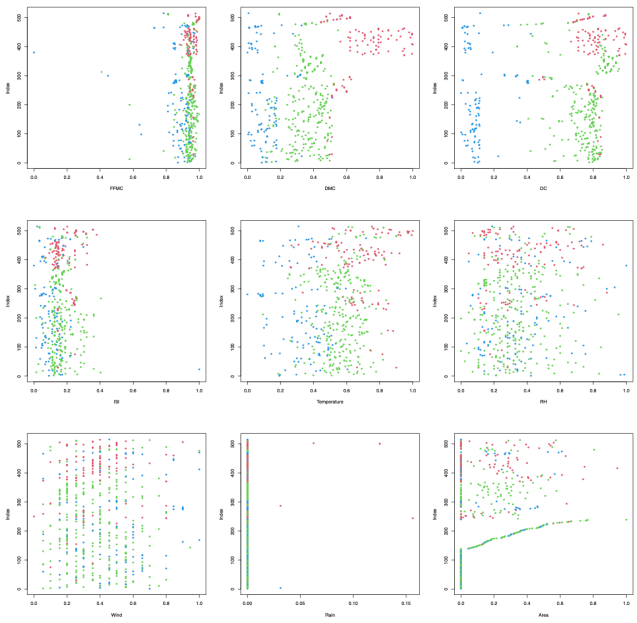


Fig. 17. K-means clusters for $k = 3$.

B. *Hierarchical Clustering*

Next, agglomerative clustering, which is a hierarchical clustering method, was applied to the dataset. Agglomerative clustering processes in a bottom-up approach, starting from each individual point as a cluster and merging them based on the distance to neighboring clusters. Three distance calculation methods, minimum, maximum and average, are used to calculate the distance when merging clusters. Hierarchical clustering can be performed based on these three distance measures, which correspond to the three linkage types, single, complete and average respectively. These three linkage methods were applied to the dataset, to identify the best distance measure that groups data points effectively. The dendrograms resulting from applying the tree methods are shown in the figures below.

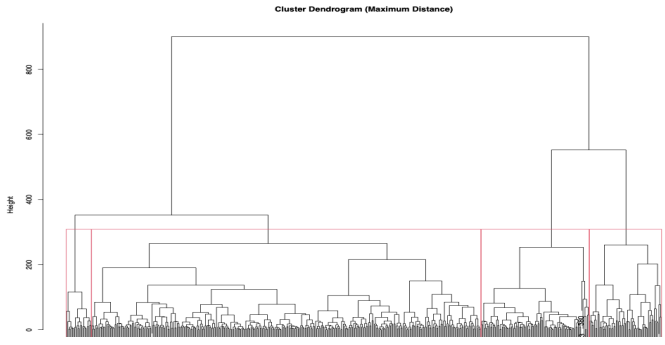


Fig. 18. Hierachical clustering with complete linkage.

This dendrogram depicts that there are four clusters that can be distinctly identified using the gaps before splits. It was found that the four clusters had 94, 338, 63 and 22 data points. Using average distance as shown in Figure 19 gives four distinct clusters similar to complete linkage, with 88, 317, 69 and 43 data points in each of them.

Using minimum distance results in rapid grouping of clusters as shown in Figure 20. It shows that there are 3 distinct clusters, with two rather small and one large cluster. When grouped to 4, the clusters had 87, 428, 1 and 1 data points in each of them, resulting in grouping a large portion of the dataset to one cluster.

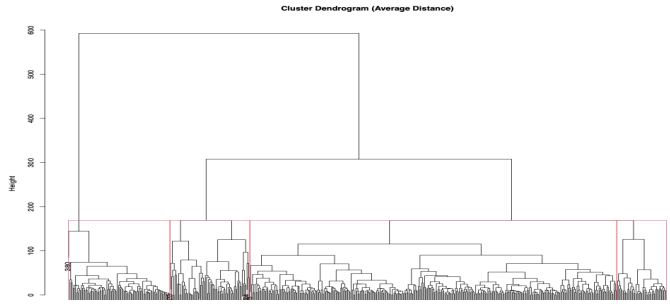


Fig. 19. Hierachical clustering with average linkage.

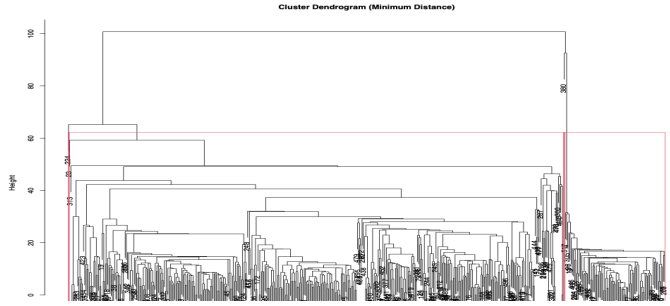


Fig. 20. Hierachical clustering with simple linkage.

When comparing the above dendrograms, it can be seen that using complete linkage results in more balanced clusters.

VII. Model Comparison and Evaluation

A. *Comparison of Regression Model Characteristics*

Linear and nonlinear regression models are summarized in the Table III below.

TABLE III. Regression Model Characteristics

	Multiple Linear Regression		Nonlinear Regression	
	Model 1	**Model 2**	**Model 1**	**Model 2**
R^2	0.02562	0.01774	**0.05345**	0.04602
Adjusted R^2	0.01022	0.01198	0.01513	**0.02709**
Delta	0.04288	**0.04162**	0.04632	0.04292
MSE	0.03844	0.03794	0.03941	**0.03754**
Significant Parameters	2 of 8	3 of 3	1 of 20	4 of 10
P-value	0.1048	0.02727	0.1185	**0.00778**

As seen in the above table, the Model 2 of Nonlinear Regression has most of the best characteristics considered, including the highest adjusted R^2, lowest Mean Squared Error and the lowest p-value denoting the high significance of the model. Therefore, the second nonlinear regression model was chosen as the best model for the prediction task at hand.

B. *Final Model Evaluation*

The final model had the following parameters, of which the coefficient estimates and p-values are shown in Table IV.

TABLE IV. FINAL MODEL ESTIMATES

Parameter	Coefficient Estimates	P-values
FFMC	0.42944	0.424699
DC	0.10483	0.045782 (*)
Temperature	-0.61668	0.004045 (**)
$Temperature^2$	0.63916	0.000907 (***)
$FFMC^2$	-0.17825	0.664921
$Wind^2$	0.06772	0.345545
DC*ISI	-0.23758	0.400599
RH*Rain	-2.35624	0.063176 (.)
ISI^2*Wind	1.14818	0.499235
ISI^2	-0.60453	0.457072
Intercept	-0.01017	0.956899

By observing the above model estimates, the final model can be presented as follows:

$$Area = -0.01017 + 0.42944*FFMC + 0.10483*DC - 0.61668*Temperature + 0.63916*Temperature^2 - 0.17825*FFMC^2 + 0.06772*Wind^2 - 0.23758*DC*ISI - 2.35624*RH*Rain + 1.14818*ISI^2*Wind - 0.60453*ISI^2$$

Four parameters of the model have significant P-values, which have catered to increase the adjusted R^2 value 0.02709 of the model. When comparing with the other models, the adjusted R^2 indicates that the final model has captured variance in the dataset while keeping the model sufficiently generalized, so that the model would perform well for unseen data. However, it indicates that only a 2% of the variance of the dataset had been captured, which shows that the model may not perform predictions well.

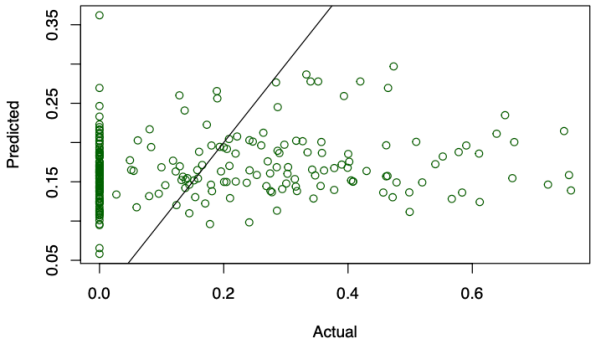


Fig. 21. Predicted against actual values for the final model.

In contrast, the model has a significantly low Mean Squared Error (MSE) value of 0.03754, indicating the model would perform well in predicting the target variable. However, when the predicted and actual values are plotted against each other as shown in Figure 21, it can be seen that only a few data points are correctly regressed by the model. This is due to the very low variance of the target variable, along with the low R^2, that gives the low MSE as shown by the equation 1.

$$R^2_{adj} = 1 - MSE / \sigma^2_y \qquad (1)$$

The distribution of the residuals of the model is shown below in Figure 22, depicting a skewed distribution, which indicates that residuals are not normally distributed.

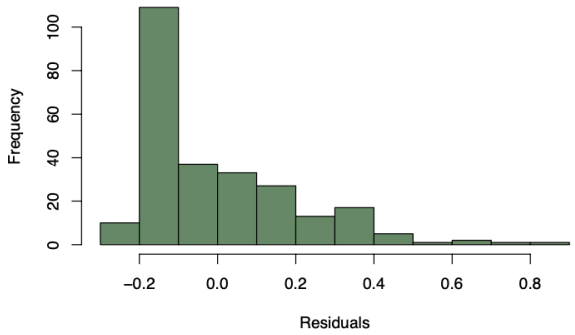


Fig. 22. Distribution of residuals for the final model.

VIII. CONCLUSION

The main objective of this research was to develop an approach based on data analytics to accurately the predict the amount of burnt area due to forest fires. The implemented model indicated that weather conditions such as temperature, rain, relative humidity and Drought Code (DC) play an important role in defining the severity of fires. However, further analysis needs to be carried out to investigate the effect of each parameter in detail and explore building regression models for groups of similar data points, which can be discovered through clustering.

ACKNOWLEDGMENT

The author would like to express their gratitude to Dr. Liwan Liyanage for their continuous support and guidance throughout the Masters Degree program, which proved quite beneficial for the completion of this project.

REFERENCES

[1] Cortez, P. and Morais, A.D.J.R., 2007. A data mining approach to predict forest fires using meteorological data.

[2] Kerdprasop, N., Poomka, P., Chuaybamroong, P. and Kerdprasop, K., 2018. Forest Fire Area Estimation using Support Vector Machine as an Approximator. In *IJCCI* (pp. 269-273).

[3] Tariq, A., Shu, H. and Siddiqui, S., 2020. Monitoring Forest Fire using Geo-Spatial Information Techniques and Spatial Statistics: One Case Study of Forest fire in Margalla Hills, Islamabad, Pakistan.

[4] Tehrany, M.S., Jones, S., Shabani, F., Martínez-Álvarez, F. and Bui, D.T., 2019. A novel ensemble modeling approach for the spatial prediction of tropical forest fire susceptibility using logitboost machine learning classifier and multi-source geospatial data. *Theoretical and Applied Climatology*, *137*(1-2), pp.637-653.

[5] UCI. 2008. *UCI Machine Learning Repository: Forest Fires Data Set*. [online] Available at: <https://archive.ics.uci.edu/ml/datasets/forest+fires> [Accessed 14 October 2020].

[6] Visit Porto. 2009. *Montesinho Natural Park*. [online] Available at: <http://www.visitportoandnorth.travel/Porto-and-the-North/Visit/Artigos/Montesinho-Natural-Park> [Accessed 14 October 2020].

[7] Zhang, G., Wang, M. and Liu, K., 2019. Forest fire susceptibility modeling using a convolutional neural network for yunnan province of china. *International Journal of Disaster Risk Science*, *10*(3), pp.386-40

APPENDIX

1. Importing Packages

```
library(ISLR)
library(tree)
library(ggplot2)
library(boot)
```

2. Data Description

```
# read and attach dataset
fire_data = read.csv("forestfires.csv")
attach(fire_data)

# get first few lines of the dataset
head(fire_data)
```

```
##   X Y month day FFMC  DMC    DC  ISI temp RH wind rain area
## 1 7 5   mar fri 86.2 26.2  94.3  5.1  8.2 51  6.7  0.0    0
## 2 7 4   oct tue 90.6 35.4 669.1  6.7 18.0 33  0.9  0.0    0
## 3 7 4   oct sat 90.6 43.7 686.9  6.7 14.6 33  1.3  0.0    0
## 4 8 6   mar fri 91.7 33.3  77.5  9.0  8.3 97  4.0  0.2    0
## 5 8 6   mar sun 89.3 51.3 102.2  9.6 11.4 99  1.8  0.0    0
## 6 8 6   aug sun 92.3 85.3 488.0 14.7 22.2 29  5.4  0.0    0
```

```
# get the size of the dataset
dim(fire_data)
```

```
## [1] 517  13
```

```
# class of each variable in the dataset
sapply(fire_data, class)
```

```
##           X           Y       month         day        FFMC         DMC
##   "integer"   "integer" "character" "character"   "numeric"   "numeric"
##          DC         ISI        temp          RH        wind        rain
##   "numeric"   "numeric"   "numeric"   "integer"   "numeric"   "numeric"
##        area
##   "numeric"
```

```
# structure of the dataset
str(fire_data)
```

```
## 'data.frame':    517 obs. of  13 variables:
##  $ X    : int  7 7 7 8 8 8 8 8 8 7 ...
##  $ Y    : int  5 4 4 6 6 6 6 6 6 5 ...
##  $ month: chr  "mar" "oct" "oct" "mar" ...
##  $ day  : chr  "fri" "tue" "sat" "fri" ...
##  $ FFMC : num  86.2 90.6 90.6 91.7 89.3 92.3 92.3 91.5 91 92.5 ...
##  $ DMC  : num  26.2 35.4 43.7 33.3 51.3 ...
##  $ DC   : num  94.3 669.1 686.9 77.5 102.2 ...
##  $ ISI  : num  5.1 6.7 6.7 9 9.6 14.7 8.5 10.7 7 7.1 ...
##  $ temp : num  8.2 18 14.6 8.3 11.4 22.2 24.1 8 13.1 22.8 ...
##  $ RH   : int  51 33 33 97 99 29 27 86 63 40 ...
##  $ wind : num  6.7 0.9 1.3 4 1.8 5.4 3.1 2.2 5.4 4 ...
##  $ rain : num  0 0 0 0 0.2 0 0 0 0 0 ...
##  $ area : num  0 0 0 0 0 0 0 0 0 0 ...
```

```
# summary of the dataset
summary(fire_data)
```

```
##        X               Y          month              day
##  Min.   :1.000   Min.   :2.0   Length:517         Length:517
##  1st Qu.:3.000   1st Qu.:4.0   Class :character   Class :character
##  Median :4.000   Median :4.0   Mode  :character   Mode  :character
##  Mean   :4.669   Mean   :4.3
##  3rd Qu.:7.000   3rd Qu.:5.0
##  Max.   :9.000   Max.   :9.0
##       FFMC            DMC              DC              ISI
##  Min.   :18.70   Min.   :  1.1   Min.   :  7.9   Min.   : 0.000
##  1st Qu.:90.20   1st Qu.: 68.6   1st Qu.:437.7   1st Qu.: 6.500
##  Median :91.60   Median :108.3   Median :664.2   Median : 8.400
##  Mean   :90.64   Mean   :110.9   Mean   :547.9   Mean   : 9.022
##  3rd Qu.:92.90   3rd Qu.:142.4   3rd Qu.:713.9   3rd Qu.:10.800
##  Max.   :96.20   Max.   :291.3   Max.   :860.6   Max.   :56.100
##       temp             RH              wind            rain
##  Min.   : 2.20   Min.   : 15.00   Min.   :0.400   Min.   :0.00000
##  1st Qu.:15.50   1st Qu.: 33.00   1st Qu.:2.700   1st Qu.:0.00000
##  Median :19.30   Median : 42.00   Median :4.000   Median :0.00000
##  Mean   :18.89   Mean   : 44.29   Mean   :4.018   Mean   :0.02166
##  3rd Qu.:22.80   3rd Qu.: 53.00   3rd Qu.:4.900   3rd Qu.:0.00000
##  Max.   :33.30   Max.   :100.00   Max.   :9.400   Max.   :6.40000
##       area
##  Min.   :   0.00
##  1st Qu.:   0.00
##  Median :   0.52
##  Mean   :  12.85
##  3rd Qu.:   6.57
##  Max.   :1090.84
```

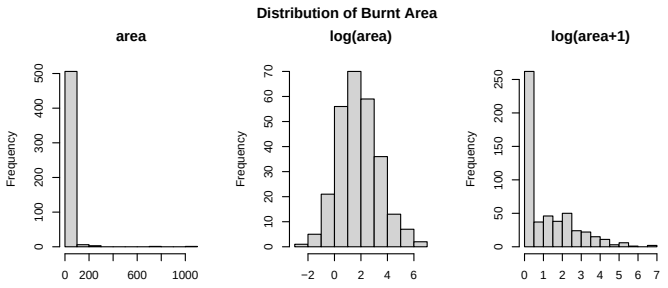
```
# check if dataset contains NA values
colSums(is.na(fire_data))
```

```
##     X     Y month   day  FFMC   DMC    DC   ISI  temp    RH  wind  rain  area
##     0     0     0     0     0     0     0     0     0     0     0     0     0
```

3. Data Preprocessing

3.1 Target Variable

```
# plot the distribution of the target variable
par(mfrow=c(1,3))
hist(fire_data$area, main = "area", xlab = "")
hist(log(fire_data$area), main = "log(area)", xlab = "")
hist(log(fire_data$area+1), main = "log(area+1)", xlab = "")
mtext("Distribution of Burnt Area", line = -1, outer = TRUE, cex = 0.8, font=2)
```




```
# calculate the percentage of zeros in the target variable
sum(fire_data$area==0)*100/dim(fire_data)[1]
```

```
## [1] 47.77563
```

```
# transform target variable of the dataset
fire_data$area_log <- log(fire_data$area+1)
```

3.2 Mapping categorical variables to numbers

```
# map month names to numbers
fire_data$month_num <- match(fire_data$month, tolower(month.abb))

days = c('Mon', 'Tue', 'Wed', 'Thu', 'Fri', 'Sat', 'Sun')
# map day names to numbers
fire_data$day_num <- match(fire_data$day, tolower(days))
```

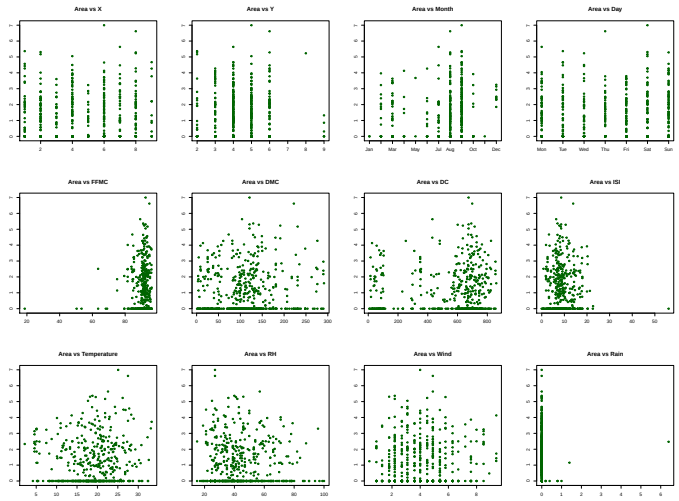
3.3 Transformed dataset

```
# create the transformed dataset
transformed_data <- fire_data[1:2]
transformed_data$Month <- fire_data$month_num
transformed_data$Day <- fire_data$day_num
transformed_data <- cbind(transformed_data, fire_data[5:8])
transformed_data$Temperature <- fire_data$temp
transformed_data <- cbind(transformed_data, fire_data[10])
transformed_data$Wind <- fire_data$wind
transformed_data$Rain <- fire_data$rain
transformed_data$Area <- fire_data$area_log
head(transformed_data)
```

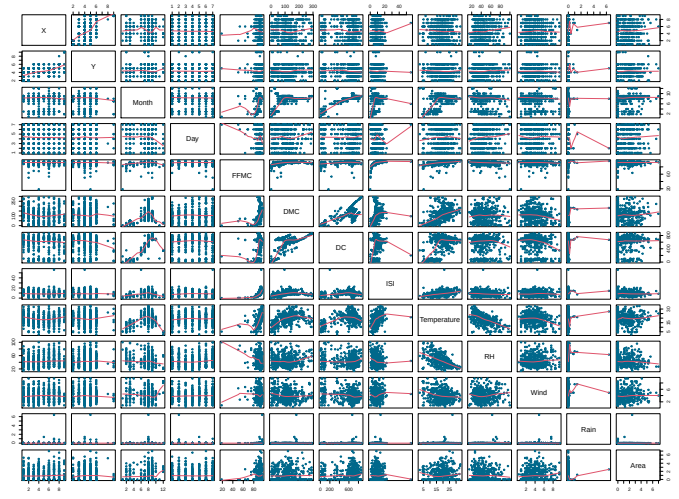
```
##   X Y Month Day FFMC  DMC    DC  ISI Temperature RH Wind Rain Area
## 1 7 5     3   5 86.2 26.2  94.3  5.1         8.2 51  6.7  0.0    0
## 2 7 4    10   2 90.6 35.4 669.1  6.7        18.0 33  0.9  0.0    0
## 3 7 4    10   6 90.6 43.7 686.9  6.7        14.6 33  1.3  0.0    0
## 4 8 6     3   5 91.7 33.3  77.5  9.0         8.3 97  4.0  0.2    0
## 5 8 6     3   7 89.3 51.3 102.2  9.6        11.4 99  1.8  0.0    0
## 6 8 6     8   7 92.3 85.3 488.0 14.7        22.2 29  5.4  0.0    0
```

3.4 Linear relationship and Correlation

```r
# plot linear relationship of target variable against independent variables
par(mfrow=c(3, 4))
fire_data_x <- transformed_data[-13]
var_names <- colnames(fire_data_x)

i <- 1
for (col in fire_data_x){
  plot_title <- paste("Area vs", var_names[i])

  if (var_names[i] == 'Month') {
    plot(col, transformed_data$Area,
        main = plot_title, xlab = "", ylab = "",
        type = "p", col = colors()[81], pch = 20, xaxt = "n")
    axis(1, at=1:12, labels=month.abb)
  }
  else if (var_names[i] == 'Day'){
    plot(col, transformed_data$Area,
        main = plot_title, xlab = "", ylab = "",
        type = "p", col = colors()[81], pch = 20, xaxt = "n")
    axis(1, at=1:7, labels=days)
  } else {
    plot(col, transformed_data$Area,
        main = plot_title, xlab = "", ylab = "",
        type = "p", col = colors()[81], pch = 20)
  }
  i <- i+1
}
```




```r
# plot the pairwise correlation plots
pairs(transformed_data, panel = panel.smooth, pch = 20, col = colors()[125])
```




```r
# get the correlation coefficients
cor(transformed_data, method = "pearson")
```

```
##                    X           Y       Month         Day        FFMC
## X          1.000000000  0.539548171 -0.06500303 -0.0249218945 -0.02103927
## Y          0.539548171  1.000000000 -0.06629179 -0.0054533368 -0.04630755
## Month     -0.065003032 -0.066291786  1.00000000 -0.0508365920  0.29147677
## Day       -0.024921895 -0.005453337 -0.05083659  1.0000000000 -0.04106833
## FFMC      -0.021039272 -0.046307546  0.29147677 -0.0410683308  1.00000000
## DMC       -0.048384178  0.007781561  0.46664525  0.0628703973  0.38261880
## DC        -0.085916123 -0.101177767  0.86869776  0.0001049027  0.33051180
## ISI        0.006209941 -0.024487992  0.18659697  0.0329092595  0.53180493
## Temperature -0.051258262 -0.024103084  0.36884151  0.0521903410  0.43153226
## RH          0.085223194  0.062220731 -0.09528038  0.0921514374 -0.30099542
## Wind        0.018797818 -0.020340852 -0.08636797  0.0324781638 -0.02848481
## Rain        0.065387168  0.033234103  0.01343813 -0.0483401530  0.05670153
## Area        0.061994908  0.038838213  0.11428008  0.0002081962  0.04679856
##                     DMC           DC          ISI Temperature          RH
## X          -0.048384178 -0.085916123  0.006209941 -0.05125826  0.08522319
## Y           0.007781561 -0.101177767 -0.024487992 -0.02410308  0.06222073
## Month       0.466645252  0.868697586  0.186596974  0.36884151 -0.09528038
## Day         0.062870397  0.000104902 7  0.032909260  0.05219034  0.09215144
## FFMC        0.382618800  0.330511795 2  0.531804931  0.43153226 -0.30099542
## DMC         1.000000000  0.682191612  0.305127835  0.46959384  0.07379494
## DC          0.682191612  1.000000000  0.229154169  0.49620805 -0.03919165
## ISI         0.305127835  0.229154169  1.000000000  0.39428710 -0.13251718
## Temperature 0.469593844  0.496208053 1 0.394287104  1.00000000 -0.52739034
## RH          0.073794941 -0.039191647 2 -0.132517177 -0.52739034  1.00000000
## Wind       -0.105342253 -0.203465690 9  0.106825888 -0.22711622  0.06941007
## Rain        0.074789982  0.035860862 0  0.067668190  0.06949055  0.09975122
## Area        0.067152740  0.066359756 0 -0.010346879  0.05348655 -0.05366216
##                    Wind        Rain          Area
## X           0.01879782  0.06538717  0.0619949083
## Y          -0.02034085  0.03323410  0.0388382135
## Month      -0.08636797  0.01343813  0.1142800820
## Day         0.03247816 -0.04834015  0.0002081962
## FFMC       -0.02848481  0.05670153  0.0467985637
## DMC        -0.10534225  0.07478998  0.0671527398
## DC         -0.20346569  0.03586086  0.0663597560
## ISI         0.10682589  0.06766819 -0.0103468787
## Temperature -0.22711622  0.06949055  0.0534865490
## RH          0.06941007  0.09975122 -0.0536621583
## Wind        1.00000000  0.06111888  0.0669734893
## Rain        0.06111888  1.00000000  0.0233113127
## Area        0.06697349  0.02331131  1.0000000000
```

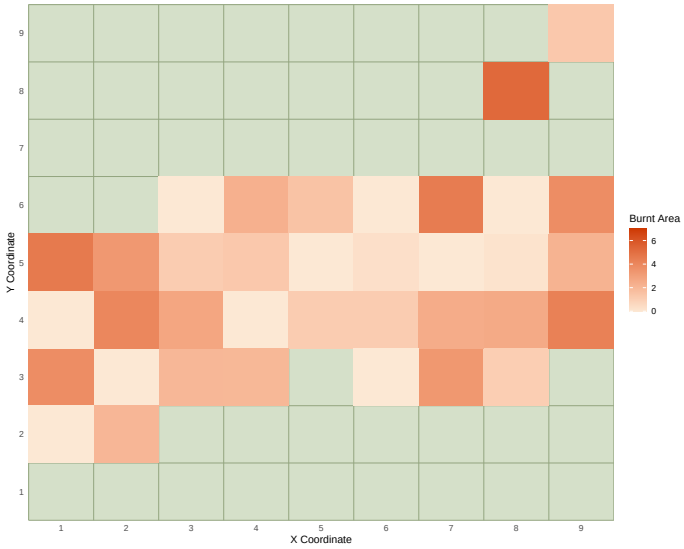
```r
# order correlations with target variable
target_cor <- cor(transformed_data$Area, transformed_data)
# get the sorted indexes to order variable names
sorted_indexes <- order(target_cor, decreasing = TRUE)
# create sorted dataframe with variable names in the sorted order
sorted_cor <- t(data.frame(colnames(target_cor)[sorted_indexes], sort(target_cor, decreasing = TRUE)))
row.names(sorted_cor) <- NULL
sorted_cor
```

```
##      [,1]            [,2]            [,3]            [,4]
## [1,] "Area"          "Month"         "DMC"           "Wind"
## [2,] " 1.0000000000" " 0.1142800820" " 0.0671527398" " 0.0669734893"
##      [,5]            [,6]            [,7]            [,8]
## [1,] "DC"            "X"             "Temperature"   "FFMC"
## [2,] " 0.0663597560" " 0.0619949083" " 0.0534865490" " 0.0467985637"
##      [,9]            [,10]           [,11]           [,12]
## [1,] "Y"             "Rain"          "Day"           "ISI"
## [2,] " 0.0388382135" " 0.0233113127" " 0.0002081962" "-0.0103468787"
##      [,13]
## [1,] "RH"
## [2,] "-0.0536621583"
```

3.5 Relationship between Categorical Variables and Target

3.5.1 Burnt areas by location

```r
heatmap_labels <- c(1:9)
# plot geographical location distribution in relation to burnt area
ggplot(transformed_data, aes(x=X, y=Y, fill=Area)) + theme_bw() + geom_tile() +
  xlab("X Coordinate") + ylab("Y Coordinate") +
  scale_x_continuous(limits = c(0.5,9.5), expand = c(0, 0),
                     breaks = seq_along(heatmap_labels), labels = heatmap_labels) +
  scale_y_continuous(limits = c(0.5,9.5), expand = c(0, 0),
                     breaks = seq_along(heatmap_labels), labels = heatmap_labels) +
  theme(
    panel.background = element_rect(fill = "#d5e0c9"),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_line(color = "#93a57f"),
    axis.ticks = element_blank(),
    panel.border = element_blank()) +
    scale_fill_gradient(low="#fce8d4", high="orangered3", name="Burnt Area")
```




```
##
## Regression tree:
## tree(formula = Area ~ ., data = transformed_data)
## Variables actually used in tree construction:
## [1] "Month"       "Temperature" "DMC"         "DC"
## Number of terminal nodes:  6
## Residual mean deviance:  1.798 = 918.9 / 511
## Distribution of residuals:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -3.6870 -0.8913 -0.7185  0.0000  0.7987  4.9530
```

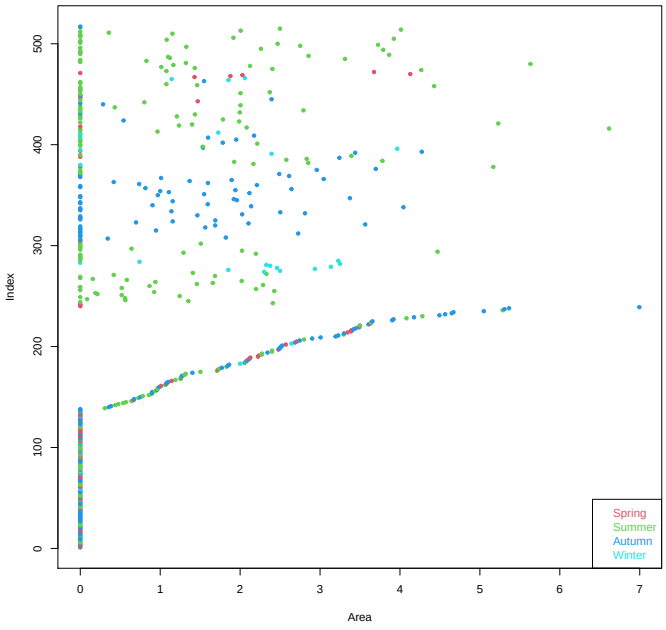
```
print(reg_tree1)
```

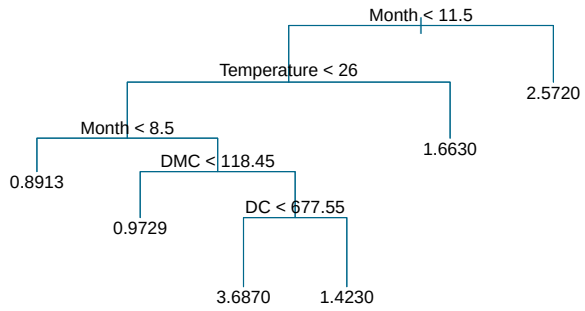
```
## node), split, n, deviance, yval
##       * denotes terminal node
##
##  1) root 517 1009.000 1.1110
##    2) Month < 11.5 508  987.900 1.0850
##      4) Temperature < 26 456  826.600 1.0190
##        8) Month < 8.5 277  434.400 0.8913 *
##        9) Month > 8.5 179  380.600 1.2170
##         18) DMC < 118.45 107  168.000 0.9729 *
##         19) DMC > 118.45 72  196.800 1.5800
##           38) DC < 677.55 5   35.810 3.6870 *
##           39) DC > 677.55 67  137.100 1.4230 *
##      5) Temperature > 26 52  142.000 1.6630 *
##    3) Month > 11.5 9    1.615 2.5720 *
```

```
# plot the tree
plot(reg_tree1, col = colors()[125])
text(reg_tree1, pretty = 0)
```

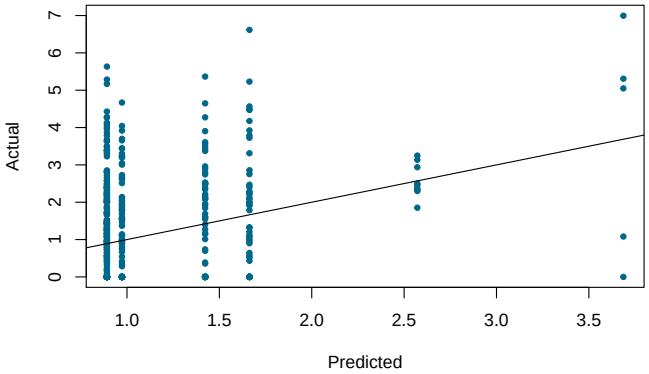
3.5.2 Burnt areas by season

```
# map months to seasons
# spring - March, April, May
# summer - June, July, Aug
# autumn - Sep, Oct, Nov
# winter - Dec, Jan, Feb

month_to_season <- c(4, 4, 1, 1, 1, 2, 2, 2, 3, 3, 3, 4)
fireseason_data <- transformed_data
fireseason_data$Season <- month_to_season[fireseason_data$Month]
plot(fireseason_data$Area, 1:dim(fireseason_data)[1], col = fireseason_data$Season+1,
     type = "p", pch = 20, xlab = "Area", ylab = "Index")
legend( "bottomright", c("Spring", "Summer", "Autumn", "Winter"),
        text.col=c(2,3,4,5) )
```






```
# predict using the model
reg_predict0 = predict(reg_tree1, newdata = transformed_data)
reg_test0 = transformed_data[, 'Area']
plot(reg_predict0, reg_test0, xlab = "Predicted", ylab = "Actual",
     type = "p", pch = 20, col = colors()[125])
abline(0,1)
```



4. Regression Tree Modelling

4.1 Regression Tree for Complete Dataset

```
# tree model for the complete dataset
reg_tree1 <- tree(Area~., data = transformed_data)
summary(reg_tree1)
```

```
# calculate the mean squared error
MSE0 <- mean((reg_predict0-reg_test0)^2)
cat("MSE: ", MSE0, "\t", "Sqrt MSE: ", sqrt(MSE0), "\n", sep = "")
```

```
## MSE: 1.777461    Sqrt MSE: 1.333214
```

4.2 Regression Tree with cross validation

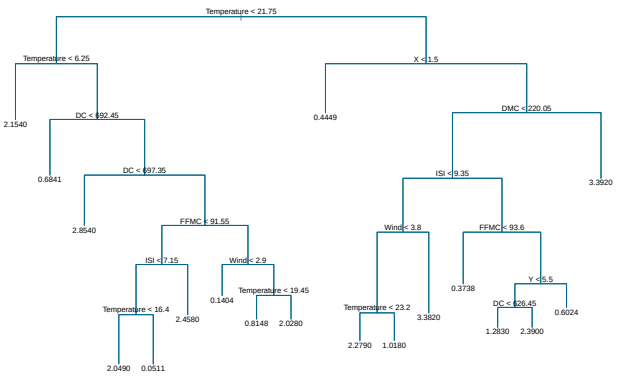
```
# set the seed for sampling
set.seed(2)

# get the train test split
reg_train_split <- sample(1:nrow(transformed_data), nrow(transformed_data)*0.5)

# train the tree model
reg_tree2 <- tree(Area~., data = transformed_data, subset = reg_train_split)
summary(reg_tree2)
```

```
##
## Regression tree:
## tree(formula = Area ~ ., data = transformed_data, subset = reg_train_split)
## Variables actually used in tree construction:
## [1] "Temperature" "DC"          "FFMC"        "ISI"         "Wind"
## [6] "X"           "DMC"         "Y"
## Number of terminal nodes:  18
## Residual mean deviance:  1.306 = 313.4 / 240
## Distribution of residuals:
##     Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -3.3920 -0.6841 -0.3002  0.0000  0.6339  3.6130
```

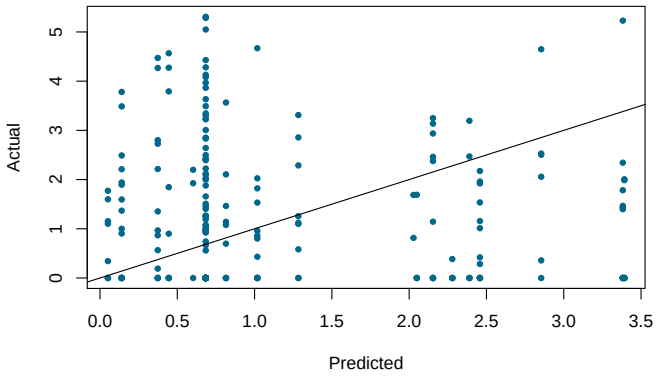
```
print(reg_tree2)
```

```
## node), split, n, deviance, yval
##       * denotes terminal node
##
##   1) root 258 516.0000 1.1300
##     2) Temperature < 21.75 176 280.7000 0.9806
##       4) Temperature < 6.25 10   2.9750 2.1540 *
##       5) Temperature > 6.25 166 263.1000 0.9099
##        10) DC < 692.45 105 117.1000 0.6841 *
##        11) DC > 692.45 61 131.4000 1.2990
##          22) DC < 697.35 5  16.1800 2.8540 *
##          23) DC > 697.35 56 102.1000 1.1600
##            46) FFMC < 91.55 24  52.5500 1.6540
##              92) ISI < 7.15 13  22.9800 0.9734
##               184) Temperature < 16.4 6   9.9670 2.0490 *
##               185) Temperature > 16.4 7   0.1097 0.0511 *
##              93) ISI > 7.15 11  16.4500 2.4580 *
##            47) FFMC > 91.55 32  39.2900 0.7895
##              94) Wind < 2.9 12   1.2050 0.1404 *
##              95) Wind > 2.9 20  29.9900 1.1790
##               190) Temperature < 19.45 14  16.5500 0.8148 *
##               191) Temperature > 19.45 6   7.2620 2.0280 *
##     3) Temperature > 21.75 82 223.0000 1.4500
##       6) X < 1.5 11   5.9680 0.4449 *
##       7) X > 1.5 71 204.2000 1.6050
##        14) DMC < 220.05 66 159.4000 1.4700
##          28) ISI < 9.35 28  96.5600 2.0090
##            56) Wind < 3.8 20  32.6000 1.4590
##              112) Temperature < 23.2 7   5.5000 2.2790 *
##              113) Temperature > 23.2 13  19.8600 1.0180 *
##            57) Wind > 3.8 8  42.8300 3.3820 *
##          29) ISI > 9.35 38  48.7100 1.0730
##            58) FFMC < 93.6 16   6.8250 0.3738 *
##            59) FFMC > 93.6 22  28.3800 1.5810
##              118) Y < 5.5 17  18.3200 1.8690
##                236) DC < 626.45 8   5.7210 1.2830 *
##                237) DC > 626.45 9   7.4100 2.3900 *
##              119) Y > 5.5 5   3.8610 0.6024 *
##        15) DMC > 220.05 5  27.6400 3.3920 *
```

```
# plot the trained tree
plot(reg_tree2, col = colors()[125])
text(reg_tree2, pretty = 0)
```




```
# predict using the model
reg_predict = predict(reg_tree2, newdata = transformed_data[-reg_train_split,])
reg_test = transformed_data[-reg_train_split, 'Area']
plot(reg_predict, reg_test, xlab = "Predicted", ylab = "Actual",
     type = "p", pch = 20, col = colors()[125])
abline(0,1)
```



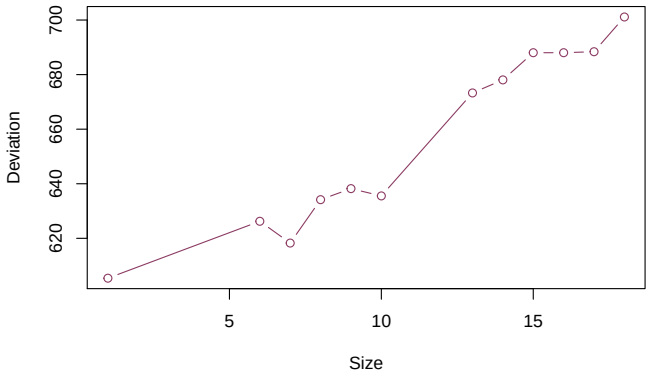

```
# calculate the mean squared error
MSE <- mean((reg_predict-reg_test)^2)
cat("MSE: ", MSE, "\t", "Sqrt MSE: ", sqrt(MSE), "\n", sep = "")
```

```
## MSE: 2.791228    Sqrt MSE: 1.670697
```

```
# train model using cross validation
reg_cv <- cv.tree(reg_tree2)
reg_cv
```

```
## $size
##  [1] 18 17 16 15 14 13 10  9  8  7  6  1
##
## $dev
##  [1] 701.1168 688.3910 688.0291 688.0291 678.0657 673.2833 635.5436 638.2141
##  [9] 634.1649 618.2709 626.2499 605.3920
##
## $k
##  [1]      -Inf  5.185608  6.185315  6.200202  7.238985  8.088079 12.091694
##  [8] 13.176198 13.506280 14.578840 14.600544 15.511457
##
## $method
## [1] "deviance"
##
## attr(,"class")
## [1] "prune"         "tree.sequence"
```

```
# plot the cross validation results, deviation explained by the model against the size of the tree
plot(reg_cv$size, reg_cv$dev, type = "b", col = colors()[371],
     xlab = "Size", ylab = "Deviation")
```



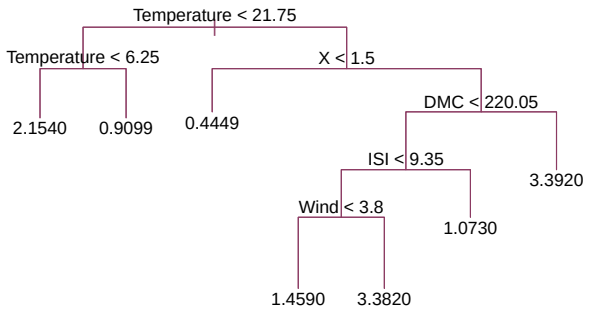
4.3 Pruned Regression Tree

```
# prune tree based on the cv results
reg_pruned_tree = prune.tree(reg_tree2, best = 7)
summary(reg_pruned_tree)
```

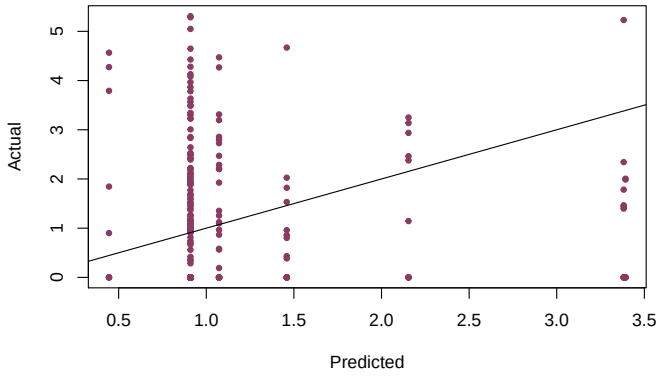
```
##
## Regression tree:
## snip.tree(tree = reg_tree2, nodes = c(56L, 29L, 5L))
## Variables actually used in tree construction:
```

```
## [1] "Temperature" "X"           "DMC"         "ISI"         "Wind"
## Number of terminal nodes:  7
## Residual mean deviance:  1.688 = 423.8 / 251
## Distribution of residuals:
##     Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -3.3920 -0.9099 -0.4734  0.0000  0.8257  4.4550
```

```
# plot the pruned tree
plot(reg_pruned_tree, col = colors()[371])
text(reg_pruned_tree, pretty = 0)
```




```
# predict using the pruned tree
reg_predict2 = predict(reg_pruned_tree, newdata = transformed_data[-reg_train_split,])
reg_test2 = transformed_data[-reg_train_split, 'Area']
plot(reg_predict2, reg_test2, xlab = "Predicted", ylab = "Actual",
     type = "p", pch = 20, col = colors()[371])
abline(0,1)
```




```
# calculate the mean squared error
MSE2 <- mean((reg_predict2-reg_test2)^2)
cat("MSE: ", MSE2, "\t", "Sqrt MSE: ", sqrt(MSE2), "\n", sep = "")
```

```
## MSE: 2.363859    Sqrt MSE: 1.537485
```

5. Multiple Linear Regression

```
# filter only the numerical variables for regression
numerical_data <- transformed_data[5:13]

# normalise dataset
normalise <- function(x){
  return((x-min(x))/(max(x)-min(x)))
}
fire_norm <- as.data.frame(lapply(numerical_data, normalise))
head(fire_norm)
```

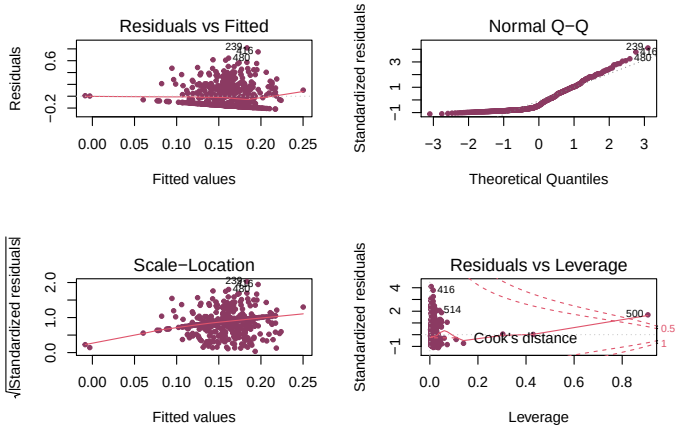
```
##        FFMC        DMC         DC        ISI Temperature        RH       Wind
## 1 0.8709677 0.08649207 0.10132520 0.09090909   0.1929260 0.4235294 0.70000000
## 2 0.9277419 0.11819435 0.77541926 0.11942959   0.5080386 0.2117647 0.05555556
## 3 0.9277419 0.14679531 0.79629412 0.11942959   0.3987138 0.2117647 0.10000000
## 4 0.9419355 0.11095796 0.08162308 0.16042781   0.1961415 0.9647059 0.40000000
## 5 0.9109677 0.17298415 0.11058989 0.17112299   0.2958199 0.9882353 0.15555556
## 6 0.9496774 0.29014473 0.56303507 0.26203209   0.6430868 0.1647059 0.55555556
##      Rain Area
## 1 0.00000    0
## 2 0.00000    0
## 3 0.00000    0
## 4 0.03125    0
## 5 0.00000    0
## 6 0.00000    0
```

5.1 Model 1 - Using all Numerical Variables

```
# linear model using all the numerical variables
lin_model1 <- lm(Area ~ ., data = fire_norm)
summary(lin_model1)
```

```
##
## Call:
## lm(formula = Area ~ ., data = fire_norm)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.21733 -0.15908 -0.08803  0.12561  0.81653
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.04691    0.15062   0.311    0.756
## FFMC         0.08539    0.16051   0.532    0.595
## DMC          0.04943    0.06074   0.814    0.416
## DC           0.03336    0.04351   0.767    0.444
## ISI         -0.19206    0.13573  -1.415    0.158
## Temperature  0.01094    0.07673   0.143    0.887
## RH          -0.06285    0.06305  -0.997    0.319
## Wind         0.09748    0.04711   2.069    0.039 *
## Rain         0.08829    0.19408   0.455    0.649
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1995 on 508 degrees of freedom
## Multiple R-squared:  0.01988,    Adjusted R-squared:  0.004446
## F-statistic: 1.288 on 8 and 508 DF,  p-value: 0.2472
```

```
par(mfrow=c(2,2))
plot(lin_model1, pch = 20, col = colors()[371])
```



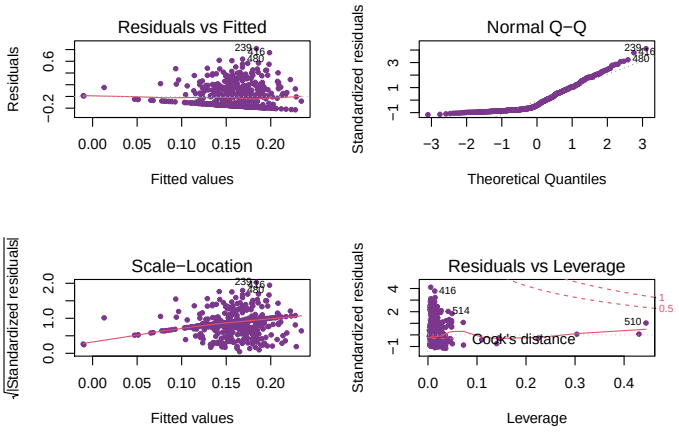
Model 1 Improvement - Excluding Influential Data Points

```
# linear model without influential data points
# removing point 500
fire_norm1 <- fire_norm[-c(500),]
lin_model1a <- lm(Area ~ ., data = fire_norm1)
summary(lin_model1a)
```

```
##
## Call:
## lm(formula = Area ~ ., data = fire_norm1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.22763 -0.15751 -0.08224  0.12757  0.81617
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.01901    0.15125   0.126   0.9000
## FFMC         0.09926    0.16042   0.619   0.5364
## DMC          0.04854    0.06063   0.801   0.4237
## DC           0.03301    0.04343   0.760   0.4476
## ISI         -0.19780    0.13552  -1.460   0.1450
```

```
## Temperature  0.02400    0.07697   0.312   0.7554
## RH          -0.04411    0.06390  -0.690   0.4903
## Wind         0.10827    0.04745   2.282   0.0229 *
## Rain        -0.92471    0.62872  -1.471   0.1420
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1991 on 507 degrees of freedom
## Multiple R-squared:  0.02361,    Adjusted R-squared:  0.008199
## F-statistic: 1.532 on 8 and 507 DF,  p-value: 0.1432
```

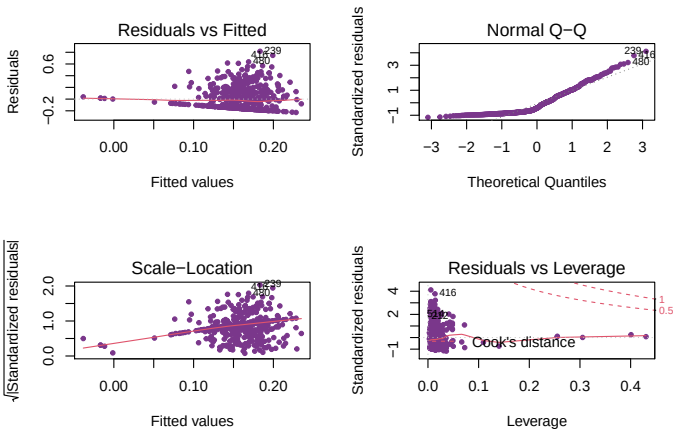
```
par(mfrow=c(2,2))
plot(lin_model1a, pch = 20, col = colors()[466])
```




```
# removing point 510
fire_norm <- fire_norm[-c(500, 510),]
lin_model1b <- lm(Area ~ ., data = fire_norm)
summary(lin_model1b)
```

```
##
## Call:
## lm(formula = Area ~ ., data = fire_norm)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.22845 -0.15751 -0.08198  0.12630  0.81668
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.01460    0.15130   0.097   0.9231
## FFMC         0.10227    0.16044   0.637   0.5241
## DMC          0.04938    0.06063   0.814   0.4158
## DC           0.03136    0.04346   0.722   0.4708
## ISI         -0.18888    0.13579  -1.391   0.1648
## Temperature  0.02464    0.07697   0.320   0.7490
## RH          -0.04059    0.06399  -0.634   0.5261
## Wind         0.10736    0.04746   2.262   0.0241 *
## Rain        -1.48348    0.83242  -1.782   0.0753 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1991 on 506 degrees of freedom
## Multiple R-squared:  0.02562,    Adjusted R-squared:  0.01022
## F-statistic: 1.663 on 8 and 506 DF,  p-value: 0.1048
```

```
par(mfrow=c(2,2))
plot(lin_model1b, pch = 20, col = colors()[466])
```



5.2 Model 2 - Using step-wise Feature Selection

```
# perform step-wise feature selection
step(lm(Area~., data = fire_norm), direction = 'both')
```

```
## Start:  AIC=-1653.54
## Area ~ FFMC + DMC + DC + ISI + Temperature + RH + Wind + Rain
##
##               Df Sum of Sq    RSS     AIC
## - Temperature  1  0.004061 20.059 -1655.4
## - RH           1  0.015948 20.070 -1655.1
## - FFMC         1  0.016103 20.071 -1655.1
## - DC           1  0.020641 20.075 -1655.0
## - DMC          1  0.026283 20.081 -1654.9
## - ISI          1  0.076681 20.131 -1653.6
## <none>                     20.055 -1653.5
## - Rain         1  0.125876 20.180 -1652.3
## - Wind         1  0.202828 20.257 -1650.4
##
## Step:  AIC=-1655.44
## Area ~ FFMC + DMC + DC + ISI + RH + Wind + Rain
##
##               Df Sum of Sq    RSS     AIC
## - FFMC         1  0.015512 20.074 -1657.0
## - DC           1  0.026444 20.085 -1656.8
## - DMC          1  0.036661 20.095 -1656.5
## - RH           1  0.044533 20.103 -1656.3
## - ISI          1  0.072807 20.131 -1655.6
## <none>                     20.059 -1655.4
## - Rain         1  0.122435 20.181 -1654.3
## + Temperature  1  0.004061 20.055 -1653.5
## - Wind         1  0.200690 20.259 -1652.3
##
## Step:  AIC=-1657.04
## Area ~ DMC + DC + ISI + RH + Wind + Rain
##
##               Df Sum of Sq    RSS     AIC
## - DC           1  0.029078 20.103 -1658.3
## - DMC          1  0.048869 20.123 -1657.8
## - ISI          1  0.057331 20.131 -1657.6
## - RH           1  0.069904 20.144 -1657.2
## <none>                     20.074 -1657.0
## - Rain         1  0.117912 20.192 -1656.0
## + FFMC         1  0.015512 20.059 -1655.4
## + Temperature  1  0.003471 20.071 -1655.1
## - Wind         1  0.197639 20.272 -1654.0
##
## Step:  AIC=-1658.29
## Area ~ DMC + ISI + RH + Wind + Rain
##
##               Df Sum of Sq    RSS     AIC
## - ISI          1  0.054068 20.157 -1658.9
## <none>                     20.103 -1658.3
## - RH           1  0.080173 20.183 -1658.2
## - Rain         1  0.118826 20.222 -1657.3
## + DC           1  0.029078 20.074 -1657.0
## + FFMC         1  0.018146 20.085 -1656.8
## + Temperature  1  0.009245 20.094 -1656.5
## - Wind         1  0.177112 20.280 -1655.8
## - DMC          1  0.193437 20.297 -1655.4
##
## Step:  AIC=-1658.91
## Area ~ DMC + RH + Wind + Rain
##
##               Df Sum of Sq    RSS     AIC
## - RH           1  0.059457 20.217 -1659.4
## <none>                     20.157 -1658.9
## + ISI          1  0.054068 20.103 -1658.3
## + DC           1  0.025814 20.131 -1657.6
## - Rain         1  0.134415 20.292 -1657.5
## - DMC          1  0.148047 20.305 -1657.1
## - Wind         1  0.151671 20.309 -1657.0
## + Temperature  1  0.000586 20.157 -1656.9
## + FFMC         1  0.000315 20.157 -1656.9
##
## Step:  AIC=-1659.39
## Area ~ DMC + Wind + Rain
##
##               Df Sum of Sq    RSS     AIC
## <none>                     20.217 -1659.4
## + RH           1  0.059457 20.157 -1658.9
## + DC           1  0.034871 20.182 -1658.3
## + ISI          1  0.033352 20.183 -1658.2
## + Temperature  1  0.031382 20.185 -1658.2
## - DMC          1  0.137138 20.354 -1657.9
## - Wind         1  0.141748 20.358 -1657.8
## + FFMC         1  0.011361 20.205 -1657.7
## - Rain         1  0.165236 20.382 -1657.2
```

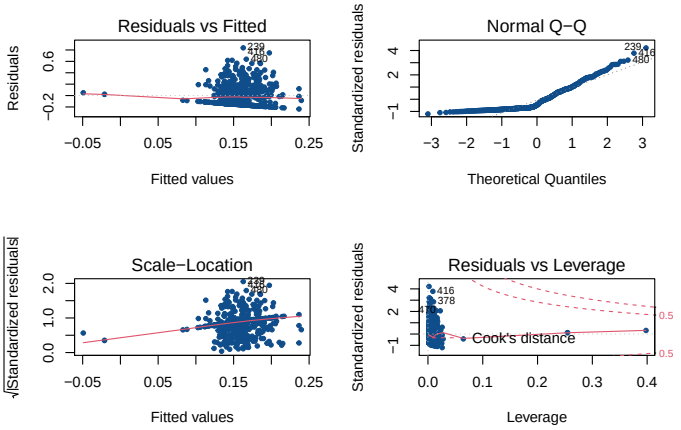
```
##
## Call:
## lm(formula = Area ~ DMC + Wind + Rain, data = fire_norm)
##
## Coefficients:
## (Intercept)          DMC         Wind         Rain
##     0.09797      0.07485      0.08459     -1.66210
```

```
# linear model using variables selected from step-wise selection
lin_model2 <- lm(Area ~ DMC+Rain+Wind, data = fire_norm)
summary(lin_model2)
```

```
##
## Call:
## lm(formula = Area ~ DMC + Rain + Wind, data = fire_norm)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.23663 -0.15348 -0.08883  0.12513  0.83724
```

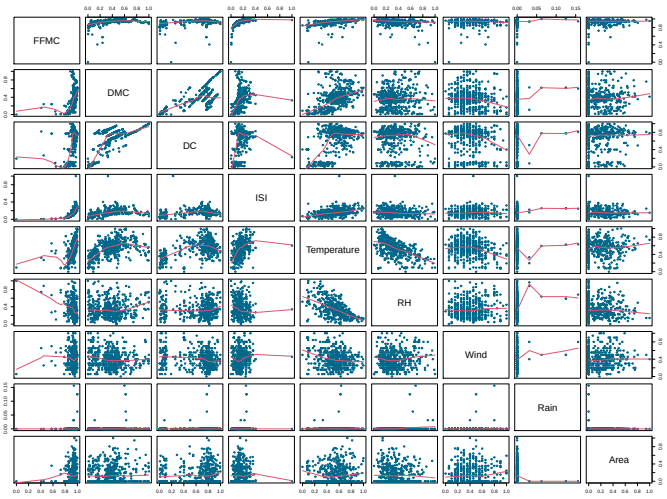
```
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.09797    0.02624   3.734  0.00021 ***
## DMC          0.07485    0.04020   1.862  0.06320 .
## Rain        -1.66210    0.81330  -2.044  0.04150 *
## Wind         0.08459    0.04469   1.893  0.05894 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1989 on 511 degrees of freedom
## Multiple R-squared:  0.01774,    Adjusted R-squared:  0.01198
## F-statistic: 3.077 on 3 and 511 DF,  p-value: 0.02727
```

```
par(mfrow=c(2,2))
plot(lin_model2, pch = 20, col = colors()[132])
```



6. Non-linear Models

```
# pairplot
pairs(fire_norm, panel = panel.smooth, pch = 20, col = colors()[125])
```

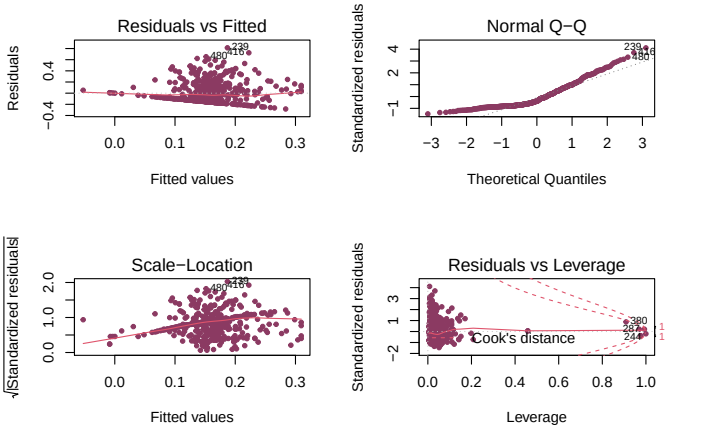


6.1 Model 1 - Using Polynomial and Interaction Terms

```
# complex model
nolin_model1 <- lm(Area ~ .+ I(Temperature^2) + I(ISI^2) + I(FFMC^2) + I(DMC^2) + I(Wind^2)
                   + DC*ISI + RH*Wind*Rain + Temperature*Wind + I(ISI^2)*Wind + FFMC*ISI
                   + Temperature*DMC -ISI -RH, data = fire_norm)
summary(nolin_model1)
```

```
##
## Call:
## lm(formula = Area ~ . + I(Temperature^2) + I(ISI^2) + I(FFMC^2) +
##     I(DMC^2) + I(Wind^2) + DC * ISI + RH * Wind * Rain + Temperature *
##     Wind + I(ISI^2) * Wind + FFMC * ISI + Temperature * DMC -
##     ISI - RH, data = fire_norm)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.28372 -0.15331 -0.07441  0.11033  0.81299
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       -0.04593    0.19456  -0.236   0.8135
## FFMC               0.35580    0.60780   0.585   0.5586
## DMC                0.04096    0.27361   0.150   0.8811
## DC                 0.07331    0.09869   0.743   0.4579
## Temperature       -0.37352    0.29746  -1.256   0.2098
## Wind               0.26432    0.25523   1.036   0.3009
## Rain             -13.64798   36.06037  -0.378   0.7052
## I(Temperature^2)   0.44790    0.24549   1.825   0.0687 .
## I(ISI^2)          -1.24048    1.05953  -1.171   0.2422
## I(FFMC^2)         -0.16472    0.50863  -0.324   0.7462
## I(DMC^2)          -0.12495    0.19214  -0.650   0.5158
## I(Wind^2)         -0.05290    0.19427  -0.272   0.7855
## DC:ISI            -0.16713    0.59420  -0.281   0.7786
## RH:Wind           -0.02011    0.12447  -0.162   0.8717
## RH:Rain           13.11708   52.75882   0.249   0.8038
## Wind:Rain         38.70435   70.21708   0.551   0.5817
## Temperature:Wind  -0.39978    0.30381  -1.316   0.1888
## Wind:I(ISI^2)      2.58394    2.03470   1.270   0.2047
## FFMC:ISI          -0.03150    0.68553  -0.046   0.9634
## DMC:Temperature    0.24277    0.30800   0.788   0.4310
## RH:Wind:Rain     -50.35345  100.88278  -0.499   0.6179
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1986 on 494 degrees of freedom
## Multiple R-squared:  0.05345,    Adjusted R-squared:  0.01513
## F-statistic: 1.395 on 20 and 494 DF,  p-value: 0.1185
```

```
par(mfrow=c(2,2))
plot(nolin_model1, pch = 20, col = colors()[371])
```



6.2 Model 2 - Using Most Significant Terms

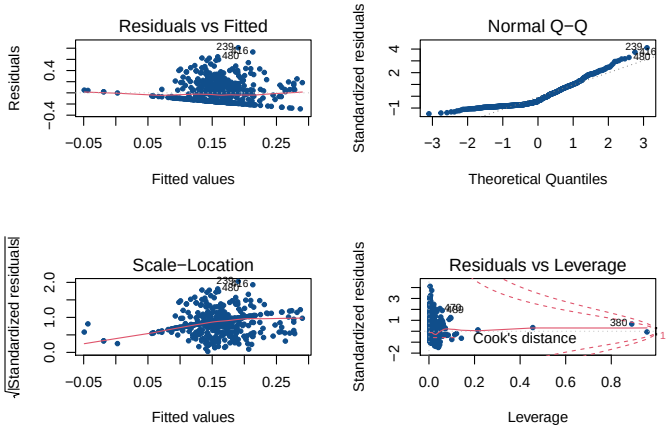
```
# simplified model
nolin_model2 <- lm(Area ~ .+ I(Temperature^2) + I(FFMC^2) + I(Wind^2)
                   + DC*ISI + RH*Rain + I(ISI^2)*Wind
                   -ISI -RH -DMC -Rain -Wind, data = fire_norm)
summary(nolin_model2)
```

```
##
## Call:
## lm(formula = Area ~ . + I(Temperature^2) + I(FFMC^2) + I(Wind^2) +
##     DC * ISI + RH * Rain + I(ISI^2) * Wind - ISI - RH - DMC -
##     Rain - Wind, data = fire_norm)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.28714 -0.15199 -0.07169  0.11903  0.80952
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       -0.01017    0.18809  -0.054 0.956899
## FFMC               0.42944    0.53750   0.799 0.424699
## DC                 0.10483    0.05235   2.002 0.045782 *
## Temperature       -0.61668    0.21354  -2.888 0.004045 **
## I(Temperature^2)   0.63916    0.19150   3.338 0.000907 ***
## I(FFMC^2)         -0.17825    0.41131  -0.433 0.664921
## I(Wind^2)          0.06772    0.07173   0.944 0.345545
## I(ISI^2)          -0.60453    0.81226  -0.744 0.457072
## DC:ISI            -0.23758    0.28241  -0.841 0.400599
## RH:Rain           -2.35624    1.26539  -1.862 0.063176 .
## Wind:I(ISI^2)      1.14818    1.69804   0.676 0.499235
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1974 on 504 degrees of freedom
## Multiple R-squared:  0.04602,    Adjusted R-squared:  0.02709
## F-statistic: 2.431 on 10 and 504 DF,  p-value: 0.007784
```

```
# calculate MSE
mean(residuals(nolin_model2)^2)
```

```
## [1] 0.03812565
```

```
par(mfrow=c(2,2))
plot(nolin_model2, pch = 20, col = colors()[132])
```

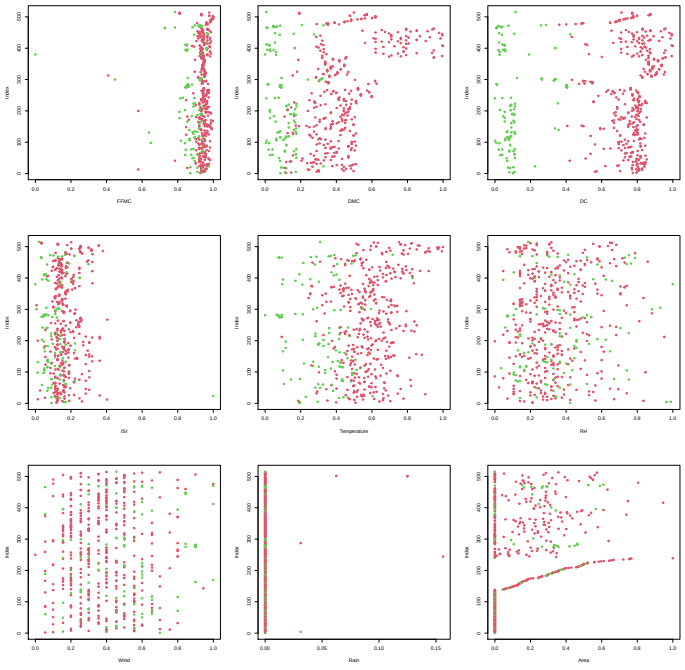


8. Clustering

8.1 K-means Clustering

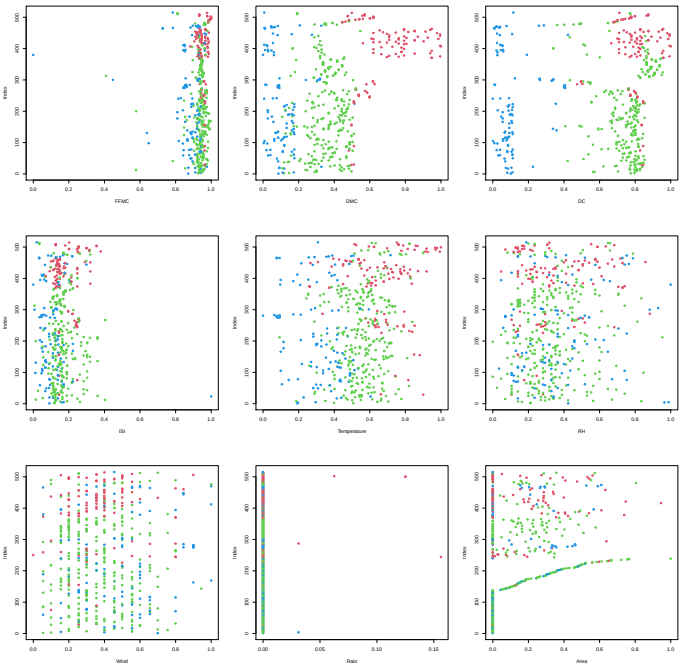
```
# kmeans clustering for k=2
km_model <- kmeans(fire_norm[,1:8], centers = 2)
km_clusters <- fitted(km_model, "classes")
```

```
par(mfrow=c(3,3))
i <- 1
for (col in fire_norm) {
  plot(col, 1:dim(fire_norm)[1],  col = km_clusters+1, type = "p",
       pch = 20, xlab = colnames(fire_norm)[i], ylab = "Index")
  i <- i + 1
}
```



```
# kmeans clustering for k=3
km_model <- kmeans(fire_norm[,1:8], centers = 3)
km_clusters <- fitted(km_model, "classes")
```

```
par(mfrow=c(3,3))
i <- 1
for (col in fire_norm) {
  plot(col, 1:dim(fire_norm)[1],  col = km_clusters+1, type = "p",
       pch = 20, xlab = colnames(fire_norm)[i], ylab = "Index")
  i <- i + 1
}
```



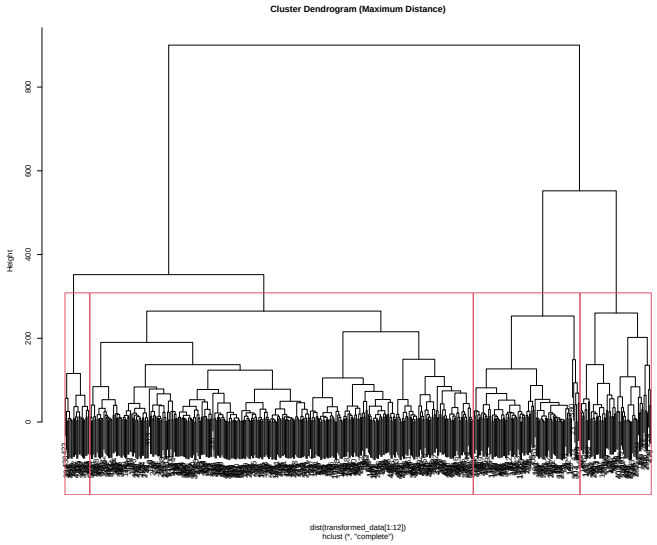
8.2. Hierarchical Clustering

8.2.1 Using Complete Linkage (Maximum distance)

```
hc_complete = hclust(dist(transformed_data[1:12]), method = "complete")
h_clusters_c = cutree(hc_complete, k=4)
table(h_clusters_c)
```

```
## h_clusters_c
##   1   2   3   4
##  94 338  63  22
```

```
plot(hc_complete, main = "Cluster Dendrogram (Maximum Distance)")
rect.hclust(hc_complete, k=4)
```

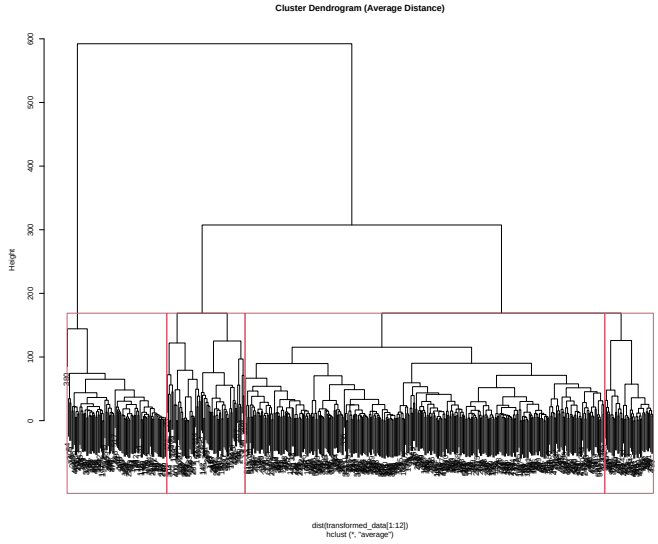


8.2.2 Using Average Linkage (Average Distance)

```
hc_average = hclust(dist(transformed_data[1:12]), method = "average")
h_clusters_a = cutree(hc_average, k=4)
table(h_clusters_a)
```

```
## h_clusters_a
##   1   2   3   4
##  88 317  69  43
```

```
plot(hc_average, main = "Cluster Dendrogram (Average Distance)")
rect.hclust(hc_average, k=4)
```

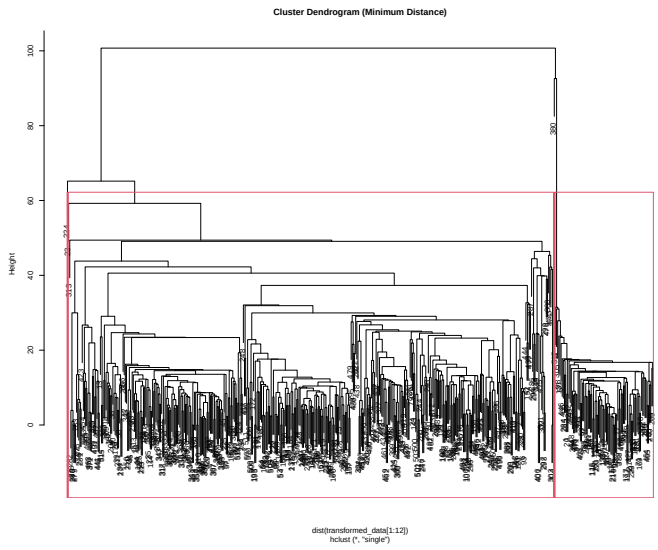


8.2.3 Using Single Linkage (Minimum Distance)

```
hc_simple = hclust(dist(transformed_data[1:12]), method = "single")
h_clusters_s = cutree(hc_simple, k=4)
table(h_clusters_s)
```

```
## h_clusters_s
##   1   2   3   4
##  87 428   1   1
```

```
plot(hc_simple, main = "Cluster Dendrogram (Minimum Distance)")
rect.hclust(hc_simple, k=4)
```



9. Model Comparison and Evaluation

```
# split into train and test sets
# set seed for sampling
set.seed(2)
dim(fire_norm)
```

```
## [1] 515   9
```

```
fire_norm <- fire_norm[-c(500, 510),]
# divide the training and testing datasets to 50:50 ratio
reg_split <- sample(1:nrow(fire_norm), nrow(fire_norm)*0.5)
reg_train <- fire_norm[reg_split,]
reg_test <- fire_norm[-reg_split,]

dim(reg_train)
```

```
## [1] 256   9
```

```
dim(reg_test)
```

```
## [1] 257   9
```

9.1 Cross Validation for Linear Models

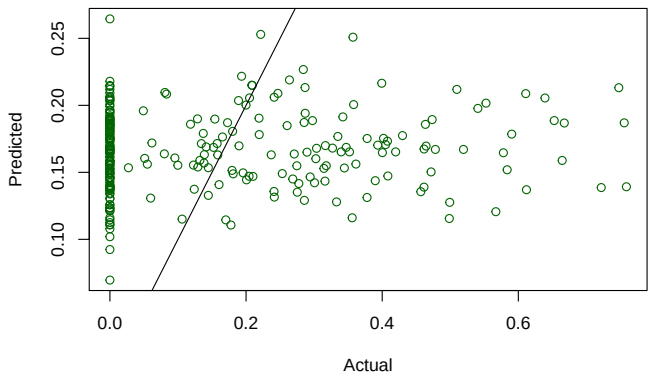
```
# Model 1
m1 <- glm(Area~., data = reg_train)
cv1 <- cv.glm(reg_train, m1, K=10)
cv1$delta
```

```
## [1] 0.04288920 0.04274935
```

```
pred1 <- predict(m1, reg_test)
mse1 <- mean((fire_norm$Area[-reg_split]-pred1)^2)
mse1
```

```
## [1] 0.0384408
```

```
plot(fire_norm$Area[-reg_split], pred1, xlab = "Actual", ylab = "Predicted",
     col = colors()[81])
abline(0,1)
```



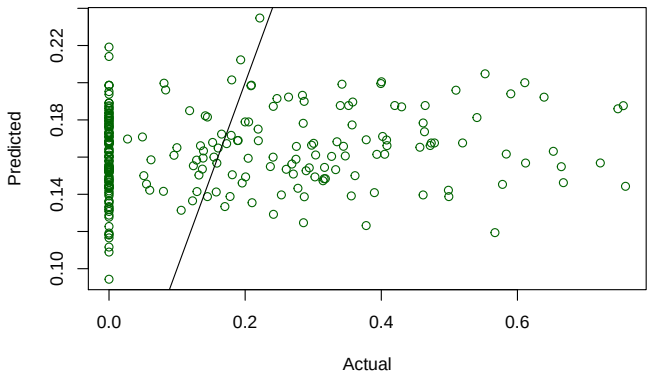

```
# Model 2
m2 <- glm(Area~DMC+Rain+Wind, data = reg_train)
cv2 <- cv.glm(reg_train, m2, K=10)
cv2$delta
```

```
## [1] 0.04162815 0.04158624
```

```
pred2 <- predict(m2, reg_test)
mse2 <- mean((fire_norm$Area[-reg_split]-pred2)^2)
mse2
```

```
## [1] 0.03794638
```

```
plot(fire_norm$Area[-reg_split], pred2, xlab = "Actual", ylab = "Predicted",
     col = colors()[81])
abline(0,1)
```



9.2 Cross Validation for Non-Linear Models

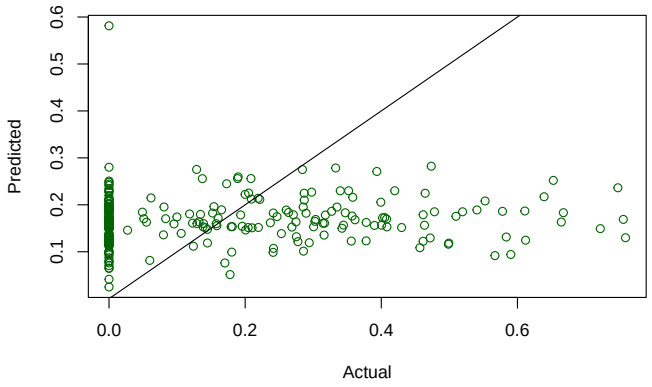
```
# Model 3
m3 <- glm(Area~.+ I(Temperature^2) + I(ISI^2) + I(FFMC^2) + I(DMC^2) + I(Wind^2)
                  + DC*ISI + RH*Wind*Rain + Temperature*Wind + I(ISI^2)*Wind + FFMC*ISI
                  + Temperature*DMC -ISI -RH, data = reg_train)
cv3 <- cv.glm(reg_train, m3, K=10)
cv3$delta
```

```
## [1] 0.04632050 0.04587198
```

```
pred3 <- predict(m3, reg_test)
mse3 <- mean((fire_norm$Area[-reg_split]-pred3)^2)
mse3
```

```
## [1] 0.03941836
```

```
plot(fire_norm$Area[-reg_split], pred3, xlab = "Actual", ylab = "Predicted",
     col = colors()[81])
abline(0,1)
```



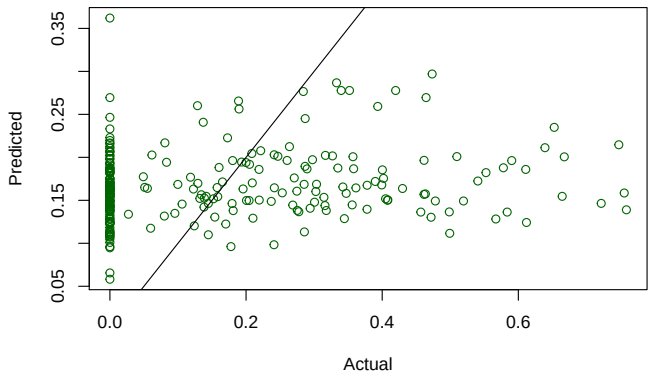

```
# Model 4
m4 <- glm(Area~.+ I(Temperature^2) + I(FFMC^2) + I(Wind^2)
                  + DC*ISI + RH*Rain + I(ISI^2)*Wind
                  -ISI -RH -DMC -Rain -Wind, data = reg_train)
cv4 <- cv.glm(reg_train, m4, K=10)
cv4$delta
```

```
## [1] 0.04292929 0.04275025
```

```
pred4 <- predict(m4, reg_test)
mse4 <- mean((fire_norm$Area[-reg_split]-pred4)^2)
mse4
```

```
## [1] 0.03754266
```

```
plot(fire_norm$Area[-reg_split], pred4, xlab = "Actual", ylab = "Predicted",
     col = colors()[81])
abline(0,1)
```



9.3 Histogram of residuals

```
hist(m4$residuals, main = "Distribution of Residuals",
     xlab = "Residuals", col = colors()[106])
```