Prediction of Area under Forest Fire Using Two-Stage Classification-Regression Model

Submitted by: Amishi, Vanshika Tiwari, Yusuf Ahmed
Course and College: M.Sc. Statistics (II), Department of Statistics, University of Delhi

Submission for: ISPS M.Sc. Student Project Competition 2024

Abstract

Forest fires are a significant environmental hazard, causing widespread damage to ecosystems, wildlife, human life, and property. They contribute to severe environmental degradation, including deforestation, biodiversity loss, and air pollution. Accurate prediction of forest fires is essential for safeguarding human lives, minimizing economic losses, protecting ecosystems, and ensuring the efficient allocation of firefighting resources. In the context of rising global temperatures and more frequent wildfires, the need for precise fire prediction is more urgent than ever. In this project, we explored various regression and classification techniques to predict the extent of forest fires in Montesinho Natural Park. Our best-performing model is a **two-stage approach combining a Random Forest Classifier and a Random Forest Regressor**. The classifier first predicts whether the area burnt under forest fire is 0 hectares or more (binary output). If the burnt area of 0 hectares is predicted, the regression model outputs zero for the burned area. If a burnt area of more than 0 hectares is predicted, the regression model estimates the area affected. The final model achieved a **Root Mean Squared Error (RMSE)** of **2.5436 hectares**, **R²** of **0.648**, and an **adjusted R²** of **0.629**. This model can be further refined by incorporating additional physical factors and constraints for improved accuracy.

Keywords: Forest fires, Regression, Classification, Random Forest

Introduction

Forest fires are among the most destructive natural disasters, causing widespread ecological damage, releasing harmful emissions into the atmosphere, and posing significant risks to wildlife and human populations. These fires' unpredictability and rapid spread make early prediction essential for effective disaster prevention and resource management. Accurately identifying fire-prone areas allows for timely mitigation efforts, helping to protect ecosystems and minimize loss of life and property.

The global forest fire situation is becoming more severe, driven by the twin forces of climate change and human activity. Wildfires are not only a growing environmental and economic threat but also a public health crisis. Tackling this problem requires a coordinated global effort, with a focus on climate action, sustainable land management, and better fire prediction and response systems. Forest fires are one of the major causes of forest degradation in India. It is reported that almost 90% of fires are caused by manmade factors.

In this project, we aim to predict the area affected by forest fires using machine learning techniques. Machine learning plays a crucial role in this domain because it processes large volumes of environmental data, such as temperature, wind speed, humidity, and rainfall. These factors, often complex and non-linear in their relationship to fire outbreaks, require advanced techniques to detect patterns and make reliable predictions.

We have used Random Forest, an ensemble learning algorithm in this project. It is particularly well-suited for this task because of its ability to handle complex and non-linear data. Unlike simpler models, Random Forest constructs multiple decision trees during training, ensuring a more robust prediction by averaging the results of individual trees. This method allows the model to capture intricate relationships between environmental factors and fire behaviour, leading to higher accuracy and generalization when predicting fire-affected areas. **Kumar and Babu (2015)** also focused on predicting forest fire risk using machine learning techniques. It compared various algorithms like Random Forest and Support Vector Machines, concluding that machine learning provides robust solutions for complex environmental datasets.

The dataset used in this project is based on historical forest fire records, which are analysed to build and validate the machine learning models. By improving the precision of fire prediction models, this project contributes to the ongoing efforts to better manage forest fire risks and deploy resources more efficiently, thus minimizing the devastating impact of forest fires on both the environment and society. **Cortez and Morais (2007)** used the same dataset to explore forest fire prediction using data mining techniques such as neural networks and support vector machines. Their study highlighted the importance of accurate fire prediction models for forest management and disaster prevention.

Material and Methods

Data Description

Source: The "Forest Fires" dataset was collected from the Montesinho Natural Park in the Trás-os-Montes region of Portugal. The data was recorded between January 2000 and December 2003 (Cortez, P. & Morais, A. (2007). Forest Fires [Dataset]. UCI Machine Learning Repository. https://doi.org/10.24432/C5D88D.)

Overview: This dataset contains 517 observations with 13 attributes, recording forest fire incidents and associated weather conditions. These attributes are categorized into spatial, temporal, and meteorological features, enabling comprehensive fire behaviour analysis across different environmental factors.

Variables Description:

1. **Spatial Variables**:
   - *X-coordinate* and *Y-coordinate*: These represent spatial positions within Montesinho Natural Park. The total area of 74225 hectares (ha) was divided into 9×9 grid, each square representing 916 ha. The X-axis ranges from 1 to 9 and the Y-axis from 1 to 9. These coordinates identify specific locations in the park for each fire incident, allowing for geographic analysis of fire distribution.

2. **Temporal Variables**:
   - *Month*: Indicates the month of the year, recorded as categorical values from 'jan' to 'dec'. This allows for seasonal trend analysis of fire occurrences.
   - *Day*: Specifies the day of the week from 'mon' to 'sun'. This can be used to observe daily patterns in fire occurrences.

3. **Fire Weather Index (FWI) System Variables**:
   - **FFMC (Fine Fuel Moisture Code)**: A scale from 18.7 to 96.2, measuring the ease with which small, dry fuels ignite and burn. A higher FFMC value indicates greater flammability.
   - **DMC (Duff Moisture Code)**: Ranges from 1.1 to 291.3, measuring moisture content in deeper organic material (1.2-7 cm depth). Drier duff layers result in higher DMC values, which are crucial for assessing medium-term fire danger.
   - **DC (Drought Code)**: A range of 7.9 to 860.6 that estimates moisture levels in deep organic layers (7+ cm depth). Higher DC values indicate prolonged dry conditions, making it an indicator of seasonal drought.
   - **ISI (Initial Spread Index)**: Ranges from 0.0 to 56.10, predicting the speed at which a fire could spread based on FFMC and wind speed.

4. **Additional Meteorological Variables**:
   - **Temperature (°C)**: Varies from 2.2 to 33.30°C, capturing the atmospheric conditions associated with fire risks. Higher temperatures are often linked to increased fire danger.
   - **Relative Humidity (%)**: Ranges from 15.0 to 100, indicating the air's moisture content, where lower humidity generally leads to drier vegetation and higher fire risks.

- **Wind Speed (km/h)**: Ranges from 0.40 to 9.40 km/h, influencing the spread and intensity of fires.

- **Rainfall (mm)**: Varies from 0.0 to 6.4 mm, where higher rainfall typically reduces fire risk by increasing fuel moisture.

5. **Target Variable**:

    - **Burned Area (hectare, ha.)**: The area of the forest burned by the fire, ranging from 0.00 to 1,090.84 ha. This variable is heavily skewed toward smaller values, making logarithmic transformation a potential method for better predictive modelling.

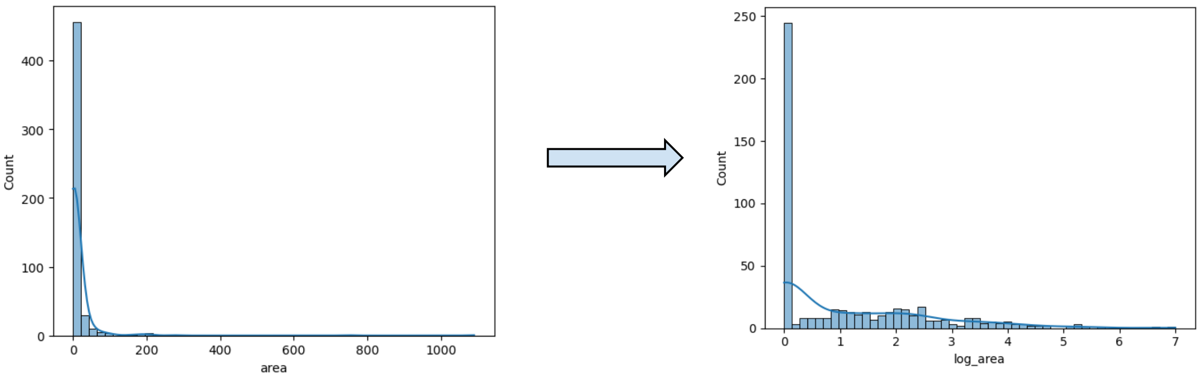


Fig.1 Change in the distribution of Area after log transformation

Canadian Forest Fire Weather Index (FWI) System*: The FWI system, developed in Canada, is widely used to assess fire danger by evaluating the moisture content in different layers of forest fuel and predicting fire behaviour under varying weather conditions. It includes indices like the FFMC, DMC, DC, and ISI, which account for daily weather observations such as temperature, wind, and rainfall. These indices provide insight into how fire-prone an area is under current environmental conditions, making them valuable predictors in forest fire modelling.*

EDA - Exploratory Data Analysis

Exploratory Data Analysis (EDA) is the first step for any project, through this we aim to understand the data better by analysing and visualizing the characteristics. Here, we used the software Python for all our computations

To perform EDA, we proceed as below :

- **Understanding the Size of the data:** Using this we see how many rows and columns our data set has, through this, we can find the number of independent variables we might have in the dataset.

  Our observation: We can see that we have 517 rows and 13 columns, which makes up about 6,721 data points.

- **Sample:** To understand the data in a better way, we can pull up a random sample of our preferred size, through this we can analyse the values of the variables of random data points.

- **Duplicate values:** Duplicate values are the data points that record the same data under all the variables. To handle the duplicate values, we remove the same to ensure accurate insights.

Our Observation & Rectification: We observed 4 duplicated rows in the dataset. To handle the same, we removed the duplicate values. This led to reducing the size of the data set dimensions to 513 rows and 13 columns.

- In this process, we also try to understand and study the **types of data** present in our dataset to categorize them into numerical or categorical data types for further analysis. Machine learning algorithms are not compatible enough to work on categorical data sets so we use encoding algorithms such as label encoder (for the target variable), One-hot encoding, or ordinal encoding (for independent variables) to convert categorical columns into numerical ones.

  Our observation & Rectification: We observe that there are only 2 categorical columns, i.e., the day and month columns and the rest are numerical type columns.

  Here we use **"One-Hot Encoding"**. One-hot encoding is a technique in machine learning that turns categorical data into numerical data for machines to understand. It creates new binary columns for each category, with a 1 marking the presence of that category and 0s elsewhere. This encoding resulted in increasing the number of columns in our dataset to 28.

- **Missing Values:** Addressing the variables that lack information is very important as the absence of data might introduce bias, and decrease the accuracy or reliability of the data as well.

  Our observation: We have observed no missing values in our dataset.

- **Correlation analysis:** The degree of linear association between 2 variables is measured by the correlation coefficient. It is a measure of the linear relationship among the variables. It is measured on a scale that varies from + 1 (perfect positive correlation) through 0 (no correlation) to – 1 (perfect negative correlation).

  Our observation: Here, we use a heatmap to visualize the degrees of associations.

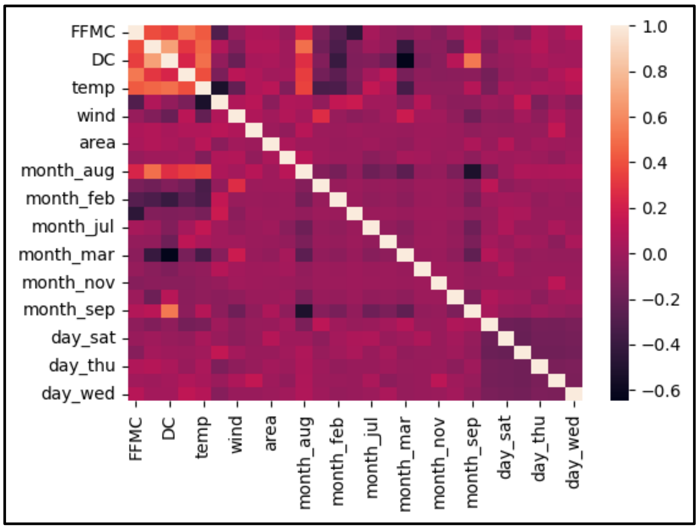


Fig.2 Heatmap of the correlation coefficients of the variables

- High positive correlation between FFMC & ISI (0.532), DMC and DC (0.682), temp and FFMC (0.432), temp and DMC (0.470), temp and DC (0.498), temp and ISI (-0.394)

- High negative correlation between RH and temp (-0.529)

- Month wise: September and DC (0.532), March & DC (-0.649)

- High positive correlation between X and Y (0.543).

- On studying the X and Y spatial coordinates separately we observed that the spatial coordinates towards the centre of the park have high occurrences of forest fire in the data. That means these areas are more prone to fire. Also, the coordinate (8, 6) has the highest number of occurrences of fire, i.e., 52.

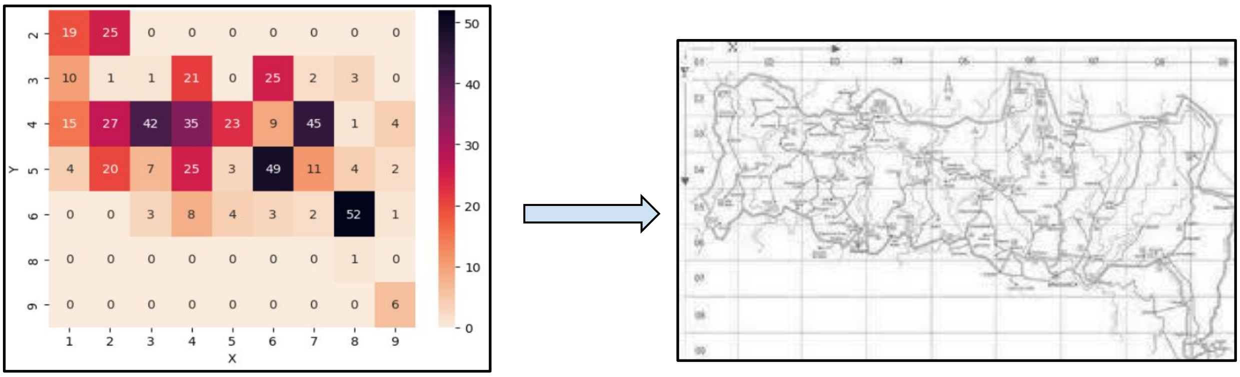


Fig.3 Heatmap corresponding to the original map of the region

- **Outlier detection**

An outlier is an unusual or abnormal data point that stands out in the dataset. It might be an extremely high or low value as compared to other observations in the dataset.

Methods of detection: There are various statistical, distance-based, clustering-based, and graphical methods of outlier detection. Here we discuss some of them.

1. **Graphical Method of Boxplots**: A box plot (aka box and whisker plot) uses boxes and lines to depict the distributions of one or more groups of numeric data. Box limits indicate the range of the central 50% of the data, with a central line marking the median value. Lines extend from each box to capture the range of the remaining data, with dots placed past the line edges to indicate outliers. They are built to provide high-level information at a glance, offering general information about a group of data's symmetry, skew, variance, and outliers.

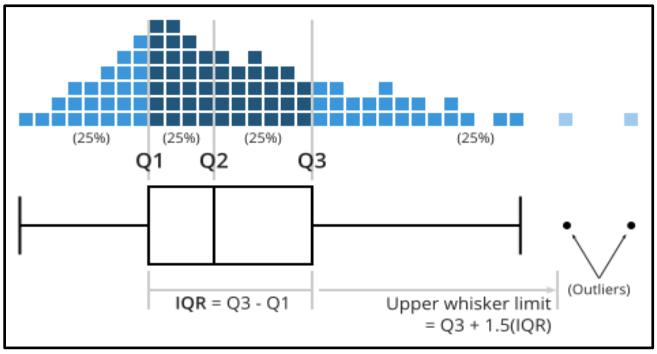


Fig.4 Explanation of construction of a boxplot

(source: www.atlassian.com)

2. **IQR Method:** IQR stands for interquartile range, which is the difference between Q3 (75th percentile) and Q1 (25th percentile). The IQR method computes lower bound and upper bound to identify outliers.

$$Lower\ Bound = Q_1 - 1.5 \times IQR$$
$$Upper\ Bound = Q_3 + 1.5 \times IQR$$

Any value below the lower bound and above the upper bound are considered to be outliers.

3. **Z-Score Method:** This method is generally used when a variable's distribution looks close to Gaussian. Z-score is the number of standard deviations a value of a variable is away from the variable's mean.

$$Z - Score = \frac{(X-Mean)}{Standard\ Deviation}$$

When the values of a variable are converted to Z-scores, then the distribution of the variable is called standard normal distribution with mean=0 and standard deviation=1. The Z-score method requires a cut-off specified by the user, to identify outliers. The widely used lower-end cut-off is -3 and the upper-end cut-off is +3. The reason behind using these cut-offs is, 99.7% of the values lie between -3 and +3 in a standard normal distribution.

Extreme Values v/s Outliers.
An extreme value is a data point that despite being abnormal concerns the data might hold some credibility and we can explain its observation by logical reasoning.

Methods of handling: To handle outliers we have many techniques, some of which are discussed below,

1. Trimming/Removal of data points
2. Flooring/ Capping of the data
3. Scaling
4. Log transformations

Our observation:

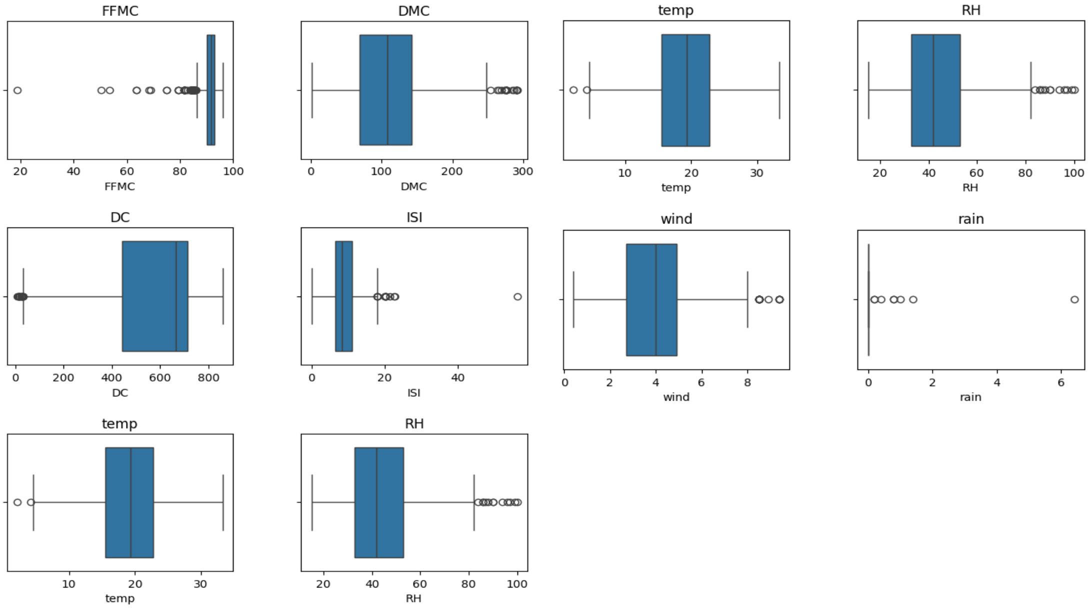


Fig.5 Boxplots used for the detection of outliers

Through the boxplots, we managed to identify the presence of outliers in the data. We moved further to examine the target variable first. To do this we first examined the distribution, observed left skewness in the data, and transformed it into close to Gaussian distribution by using a log transformation. We inspected the outliers in the data concerning the variable of the log of the area by the method of Z-scores. The scores were 5.3061 (highest allowed) and -3.0799 (lowest allowed). Through this, we identified the outliers (4) and studied them to conclude that the data points stated as outliers statistically were extreme values.

For the outliers detected by the mathematical formula in log_area, we notice that the areas where extreme values of the burned area have occurred are the ones which have one of the highest numbers of forest fires in the data. Hence, we are considering these values as extreme values and not outliers.

For all the other variables, we used the IQR method and detected the outliers. Here, we saw that most of the outliers could be classified as extreme values and hence we did not need to rectify the outliers for a lot of data.

We found 2 data points corresponding to FFMC & rain variables which seemed fair enough to be classified as outliers and opted to cap them with their upper bounds.

- **Visualization**

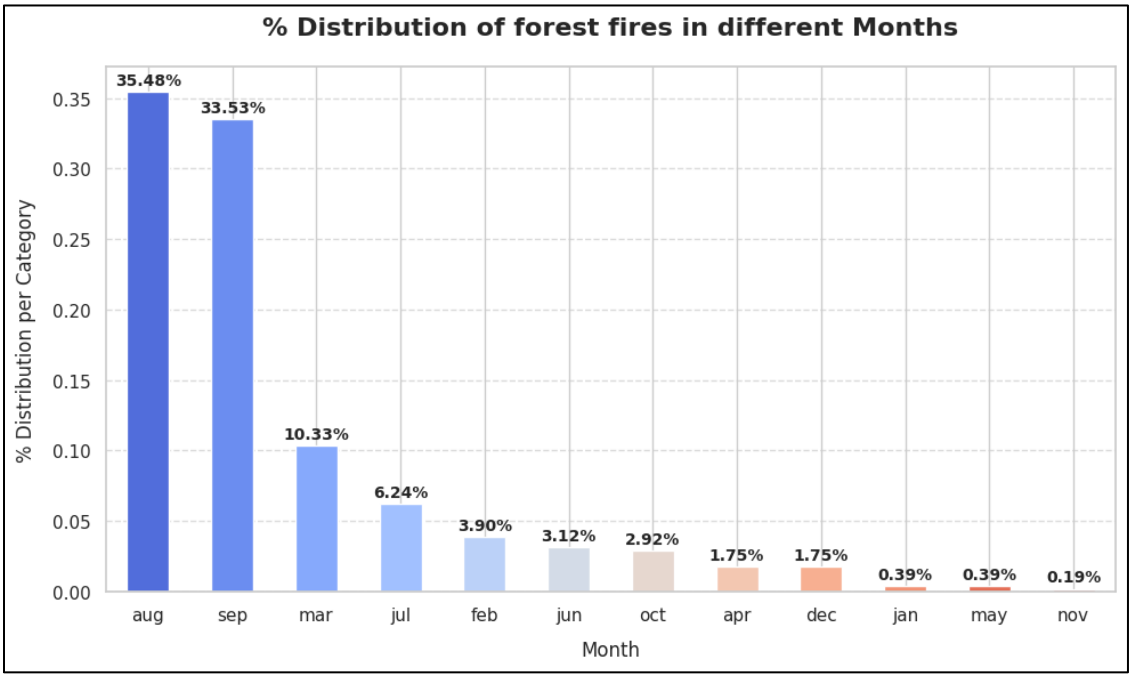


Fig.6 Distribution of Forest Fires in different Months

This graph displays the **percentage distribution of forest fires across different months**. The key observations are:

- **August** has the highest percentage of forest fires, with **35.48%** of total fires occurring during this month.
- **September** follows closely behind, accounting for **33.53%** of forest fires.
- **March** contributes **10.33%** to the total fires.
- **July** has **6.24%** of the fires, while **February** and **June** see a lower percentage at **3.90%** and **3.12%**, respectively.
- **October** has **2.92%**, and **April** and **December** both contribute **1.75%** each.

- The months of **January** and **May** have only **0.39%** each, and **November** has the least, with **0.19%**.

This suggests that **August and September** are the peak months for forest fires, while **November, January, and May** have very few fires. The bar chart provides a clear, visual representation of these seasonal patterns in forest fire occurrences. This aligns with the weather patterns in this region of Portugal since the driest and hottest month of the region is August, while December and January are the wettest and coldest months respectively.

Feature Transformation

In statistics, the normal distribution of the data is one that a statistician desires to be. To attain this distribution, we may implement feature transformations. The transformers are the type of functions that are applied to data that is not normally distributed, with an expectation to attain Gaussian-like distribution.

There are mainly 3 types of Feature transformation techniques:

1. Function Transformers - Log transformation, square root transformation, etc.
2. Power Transformers - Box-Cox Transform and Yeo-Johnson Transform
3. Quantile Transformers

Box-Cox Transform: The mathematical formulation of this transform is as follows:

$$X_i^\lambda = \begin{cases} \ln X \, ; for\ \lambda = 0 \\ \dfrac{X_i^\lambda - 1}{\lambda} \, ; for\ \lambda \neq 0 \end{cases} \quad ; \ \lambda \epsilon [-5, 5]$$

Where λ is the power of the data observations.

One major disadvantage associated with this transformation technique is that this technique can only be applied to positive observations.

Yeo-Johnson Transform: This is an advanced form of a box cox transformation technique where it can be applied to even zero and negative values of data observations. The mathematical formulation of this transformation technique is as follows:

$$X_i = \begin{cases} \dfrac{(y+1)^\lambda - 1}{\lambda} \, ; for\ y \geq 0\ and\ \lambda \neq 0 \\ \log(y+1) \, ; for\ y \geq 0\ and\ \lambda = 0 \\ \dfrac{(1-y)^{2-\lambda} - 1}{2-\lambda} \, ; for\ y < 0\ and\ \lambda \neq 2 \\ -\log(1-y) \, ; for\ y < 0\ and\ \lambda = 2 \end{cases}$$

Our observation:

By plotting histograms and by theoretical calculation, we observed skewness in some variables. We went forward and conducted an independent analysis for each variable. We applied the necessary transformations to the variables which were highly skewed and had sharp peaks after splitting the dataset into train and test sets:

- **FFMC-** We observed a right skew in the distribution. We applied function transformations which did not yield convincing results so we moved on to apply the box-cox transformation which showed normality in the data.

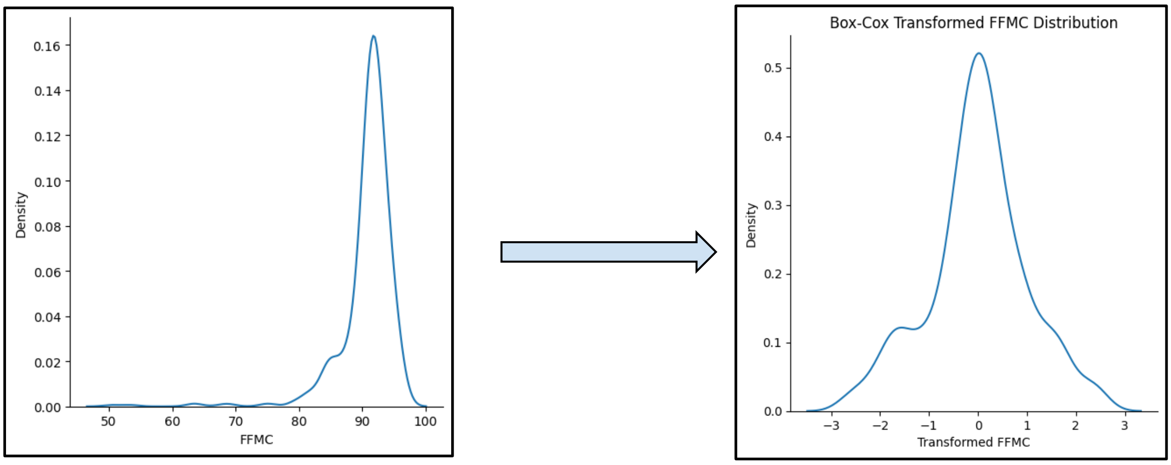


Fig.7 Initial and transformed distribution shape of the variable "FFMC"

- **ISI -** The left skew in data distribution was first treated by log transformation. Still, due to unsatisfactory results, we moved on and used the Yeo-Johnson Transformation and box-cox transformation out of which box-cox yielded better results.

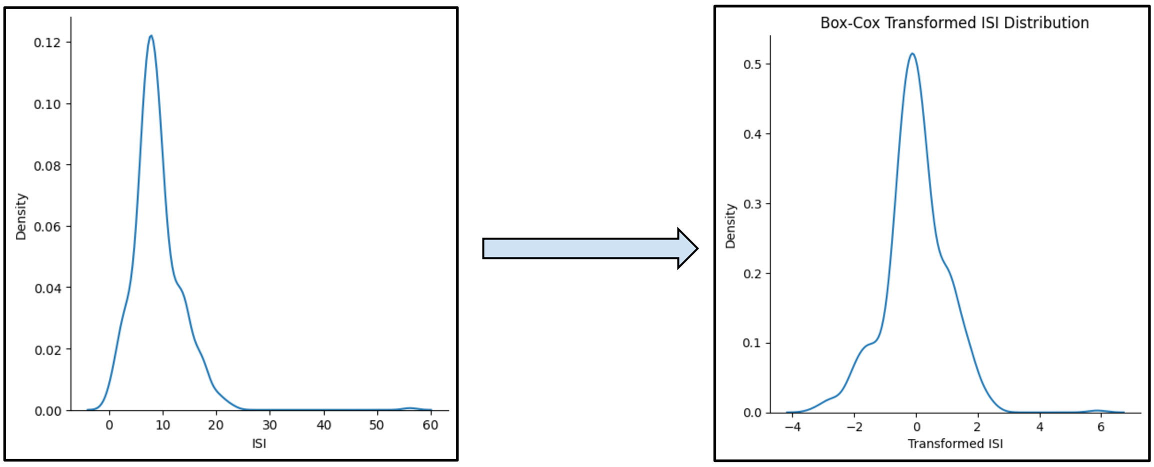


Fig.8 Initial and transformed distribution shape of the variable "ISI"

- **Rain -** The zero inflation (with less than 2% non-zero values in the rain column) could not be treated by any transformation, so we moved forward by performing **binarization** of the rain variable using the following logic,

{1 if rain occurred and 0 if rain did not occur}

Feature Construction

Feature construction is the process of creating new features or variables from existing data that can be used to improve the performance of machine learning models.

Our observation: In a default setup, machine learning interprets the spatial coordinates (1-9) as numeric data types, giving more weightage to the higher numbers. However, spatial coordinates should be studied nominally.

We created a new variable called "num_fires" which measured the number of fires caused in a specific pair of spatial coordinates. Through this, the machine gives more weightage to the coordinates having a higher number of fires irrespective of the numeric value of the coordinates.

Model Fitting

Linear Regression:

Simple linear regression is a statistical approach for predicting a quantitative response Y based on a single predictor variable X. It assumes that there is approximately a linear relationship between X and Y and is represented as

$$Y = \beta_0 + \beta_1 X + \varepsilon$$

When we have more than one predictor variable, Multiple Linear Regression can accommodate multiple predictors. We can do this by giving each predictor a separate slope coefficient in a single model. In general, suppose that we have p distinct predictors. Then the multiple linear regression model takes the form:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 \ldots + \beta_p X_p + \varepsilon$$

The best-fit line is estimated by minimizing the following loss function:

$$L = \frac{1}{n} \sum_{i=1}^{n} (yi - \hat{y})^2$$

Where, $\hat{y} = \widehat{\beta_0} + \widehat{\beta_1} X_1 + \widehat{\beta_2} X_2 \ldots + \widehat{\beta_p} X_p$

Assumptions of Linear Regression

Following are the assumptions of Linear Regression:

- **Linearity:** The independent and dependent variables have a linear relationship with one another. This implies that changes in the dependent variable follow those in the independent variable(s) linearly. This means that there should be a straight line that can be drawn through the data points.
- **Independence:** The observations in the dataset are independent of each other. This means that the value of the dependent variable for one observation does not depend on the value of the dependent variable for another observation. If the observations are not independent, then linear regression will not be an accurate model.
- **Homoscedasticity:** Across all levels of the independent variable(s), the variance of the errors is constant. This indicates that the amount of the independent variable(s) has no impact on the variance of the errors.
- **Normality:** The residuals should be normally distributed. This means that the residuals should follow a bell-shaped curve.
- **No multicollinearity:** This indicates that there is little or no correlation between the independent variables. Multicollinearity occurs when two or more independent variables are highly correlated with each other, which can make it difficult to determine the individual effect of each variable on the dependent variable.

Ridge Regularization:

Ridge Regression, also known as L2 regularization, is an extension to linear Regression that introduces a regularization term to reduce model complexity and help prevent overfitting. It helps minimize the sum of the squared residuals and the parameters' squared values scaled by a factor λ. The loss function in this case is given by:

$$L = \frac{1}{n}\sum_{i=1}^{n}(yi - \hat{y})^2 + \sum_{j=0}^{p}\lambda(ß_j{}^2)$$

LASSO Regularization:

LASSO (Least Absolute Shrinkage and Selection Operator) Regression is another regularization technique that prevents overfitting in linear Regression models. The difference lies in the loss function used — Lasso Regression uses L1 regularization, which aims to minimize the sum of the absolute values of coefficients multiplied by penalty factor λ.

$$L = \frac{1}{n}\sum_{i=1}^{n}(yi - \hat{y})^2 + \sum_{j=0}^{p}\lambda(|ß_j|)$$

Elastic Net Regression:

Elastic Net regression is a powerful machine learning technique that combines the strengths of both Ridge and Lasso regressions. It adds two penalty terms to the standard least-squares objective function in the following manner:

$$L = \frac{1}{n}\sum_{i=1}^{n}(yi - \hat{y})^2 + \sum_{j=0}^{p} a(|ß_j|) + \sum_{j=0}^{p} b(ß_j{}^2)$$

with $\lambda = a + b$ and $l1\_ratio = \frac{a}{a+b}$ as its two hyperparameters.

In our classification problem statement, we have the following hypothesis under test:

Null Hypothesis: The area burnt under forest fire is 0 hectares.

Alternative Hypothesis: The area burnt under forest fire is more than 0 hectares.

Decision Trees:

A decision tree is a flowchart-like structure used to make decisions (classification) or predictions (regression). It consists of nodes representing decisions or tests on attributes, branches representing the outcome of these decisions, and leaf nodes representing outcomes or predictions. Each internal node corresponds to a test on an attribute, each branch corresponds to the result of the test, and each leaf node corresponds to a class label or a continuous value.

Random Forest:

Random Forest algorithm is a powerful tree learning technique which works by creating a number of Decision Trees during the training phase. Each tree is constructed using a random subset of the data set to measure a random subset of features in each partition. This randomness introduces variability among individual trees, reducing the risk of overfitting and improving overall prediction performance.

It works as a classification model by returning the class with the highest votes as output while it works as a regression algorithm by returning the average of all the predictions made by constituent Decision Trees.

Logistic Regression:

Logistic regression is a statistical algorithm used for classification tasks where the goal is to predict the probability that an instance belongs to a given class or not. It is used for binary classification where we use a sigmoid function that takes input as independent variables and produces a probability value between 0 and 1.

For a dataset with independent features given by X, and the dependent variable Y taking only binary values, we have the multi-linear function to the input variables X given by:

$$z = (\sum_{i=1}^{m} wi \times xi) + b$$

Where x_i is the i^{th} observation and $w_m = [w_1, w_2, \ldots, w_m]$ is the weights or the coefficients.

And, the log loss function for Logistic Regression is given by:

$$Log\ L = \sum_{i=1}^{n} y_i \times log\ p(x_i) + (1 - y_i) \times log\ (1 - p(x_i))$$

Where $p(x_i) = \frac{1}{1+e^{-wX+b}}$

Assumptions of Logistic Regression

- **Independent observations**: Each observation is independent of the other. Meaning there is no correlation between any input variables.
- **Binary dependent variables**: It takes the assumption that the dependent variable must be binary or dichotomous, meaning it can take only two values.
- **Linearity** relationship between independent variables and log odds: The relationship between the independent variables and the log odds of the dependent variable should be linear.
- **No outliers**: There should be no outliers in the dataset.
- **Large sample size**: The sample size is sufficiently large

Two-stage classification and regression model

Zero Inflation

Zero inflation indicates that a dataset contains an excessive number of zeros. Here, we introduced a combination of Classification and Regression models to deal with the problem where many instances of forest fire data have a burned area of zero.

Traditional regression models struggle with such data because of the excessive zeros, which can distort predictions. Zero-Inflated Regression models, such as Zero-Inflated Poisson (ZIP) and Zero-Inflated Negative Binomial (ZINB), handle this by combining two processes: one to model the zeros and another to model the positive values.

Approach for Rectification

For our problem, first, we used Classification to determine whether the burnt area under forest fire is 0 hectares or more than that. If the classification predicted no area burnt under forest fire, the regression output was automatically set to zero. If the area under forest fire of more than 0 hectares was predicted, Regression was applied to predict the area affected by the fire. This combination efficiently handles the skewed distribution of the target variable and allows for more accurate predictions.

Evaluation Metrics:

Regression metrics:

1. **Mean Absolute Error (MAE)**

   The Mean Absolute Error gives us the average value of the total absolute differences between the predicted values output by the model and the actual values in the dataset.

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |y_i - \widehat{y_i}\ |$$

   Where y_i is the true value and $\widehat{y_i}$ is the predicted value.

   Values closer to 0 are considered better.

2. **Root Mean Square Error (RMSE)**

   Mean squared error states that finding the squared difference between actual and predicted value. RMSE is a simple square root of mean squared error.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n} (y_i - \widehat{y_i})^2}$$

   RMSE is more sensitive to large errors than MAE. It is commonly used for regression tasks where large errors are considered particularly undesirable.

3. **R² Score**

   The Coefficient of Determination — also referred to as R-squared — is a measure that tells us how well a regression model fits the actual data. It quantifies the degree to which the variance in the dependent variable is predictable from the independent variables.

$$R^2 = 1 - \frac{\sum(y_i - \widehat{y_i})^2}{\sum(y_i - \bar{y})^2}$$

   R-squared ranges from 0 to 1, with higher values, indicating that a larger proportion of variance is explained by the model. An R-squared value of 1 means the model perfectly predicts the data.

4. Adjusted R²

The adjusted R² is a modification of the R² score which accounts for the number of predictors in a model. While R² indicates how well the independent variables explain the variance of the dependent variable, it can be misleading because it always increases as you add more predictors to the model, even if they don't improve the model.

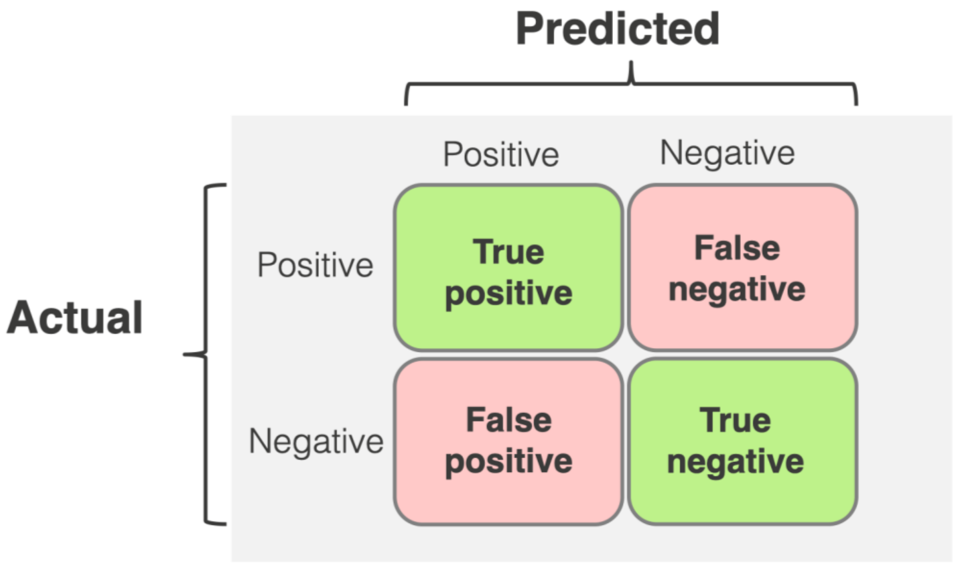
The adjusted R² penalizes for adding irrelevant predictors and provides a more accurate measure of model performance. It's particularly useful in multiple regression models.

$$Adj.R^2 = 1 - \frac{(1 - R^2)(n - 1)}{(n - k - 1)}$$

Where,
R^2 is the regular R-squared value.
n is the number of observations (sample size).
k is the number of predictors (independent variables)

Classification metrics:

Confusion matrix: A confusion matrix or error matrix is a table that shows the number of correct and incorrect predictions made by the model compared with the actual classifications in the test set or what type of errors are being made.



		Predicted	
		Positive	Negative
Actual	Positive	True positive	False negative
	Negative	False positive	True negative

Fig.9 Confusion Matrix of Classification

Here,

True Positives (TP): Number of outcomes that are actually positive and are predicted positive.

True Negatives (TN): Number of outcomes that are actually negative and are predicted negative.

False Positives (FP): Number of outcomes that are actually negative but predicted positive. These errors are also called Type 1 Errors.

False Negatives (FN): Number of outcomes that are actually positive but predicted negative. These errors are also called Type 2 Errors.

1. **Precision:** It is the ratio of True Positives to all the positives predicted by the model. The more False positives the model predicts, the lower the precision.

$$Precision = \frac{TP}{TP + FP}$$

2. **Recall:** It is the ratio of true positives to all the positives in your dataset. The more false negatives the model predicts, the lower the recall.

$$Recall = \frac{TP}{TP + FN}$$

3. **F1 Score:** It is a single metric that combines both Precision and Recall. The higher the F1 score, the better the performance of our model. The range for F1-score is [0,1]. The F1 score is the weighted average of precision and recall.

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

4. **Accuracy:** The simplest metric for model evaluation is Accuracy. It is the ratio of the number of correct predictions to the total number of predictions made for a dataset.

$$Accuracy = \frac{Total\ number\ of\ correct\ predictions}{Total\ number\ of\ predictions\ made}$$

Results and Discussion

Linear Regression model

The linear regression model, though simple, yielded a test set Root Mean Squared Error (RMSE) of 7.27 hectares, which indicates a moderate level of prediction accuracy. Given the complexity of forest fires and the non-linear interactions between environmental factors, this RMSE suggests that linear regression struggles to fully capture these dynamics.

Two-staged model of Classification and Regression

Classification models:

The performances of different classifiers are stated below:

Logistic Regression

Table 1 Classification Report (Train)

	Precision	Recall	F1 score
0	0.61	0.49	0.55
1	0.60	0.71	0.65

Train set accuracy: 0.6

The model shows slightly higher performance in detecting the positive class (1) compared to the negative class (0), with a better balance between precision and recall for class 1, leading to a higher F1 score**.**

Table 2 Classification Report (Test):

	Precision	**Recall**	**F1 score**
0	0.47	0.44	0.45
1	0.58	0.60	0.59

Test set accuracy: 0.53

The performance on the test set shows that the model performs slightly better in identifying class 1 (positive class) than class 0 (negative class). However, the overall accuracy of 53% indicates that the model is not very effective in generalizing to unseen data, with relatively low precision, recall, and F1 scores for both classes.

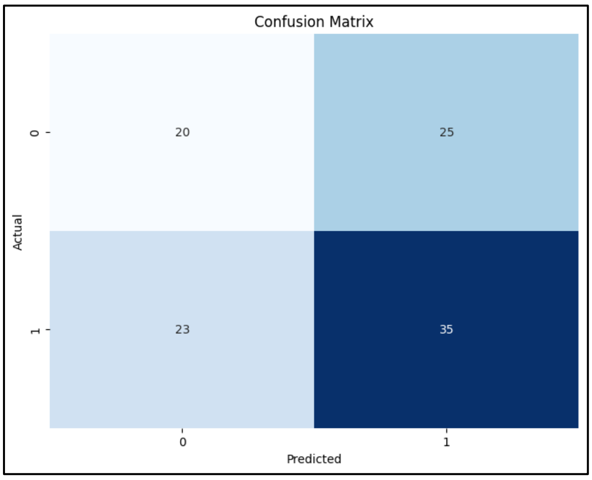


Fig.10 Confusion Matrix of Logistic Regression

Decision Tree:

Table 3 Classification Report (Train):

	Precision	**Recall**	**F1 score**
0	0.75	0.66	0.7
1	0.71	0.8	0.75

Train set accuracy: 0.73

The Decision Tree model shows a fairly balanced performance between both classes, with class 1 (positive class) having slightly better recall and F1 score. The overall accuracy of 73% indicates a moderate level of fit to the training data

Table 4 Classification Report (Test):

	Precision	**Recall**	**F1 score**
0	0.53	0.6	0.56
1	0.65	0.59	0.62

Test set accuracy: 0.59

The model demonstrates better precision for class 1 (positive class), but the recall is slightly higher for class 0 (negative class). Overall, the test set accuracy of 59% suggests that the model does not generalize very well to unseen data, with moderate performance for both classes.

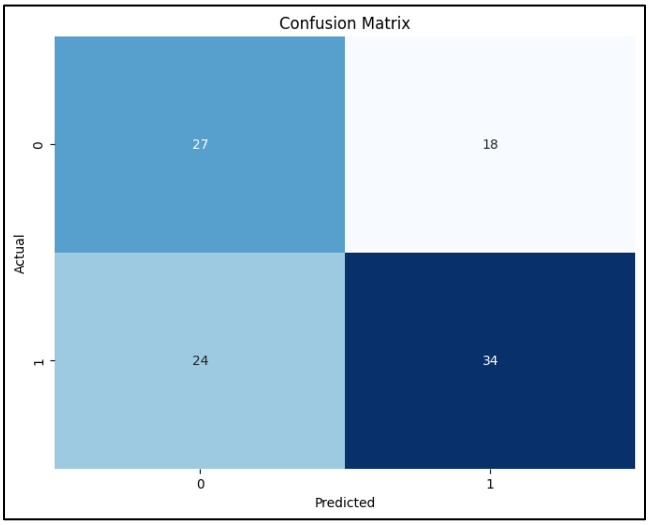


Fig.11 Confusion matrix of Decision Tree

Random Forest

Table 5 Classification Report (Train):

	Precision	Recall	F1 score
0	0.79	0.66	0.72
1	0.72	0.83	0.77

Train set accuracy: 0.75

Table 6 Classification Report (Test):

	Precision	Recall	F1 score
0	0.56	0.53	0.55
1	0.65	0.67	0.66

Test set accuracy: 0.61

- The train set shows a reasonable performance with an accuracy of 0.75, but there is a drop in performance on the test set, with accuracy dropping to 0.61.
- For Class 0, the performance on the test set is significantly lower compared to the train set in terms of both precision (0.56 vs. 0.79) and recall (0.53 vs. 0.66).
- For Class 1, the performance on the test set is also lower but more consistent, with a smaller drop in precision (0.65 vs. 0.72) and recall (0.67 vs. 0.83).

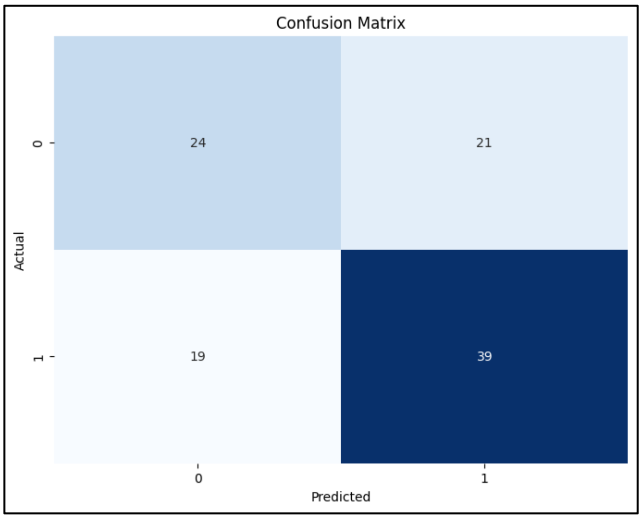


Fig.12 Confusion matrix of Random Forest

Criteria for deciding the best model:
We have 2 types of errors:
Type I error: Our model predicted area burnt under forest fire is greater than 0 hectares while it is actually 0 hectares.
Type II error: Our model predicted area burnt under forest fire is 0 hectares while it is actually greater than 0 hectares.

Clearly, Type II error is more serious in our problem. Hence, we should prioritize the minimization of Type II errors over the minimization of Type I error. The Logistic Regression classifier reported 23 instances of type II error, the Decision Tree reported 24 instances and the Random Forest reported 19 instances. Moreover, the overall accuracy is also maximum in the case of the Random Forest Classifier. Thus, we chose Random Forest over others.

Regression models:

Objective: To predict the area (in hectares) burnt under forest fire for the observations for which the classification model predicted that the burnt area was more than 0 hectares.

Linear Regression

Train set RMSE : 3.225
Test set RMSE: 3.437

Ridge Regression

Train set RMSE : 3.271
Test set RMSE: 3.435

Lasso Regression

Train set RMSE : 3.535

Test set RMSE: 3.402

Elastic Net Regression

Train set RMSE :   3.465
Test set RMSE: 3.437

Decision Tree Regression

Train set RMSE :   2.783
Test set RMSE: 4.143

Random Forest Regression

(Hyperparameters used: max_depth=5, n_estimators=150, random_state=14)

Train set RMSE : 1.634
Test set RMSE: 3.324

The Random Forest regression model performed the best, with the lowest RMSE on the test set. The RMSE values of the linear models—linear Regression, Ridge, Lasso, and Elastic Net—indicated relatively similar performances. The higher RMSE values for these linear models suggest that they struggled with the complex relationships in the data.

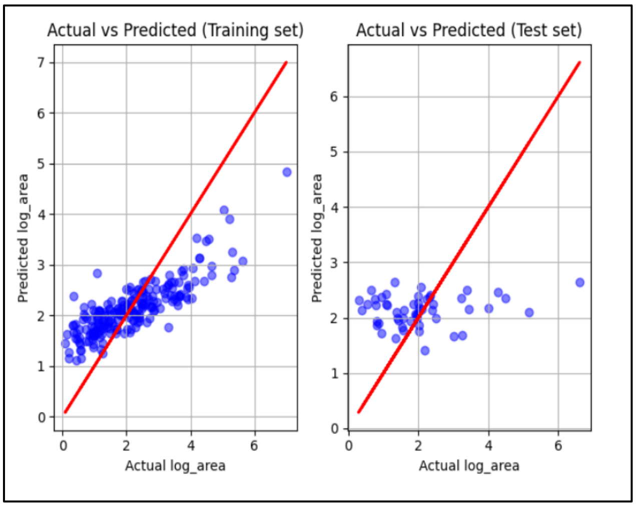


Fig.13 Actual vs Predicted observations using Random Forest Regressor

Interpretation: The closer the data points (blue dots) are to the red line, the better the fit of our model. The regressor performs better on the train set but isn't that efficient over the test set.

Combination of Classification and Regression Models

After observing the results of the individual models, we calculated the overall performance in terms of RMSE. By combining the Random Forest classifier and regressor, our model demonstrated superior performance. This method resulted in a significant improvement, achieving an RMSE of **2.535 hectares**, R² of **0.648**, and an **Adjusted R²** of **0.629** outperforming other models.

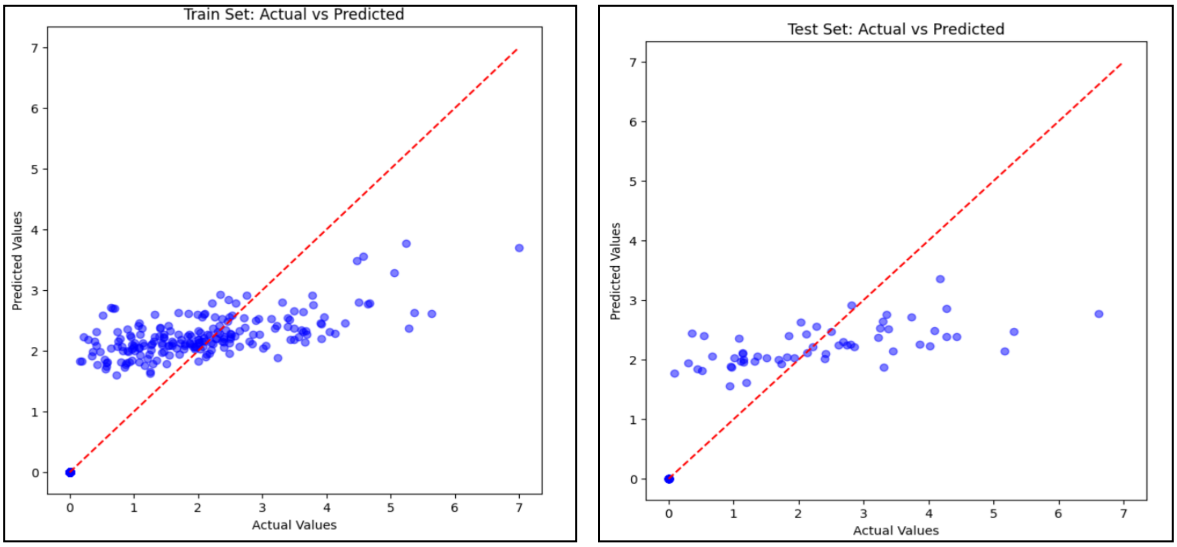


Fig.14 Actual vs Predicted observations using 2 Stage Model

Conclusion

When comparing simple linear regression with the two-stage classification and regression model, the latter improved RMSE by **65%**. Upon a fire event, the Random Forest Regression model predicted the burned area with improved precision, accounting for the complex, non-linear relationships between variables such as weather conditions and fuel moisture. Based on our experimentations, we conclude that a **two-staged model of Random Forest Classifier and Random Forest Regressor** is best suited for our problem and our final model reports an **RMSE** of **2.5436 hectares** over the test set. Thus, our model showed significant improvements over the traditional regression model, particularly in handling the zero-inflated nature of the dataset, demonstrating its suitability for predicting fire-prone areas and estimating their potential impact.

References

- *Cortez, P., & Morais, A. (2007). A Data Mining Approach to Predict Forest Fires Using Meteorological Data*. Proceedings of the 13th EPIA Conference on Artificial Intelligence.

- *Kumar, S., & Babu, R. (2015). Prediction of Forest Fire Risk Using Machine Learning Algorithms*. International Journal of Advanced Research in Computer Science and Software Engineering.

- *De Groot (1987). Interpreting the Canadian forest fire weather index (FWI) system.* A presentation made at the Fourth Central Region Fire Weather Committee Scientific and Technical Seminar, April 2, 1987, Winnipeg, Manitoba.

- *Forest Sector Report India(2019)*. Indian Council of Forestry Research and Education, Dehradun.