Project Report

On

FOREST FIRE ANALYSIS AND PREDICTION

Submitted

in partial fulfilment of the requirements for the degree of

MASTER OF SCIENCE

in

APPLIED STATISTICS AND DATA ANALYTICS

by

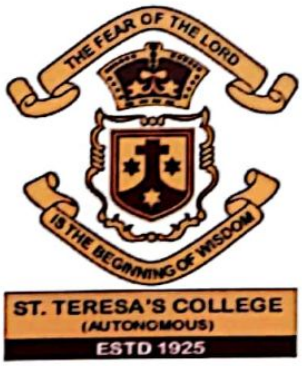
ANNA ELSA LUIZ

(Register No. SM21AS003)

(2021-2023)

Under the Supervision of

ANAKHA KURIAKOSE



DEPARTMENT OF MATHEMATICS AND STATISTICS

ST. TERESA'S COLLEGE (AUTONOMOUS)

ERNAKULAM, KOCHI - 682011

MAY 2023

CamScanner

ST. TERESA'S COLLEGE (AUTONOMOUS), ERNAKULAM

CERTIFICATE

This is to certify that the dissertation entitled, **FOREST FIRE ANALYSIS AND PREDICTION** is a bonafide record of the work done by Ms. **ANNA ELSA LUIZ** under my guidance as partial fulfillment of the award of the degree of **Master of Science in Applied Statistics and Data Analytics** at St. Teresa's College (Autonomous), Ernakulam affiliated to Mahatma Gandhi University, Kottayam. No part of this work has been submitted for any other degree elsewhere.

Date:
Place: Ernakulam

ST. TERESA'S COLLEGE DEPT. OF MATHEMATICS ERNAKULAM

ANAKHA KURIAKOSE
Assistant Professor,
Department of Mathematics and Statistics,
St. Teresa's College(Autonomous),
Ernakulam.

Smt Betty Joseph
Associate Professor & HOD,
Department of Mathematics and Statistics,
St. Teresa's College(Autonomous),
Ernakulam.

External Examiners

1: Dr. Joseph Jerlin Rebello 19/5/23

2: SARI THOMAS

CamScanner

DECLARATION

I hereby declare that the work presented in this project is based on the original work done by me under the guidance of **ANAKHA KURIAKOSE** , Assistant Professor, Department of Mathematics and Statistics, St. Teresa's College(Autonomous), Ernakulam and has not been included in any other project submitted previously for the award of any degree.

Ernakulam. **ANNA ELSA LUIZ**

Date: 28-05-2023 **SM21AS003**

CamScanner

ACKNOWLEDGEMENTS

I express my deepest gratitude to all for their contributions to this project. Without their expertise, guidance, and encouragement, this project would not have been possible.

I am extremely grateful to all my teachers especially my project guide Anakha Kuriakose and HoD Smt.Betty Joseph for the support and encouragement.

In addition, very energetic and competitive atmosphere of the Department had much to do with this work. I acknowledge with thanks to faculty, teaching and non-teaching staff of the department and Colleagues.

Above all , I thank God almighty and my parents for giving me the blessings to take over this project.

Ernakulam.
Date: 18-05-2023

ANNA ELSA LUIZ
SM21AS003

CamScanner

ABSTRACT

Forest fire prediction is an important topic in the field of environmental science, as it can help in the early detection and prevention of forest fires, thereby reducing the damage caused by them. This paper presents an abstract of the research conducted on forest fire prediction, which involves the use of various data mining and machine learning techniques. The study focuses on the analysis of weather conditions, topography, and other environmental factors that contribute to the occurrence of forest fires.

The research also involves the development of predictive models using regression models like linear , lasso , ridge, support vector ,random forest, K-nearest neighbour and classifier models like logistic , decision tree , random forest , K- nearest , Xgboost. And optimized models for predicting the Fire Weather Index (FWI) and predict whether the area is prone to fire or not based on the meterological data.

This study show that the developed models can provide accurate predictions, and can be used by forest management authorities to take appropriate measures to prevent forest fires. Overall, the study highlights the importance of early detection and prediction in forest fire management, and provides insights for future research in this field.

CamScanner

ST.TERESA'S COLLEGE (AUTONOMOUS) ERNAKULAM

Certificate of Plagiarism Check for Thesis

Author Name	ANNA ELSA LUIZ
Course of Study	MSc. Applied Statistics & Data Analytics
Name of Guide	Ms. ANAKHA KURIAKOSE
Department	Mathematics & Statistics
Acceptable Maximum Limit	20%
Submitted By	library@teresas.ac.in
Paper Title	FOREST FIRE ANALYSIS AND PREDICTION
Similarity	0%
Paper ID	724151
Submission Date	2023-04-18 16:37:56

Signature of Student

Signature of Guide

Checked By
College Librarian

* This report has been generated by DrillBit Anti-Plagiarism Software

CamScanner

Contents

CamScanner

CamScanner

CamScanner

Chapter 1

INTRODUCTION

Forest fires are one of the most significant natural disasters that can occur. These disasters can cause substantial economic, ecological, and environmental damage. With the increase in human activities and climate change, the occurrence and severity of forest fires have increased significantly. Therefore, accurate and timely prediction of forest fires has become a crucial aspect of forest management and protection.

There are several types of forest fires, including: Surface fires,Ground fires, Crown fires, Surface and crown fires ,Spot fires

Each type of forest fire has unique characteristics and requires different approaches for prevention and management. It is essential to understand the different types of forest fires to develop effective forest fire prevention and management strategies.

Used a dataset on Algerian Forest Fires from UCI. The dataset contains a culmination of forest fire observations and data in two regions of Algeria: the Bejaia region and the Sidi Bel-Abbes region. The timeline of this dataset is from June 2012 to September 2012.In this project, I focused on whether certain weather features could predict forest fires in these regions using few Machine Learning algorithms.

Machine learning algorithms such as decision trees, support vector, random forest , xgboost , lasso , k-nearest neighbour, linear , ridge, logistic will be utilized to develop predictive models. These models will be trained on the collected data to predict the likelihood of forest

CamScanner

fires. The system will be designed to provide early warning alerts to forest management authorities, enabling them to take timely measures to prevent or mitigate the damage caused by forest fires.

The outcomes of this project are expected to contribute significantly to forest fire management and protection. The system will provide accurate predictions of forest fire occurrence, aiding in the development of appropriate prevention and management strategies. The results of this project will provide insights into the interactions between environmental factors and forest fires and could be used to develop more effective forest fire prevention and management strategies.

1.1 Relevance of the Project

Forest fire prediction projects are highly relevant for preventing and managing forest fires, which can have significant economic, social, and environmental consequences. These projects provide critical information on fire risk and behavior, which can be used to develop prevention and response strategies, allocate resources more effectively, and reduce the impact of fires on the environment, human lives, and property. Overall, forest fire prediction projects are crucial for protecting natural ecosystems, human lives, and property, and improving the efficiency and effectiveness of forest fire management.

1.2 Problem Statement

1. Predicting the Fire Weather Index in the northeast region of Portugal, based on the spatial, temporal and weather variables where the fire is spotted.

2. To predict the occurrence of forest fire , give alert as “fire “ and “not fire”.

1.3 Objectives

1. To build Machine Learning models.

2. Compare the accuracy of the model.

CamScanner

3. Propose the most accurate model.

4. Create an user interface for testing different test data to predict FWI (fire weather index) and the occurrence of fires .

1.4 Scope of the project

1. Data collection and analysis: This involves collecting and analyzing data on factors that contribute to forest fires, such as temperature,humidity , wind speed etc.

2. Development of predictive models: Using machine learning techniques, predictive models

can be developed to estimate the probability of forest fires occurring in specific areas and under certain conditions.

3. Integration of data and models: The collected data and developed models can be integrated to create a comprehensive forest fire prediction system that takes into account multiple factors.

4. Early warning systems: Once a forest fire prediction system is developed, early warning systems can be implemented to provide timely alerts and warnings to authorities and communities.

5. Response planning: A forest fire prediction project can also include the development of response plans to mobilize resources and personnel in case of a forest fire.

The scope of a forest fire prediction project can also include research into new technologies and techniques to improve the accuracy and reliability of predictions, as well as outreach and education efforts to increase public awareness of forest fire risks and prevention measures.

CamScanner

Chapter 2

LITERATURE REVIEW

- George E. Sakr, Imad H. Elhajj, George Mitri and Uchechukwu C. Wejinya "Artificial Intelligence for Forest Fire Prediction" 2010 IEEE/ASME. International Conference on Advanced Intelligent Mechatronics Montréal, Canada, July 6-9, 2010: This paper presented a forest fire risk prediction method. The findings show that a small quantity of data can be used to estimate forest fire risk.[6]

- Mauro Castelli, Leonardo Vanneschi, and Ales Popovic "Predicting burned areas of forest fires: an artificial intelligence approach" Fire Ecology 2015: They demonstrated a novel intelligent GP-based approach for examining burned areas in this demonstration. The major goal was to create a system that could forecast how much land will be destroyed in the event of a forest fire. The experimental findings revealed that geometric semantic genetic programming outperforms due to the small MAE.[12]

- Chao Gao , Honglei and Haiqing Hu .Forest-Fire-Risk Prediction Based on Random Forest and Backpropagation Neural Network of Heihe Area in Heilongjiang Province, China.Based on daily historical forest-fire data from 1995 to 2015, daily meteorological data, topographic data and basic geographic information data, the main forest-fire driving factors were first analyzed by using RF importance characteristic evaluation and logistic stepwise regression.[2]

CamScanner

- **A. Kansal, Y. Singh, N. Kumar and V. Mohindru, "Detection of forest fires using machine learning technique: A perspective" 2015 Third International Conference on Image Information Processing (ICIIP), Waknaghat, 2015:[9] The use of regression and the division of datasets has been proposed in this paper as a method for detecting fire. The algorithm achieves a low R-squared and a low root mean square error. This method could be used for other calamities in the future. The use of specific transformations may also help to increase the model's efficiency.**

- **L. Yu, N. Wang, and X. Meng "Real-time forest fire detection with Wireless Sensor Networks" in Wireless Communications, Networking and Mobile Computing, 2005. Proceedings. 2005 International Conference on, vol. 2. IEEE, International Journal of Advance Research, Ideas and Innovations in Technology © 2021, www.IJARIIT.com All Rights Reserved Page —2198 2005: Ensemble learning is used at all cluster heads in this case. At the base station, SVM, a supervised machine learning technique, is used with a polynomial kernel function. Carbon dioxide, temperature, humidity, and carbon monoxide can all be detected using the sensors that have been installed. Clustered stream generates data in tabular or clustered form. After that, the SVM is used to detect fire.[10]**

- **Paulo Cortez and Anibal Morais "A Data Mining Approach to Predict Forest Fires using Meteorological Data": They investigate a Data Mining approach for predicting the burned area of forest fires in this paper.[3] The optimal configuration combines an SVM with four meteorological inputs to forecast the burned area of minor fires. Such information is especially valuable for bettering the administration of firefighting resources. In this work, we explore a Data Mining (DM) approach to predict the burned area of forest fires. Five different DM techniques, e.g. Support Vector Machines (SVM) and Random Forests, and four distinct feature se lection setups (using spatial, temporal, FWI components and weather at-**

CamScanner

tributes),were tested on recent real-world data collected from the northeast region of Por tugal. The best configuration uses a SVM and four meteorological inputs (i.e.temperature, relative humidity, rain and wind) and it is capable of predicting the burned area of small fires, which are more frequent.

CamScanner

Chapter 3

METHODOLOGY

3.1 Data Collection and Understanding

Data used for this particular study is collected from UCI Machine Learning repository.The dataset includes 244 instances that regroup a data of two regions of Algeria ,namely the Bejaia region located in the northeast of Algeria and the Sidi Bel-abbes region located in the northwest of Algeria.
122 instances for each region.
The period from June 2012 to September 2012.The dataset includes 11 attribues and 1 output attribue (class).The 244 instances have been classified into fire (138 classes) and not fire (106 classes) classes.

3.2 Data Exploration

In this step, apply Exploratory Data Analysis (EDA) on the given dataset. And extract meaningfull insights from the dataset to know which features have contributed more in predicting Forest fire fire by performing Data Analysis using Pandas and Data visualization using Matplotlib Seaborn

3.2.1 Explorartory Data Analysis

1. Data Cleaning: Check for missing data, duplicates, and outliers in the data set. Remove any irrelevant or redundant data points that do

CamScanner

not contribute to predicting forest fires.

2. Data Visualization: Create graphs and charts to visualize the data and identify any patterns or trends. For example, you may want to plot the frequency of forest fires by month, or the relationship between temperature and the severity of forest fires.

3. Univariate Analysis: This involves analyzing one variable at a time. The purpose of univariate analysis is to identify the distribution and summary statistics (mean, median, mode, standard deviation, range) of the variable.

4. Bivariate Analysis: This involves analyzing the relationship between two variables. Bivariate analysis helps identify patterns and relationships between variables.

5. Correlation Analysis: Perform correlation analysis to identify the strength and direction of the relationship between predictor variables and the target variable (i.e. forest fires). This will help you identify the most important predictors and eliminate redundant features.

6. Multivariate Analysis: This involves analyzing the relationship between more than two variables. Multivariate analysis helps identify complex patterns and relationships between variables.

The goal of EDA is to gain a better understanding of the data and identify patterns that can inform further analysis. By following these steps, you can gain insights into the data and identify the most important variables for further analysis.

3.2.2 Analysis Report

Reporting based on exploratory data analysis (EDA) involves summarizing the key findings and insights from the analysis in a clear and concise manner. The report should include a brief introduction to the data, the methods used for the analysis, and the main results and conclusions.

CamScanner

3.3 Data preprocessing for Regression and classification

3.3.1 Train-test splitting:

It is a technique used in machine learning to evaluate the performance of a predictive model. It involves dividing the available data into two separate sets: a training set and a testing set. The training set is used to train the predictive model, while the testing set is used to evaluate the performance of the model on new, unseen data. This technique allows us to estimate how well the model will perform on new data and avoid overfitting. The split is usually done randomly with a common split of 70-80

3.3.2 Feature scaling:

Feature Scaling is a technique used in machine learning to transform features to be on the same scale. StandardScaler is a common technique for feature scaling, where each feature is transformed to have a mean of 0 and standard deviation of 1. This involves subtracting the mean of each feature from its values and dividing by the standard deviation. StandardScaler is useful for algorithms that assume that features are normally distributed, and can improve model convergence and performance. It's important to apply StandardScaler after train-test splitting, and to fit the scaler on the training data and transform both the training and testing data using the same scaler to avoid information leakage.

3.4 Model building for Regression

3.4.1 Linear Regression

It is one of the easiest and most popular machine learing algorithm.It is a statistical method that is used for predictive analysis. The linear regression algorithm shows a linear relationship between a dependant (y) and one or more independent (x) variables , hence called linear regression. A linear regressor is a specific implementation of linear regression that uses a linear equation to model the relationship between

CamScanner

the input variables and the output variable. It can be used for both simple and multiple regression problems, where there is one or more input variables respectively.

The linear regression model provides a sloped straight line representing the relationship between the variables.It can be expressed mathematically as:

y= $a_o + a_1 x + \epsilon$

where y is Dependant Variable (Target Variable), x is the Independent Variable (Predictor Variable) , a_o is intercept of the line (gives an additional degree of freedom), a_1 is the Linear regression coefficient (scale factor to each input value) and ϵ is random error

The values for x and y variables are training datasets for Linear Regression model representation

3.4.2 Lasso Regression

Lasso regression, also known as L1 regularization, is a linear regression technique that adds a penalty term to the cost function of the standard linear regression model. This penalty term is the sum of the absolute values of the model coefficients, which encourages the model to reduce the coefficients of irrelevant or redundant features to zero. The Lasso regression model can be represented using the following formula:

y= $\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots\ldots + \beta_p X_p + \epsilon$

where y is the dependant variable, $x_1, x_2, ..., x_p$ are the independent variables, β_o is the intercept, $\beta_1, \beta_2, ..., \beta_p$ are the coefficients, ϵ is the error term.

3.4.3 Ridge Regression

Ridge regression is a linear regression technique that adds a penalty term to the cost function of the standard linear regression model. This penalty term is the sum of the squared values of the model coefficients, which encourages the model to reduce the coefficients of irrelevant or redundant features to small values, but not necessarily zero.

The formula for Ridge regression can be represented as:

y= $\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots\ldots + \beta_p X_p + \epsilon$

CamScanner

where y is the dependant variable, $x_1, x_2, ..., x_p$ are the independent variables, β_o is the intercept, $\beta_1, \beta_2, ..., \beta_p$ are the coefficients, ϵ is the error term.

3.4.4 Support Vector Regressor

Support Vector Regression (SVR) is a regression algorithm based on Support Vector Machines (SVM). It is used for regression tasks where the goal is to predict continuous output values. The formula for SVR can be represented as:

$y = f(x) = b + \sum(\alpha_i * k(x_i, x))$

where y is the predicted output,b is the bias term, α_i are the Lagrange multipliers, x_i are the training input samples, x is the input sample to be predicted,and $K(x_i, x)$ is a kernel function that computes the similarity between the training samples and the test sample.

The objective of SVR is to find a hyperplane that best fits the training data, subject to a tolerance margin . This is achieved by minimizing the following cost function.

3.4.5 Random Forest Regressor

Random Forest Regressor is a type of machine learning algorithm used for regression tasks. It is an ensemble learning method that combines multiple decision trees to make more accurate predictions.

The formula for a Random Forest Regressor can be described as follows:

$y = f(X) + \epsilon$

where:

y is the dependent variable or target variable that we want to predict f(X) is the function that represents the relationship between the independent variables X and the target variable y. In the case of a Random Forest Regressor, this function is the average of the predicted values from the individual decision trees in the forest.

X is a matrix of independent variables or features that are used to predict the target variable y.

CamScanner

ϵ is the random error or noise in the data that cannot be explained by the independent variables. The Random Forest Regressor algorithm works by building multiple decision trees on random subsets of the training data and then averaging the predicted values from these trees to make the final prediction. This approach helps to reduce overfitting and improve the accuracy of the model.

3.4.6 K- Nearest Neighbors Regressor

K-Nearest Nearest Neighbors Regressor is a machine learning algorithm used for regression tasks. It works by finding the k nearest neighbors to a given data point and making predictions based on the average of the values of the k nearest neighbors.

The formula for K-Nearest Neighbors Regressor can be described as follows:

$$y = \frac{1 * \sum(y1 + y2 + ... + yn)}{k}$$

where y is the predicted value for a new data point,k is the number of nearest neighbors to consider when making predictions, y1, y2, ..., yn are the values of the k nearest neighbors of the new data point
In this formula, the predicted value y for a new data point is calculated as the average of the values of the k nearest neighbors of the new data point.

3.5 Camparison of accuracy of the models for Regression

3.5.1 Mean Absolute Error

Mean Absolute Error (MAE) is a metric used to measure the average absolute difference between the predicted and actual values in a regression problem. It gives an idea of how close the predictions are to the true values on average. The formula for calculating MAE is:

CamScanner

$$MAE = \frac{1}{n} * \sum_{i=1}^{n} |actual_i - predicted_i|$$

where:n is the number of samples, $actual_i$ is the actual value of the i-th sample , $predicted_i$ is the predicted value of the i-th sample

3.5.2 $R^2 Score$

R^2 score, also known as the coefficient of determination, is a metric used to measure the proportion of variance in the target variable that is explained by the regression model. It ranges from 0 to 1, with 1 indicating a perfect fit. The formula for calculating R_2 score is:

$R_2 = 1 - \frac{SS_res}{SS_tot}$

where SS_res is the sum of squared residuals (the sum of the squared differences between the actual and predicted values),SS_tot is the total sum of squares (the sum of the squared differences between the actual values and the mean of the actual values)

3.6 Hyperparameter Tuning

Hyperparameter tuning is the process of optimizing the hyperparameters of a machine learning algorithm to achieve the best possible performance. Hyperparameters are user-defined parameters that affect the behavior of the model during training, such as learning rate, regularization strength, and the number of layers in a neural network. Hyperparameter tuning can be done using various techniques such as grid search, random search, Bayesian optimization, or evolutionary algorithms. The goal is to find the optimal values for the hyperparameters that will result in the most accurate model on unseen data. Hyperparameter tuning is a critical step in machine learning model development and can significantly improve the model's performance.

3.6.1 Random Search

Random search is a hyperparameter tuning technique that involves randomly selecting hyperparameter values from a specified search space

CamScanner

and evaluating the performance of the resulting model. Unlike grid search, which evaluates every combination of hyperparameters in a predefined search space, random search randomly samples a subset of hyperparameters from the space, reducing the number of models that need to be trained.

The advantage of random search is that it can be more efficient than grid search when the hyperparameter space is large, as it does not require evaluating all possible combinations. Additionally, random search may be able to find better hyperparameter values than grid search because it allows for a wider exploration of the hyperparameter space.

3.7 Model building for classification

3.7.1 Logistic regression

Logistic regression is a statistical method used to predict the probability of a binary outcome based on one or more predictor variables. It is a type of supervised learning algorithm that falls under the category of linear models.

The logistic regression model uses a logistic or sigmoid function to model the relationship between the predictor variables and the output variable. The function returns a value between 0 and 1, representing the probability of the event occurring.

The logistic function can be expressed as follows:

$$P(y = 1/x) = \frac{1}{(1 + e^{(} - (0 + 1x_1 + 2x_2 + ... + px_p)))}$$

where P(y=1/x) is the probability of the event occurring given the input variables x, 0 is the intercept, 1 to p are the coefficients for the input variables x1 to xp, and exp is the exponential function.

3.7.2 Decision Tree classifier

A Decision Tree classifier is a supervised learning algorithm used for both classification and regression problems. It is a tree-like model where internal nodes represent feature tests or decisions, branches represent

CamScanner

the outcome of these tests, and leaf nodes represent the class label or regression value.

The algorithm builds the tree by recursively splitting the dataset into smaller subsets using the feature that provides the most information gain, until the tree reaches a stopping criterion, such as reaching a maximum depth or having a minimum number of samples at each leaf node.

The Decision Tree classifier predicts the class label of an instance by traversing the tree from the root to a leaf node that corresponds to the predicted class.

3.7.3 Random Forest classifier

Random Forest classifier is an ensemble learning algorithm used for classification, regression, and other tasks. It is based on constructing multiple decision trees during training and using them to make predictions by aggregating the outputs of the individual trees.

The algorithm builds each decision tree using a random subset of the features and a random subset of the training samples, which reduces the correlation between the trees and improves their diversity. During prediction, the output of each tree is aggregated using a majority vote (for classification) or an average (for regression).

The probability of an instance belonging to a particular class can be estimated by computing the fraction of trees that predict that class. The Random Forest classifier formula can be expressed as follows:

$$P(y/x) = \frac{1}{T} * \sum_{i=1}^{T} I(y = y_i)$$

where $P(y/x)$ is the predicted probability of class y given input x, T is the total number of trees, y_i is the class predicted by the i-th tree, and I is the indicator function that returns 1 if its argument is true and 0 otherwise.

CamScanner

3.7.4 K-Nearest Neighbors (KNN)

K-Nearest Neighbors (KNN) classifier is a supervised learning algorithm used for classification and regression tasks. It is a non-parametric and instance-based algorithm that stores the entire training dataset during training and makes predictions based on the k-nearest neighbors of the test instance in the feature space.

The algorithm works by computing the distance between the test instance and all the training instances, and selecting the k closest instances as neighbors. The class label of the test instance is then determined by a majority vote of the neighbors' class labels, weighted by their distances from the test instance.

The KNN classifier formula can be expressed as follows:

$$P(y/x) = \frac{1}{k} * \sum_{i=1}^{k} I(y = y_i)$$

where $P(y/x)$ is the predicted probability of class y given input x, k is the number of neighbors, y_i is the class label of the i-th neighbor, and I is the indicator function that returns 1 if its argument is true and 0 otherwise.

3.7.5 XGBoost (eXtreme Gradient Boosting) classifier

XGBoost (eXtreme Gradient Boosting) classifier is a supervised learning algorithm used for classification, regression, and ranking tasks. It is an ensemble learning method that combines multiple weak decision trees to produce a strong classifier. The algorithm works by adding decision trees sequentially, where each new tree is trained to correct the errors of the previous trees. During training, the algorithm optimizes an objective function that measures the difference between the predicted and actual values, using techniques such as gradient boosting and regularization. The XGBoost classifier formula can be expressed as follows:

CamScanner

$$P(y/x) = \frac{1}{1 + e^{(} - f(x))}$$

where $P(y/x)$ is the predicted probability of class y given input x, $f(x)$ is the sum of the outputs of all the decision trees for input x, and e is the exponential function.

3.8 Camparison of the accuracy of the models for classification

3.8.1 Accuracy

The accuracy of a machine learning model is a metric used to measure the performance of the model in making correct predictions on a given dataset. It represents the proportion of correct predictions made by the model over the total number of predictions. The accuracy can be calculated using the following formula:

$$Precision = \frac{numberofcorrectpredictions}{totalnumberofprediction}$$

3.8.2 Precision

Precision is a metric used to evaluate the performance of a machine learning classifier in terms of the number of true positive predictions over the total number of positive predictions made. It measures the model's ability to avoid false positive predictions. The precision can be calculated using the following formula:

$$Precision = \frac{numberoftruepositives}{numberoftruepositives + numberoffalsepositives}$$

CamScanner

3.8.3 Recall

Recall is a metric used to evaluate the performance of a machine learning classifier in terms of the number of true positive predictions over the total number of actual positive instances in the dataset. It measures the model's ability to identify all positive instances. The recall can be calculated using the following formula:

$$Recall = \frac{number of true positives}{number of true positives + number of false negatives}$$

3.8.4 Support

Support is a metric used to measure the number of instances in a dataset that belong to a particular class. It is used to evaluate the distribution of instances across different classes in the dataset and to determine the relative importance of each class in the model's predictions. The support can be calculated by counting the number of instances belonging to a particular class in the dataset.

3.8.5 Confusion Matrix

A confusion matrix is a table that summarizes the performance of a machine learning classifier on a dataset by showing the counts of true positive, true negative, false positive, and false negative predictions for each class. It is a tool used to evaluate the accuracy of a classification model and to identify the types of errors made by the model. The confusion matrix is a square matrix with the rows and columns representing the predicted and actual class labels, respectively. The diagonal entries of the matrix represent the correct predictions, while the off-diagonal entries represent the incorrect predictions.

CamScanner

3.9 Kfold cross validation

K-fold cross-validation is a technique used to evaluate the performance of a machine learning model. It involves dividing the dataset into k equally sized parts or "folds", and training the model on k-1 of the folds while using the remaining fold for validation. The process is repeated k times, with each fold being used once for validation. The results of the k iterations are averaged to produce a final evaluation metric, such as accuracy or mean squared error. K-fold cross-validation is commonly used to estimate the performance of a model, particularly when the dataset is small or when the model has a large number of parameters that can lead to overfitting. By dividing the dataset into k folds, we can obtain a more accurate estimate of the model's performance than by simply evaluating it on a single validation set. The choice of k depends on the size of the dataset and the computational resources available. Common choices for k include 5, 10, or 20. A higher value of k can result in a more accurate estimate of the model's performance, but can also be computationally expensive.

3.10 Mean CV accuracy score

Mean CV accuracy score refers to the average accuracy score obtained by performing k-fold cross-validation on a machine learning model. Cross-validation is a technique used to evaluate the performance of a model by dividing the dataset into k folds and training the model k times, each time using a different fold as the validation set. The mean CV accuracy score is calculated by taking the average of the accuracy scores obtained in each fold. It is a useful metric to evaluate the performance of a model because it provides an estimate of the model's accuracy on unseen data.

3.11 Feature importance

Feature importance is a measure of the relative importance of each feature or variable in a machine learning model for predicting the target

CamScanner

variable. It indicates the contribution of each feature to the overall performance of the model and can be used to identify the most important features and gain insights into the underlying relationships between the features and the target variable.

Feature Importance from Trees: For tree-based models such as random forests and decision trees, feature importance can be calculated based on the number of times a feature is used for splitting in the tree. The higher the number of times a feature is used, the more important it is.

3.12 Prediction using test data

Prediction using test data is the process of using a trained machine learning model to make predictions on a new, unseen dataset (the test data). The goal is to evaluate the performance of the model on data that it has not seen before, and to ensure that the model can generalize well to new data. The model is used to make predictions on the test data, using the predict() function or method.

CamScanner

Chapter 4

DATASET AND EXPLORARTORY DATA ANALYSIS

4.1 Dataset

The dataset is collected from University of California Irvine (UCI) Machine Learning Repository.

	day	month	year	Temperature	RH	Ws	Rain	FFMC	DMC	DC	ISI	BUI	FWI	Classes
0	01	06	2012	29	57	18	0	65.7	3.4	7.6	1.3	3.4	0.5	not fire
1	02	06	2012	29	61	13	1.3	64.4	4.1	7.6	1	3.9	0.4	not fire
2	03	06	2012	26	82	22	13.1	47.1	2.5	7.1	0.3	2.7	0.1	not fire
3	04	06	2012	25	89	13	2.5	28.6	1.3	6.9	0	1.7	0	not fire
4	05	06	2012	27	77	16	0	64.8	3	14.2	1.2	3.9	0.5	not fire
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
241	26	09	2012	30	65	14	0	85.4	16	44.5	4.5	16.9	6.5	fire
242	27	09	2012	28	87	15	4.4	41.1	6.5	8	0.1	6.2	0	not fire
243	28	09	2012	27	87	29	0.5	45.9	3.5	7.9	0.4	3.4	0.2	not fire
244	29	09	2012	24	54	18	0.1	79.7	4.3	15.2	1.7	5.1	0.7	not fire
245	30	09	2012	24	64	15	0.2	67.3	3.8	16.5	1.2	4.8	0.5	not fire

246 rows × 14 columns

- **Date : (DD/MM/YYYY) Day, month ('june' to 'september'), year**

CamScanner

(2012) Weather data observations

- **Temp : temperature noon (temperature max) in Celsius degrees: 22 to 42**
- **RH : Relative Humidity in**
- **Ws :Wind speed in km/h: 6 to 29**
- **Rain: total day in mm: 0 to 16.8**

FWI Components

- **Fine Fuel Moisture Code (FFMC) index from the FWI system: 28.6 to 92.5**
- **Duff Moisture Code (DMC) index from the FWI system: 1.1 to 65.9**
- **Drought Code (DC) index from the FWI system: 7 to 220.4**
- **Initial Spread Index (ISI) index from the FWI system: 0 to 18.5**
- **Buildup Index (BUI) index from the FWI system: 1.1 to 68**
- **Fire Weather Index (FWI) Index: 0 to 31.1**
- **Classes: two classes, namely are Fire and not Fire**

Temperature: is a key factor in predicting forest fires. Higher temperatures increase the likelihood of ignition and spread of fires by drying out vegetation and soil, making them more combustible. Extreme heatwaves can also lead to more intense and long-lasting fires. Monitoring temperature patterns is critical for predicting fire risks and informing fire management strategies.

Relative humidity: is a critical factor in predicting the risk of wildfires. As relative humidity decreases, the moisture content in vegetation and soil decreases, making them more susceptible to ignition and contributing to the spread of fires. Low relative humidity, combined with

CamScanner

high temperatures and winds, increases the likelihood and severity of wildfires.

Wind speed: is a crucial factor in predicting forest fires. Strong winds can cause fires to spread rapidly by fanning flames and carrying embers long distances. Wind direction also affects the direction of fire spread, making it challenging for firefighters to contain the fire. Monitoring wind speed and direction is thus essential in predicting and managing forest fires.

Rainfall: is a crucial factor in fire prediction. Adequate rainfall can reduce the risk of wildfires by increasing soil moisture and reducing the flammability of vegetation. Conversely, prolonged droughts can increase the risk of wildfires by creating dry conditions that promote the spread of fires. Rainfall patterns are thus an essential consideration for fire management and prediction.

The Fine Fuel Moisture Code (FFMC): is an index used in forest fire prediction to measure the moisture content of fine fuels, such as dead leaves and twigs. The FFMC is an important indicator of fire risk, as dry fuels are more prone to ignition and can quickly spread fires. By tracking the FFMC, firefighters can assess the potential for fires and adjust their strategies accordingly.

The Duff Moisture Code (DMC): is an index used in forest fire prediction to measure the moisture content of decomposing organic material, such as leaves and twigs on the forest floor. The DMC is a crucial indicator of fire risk, as dry organic material can fuel and spread fires quickly. By monitoring the DMC, firefighters can assess the potential for fires and take appropriate actions to prevent or control them.

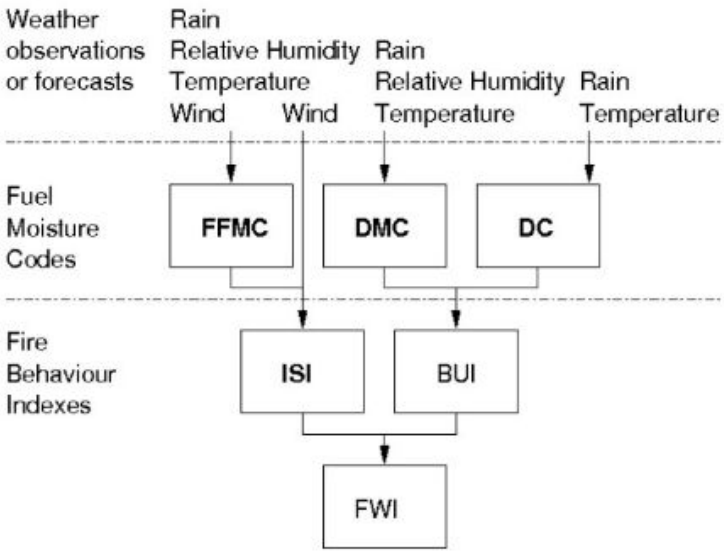
The Drought Code (DC): is an index used in forest fire prediction to measure the moisture content of deep organic layers, such as peat and decomposed wood. The DC reflects the long-term drying trend in forest fuels and is a critical indicator of fire risk. High DC values indicate dry conditions, which can increase the risk of fires and make them more difficult to control. By tracking the DC, firefighters can assess fire risks and implement appropriate preventive measures.

CamScanner

The Initial Spread Index (ISI): is an index used in forest fire prediction to assess the rate of fire spread during the initial stages of a fire. The ISI considers several factors such as wind speed, fuel type, and moisture content to estimate the potential rate of fire spread. By tracking the ISI, firefighters can anticipate how fast a fire might spread and use this information to develop an appropriate response plan.

The Buildup Index (BUI): is an index used in forest fire prediction to estimate the amount of fuel available for combustion. The BUI considers factors such as fuel loading, moisture content, and previous fire history to assess the potential for fires to start and spread. By tracking the BUI, firefighters can anticipate fire risks and implement appropriate preventive measures to reduce the buildup of fuel and control the spread of fires.

The Fire Weather Index (FWI): is a comprehensive index used in forest fire prediction that integrates several weather and fuel moisture indices to assess overall fire danger. The FWI considers factors such as temperature, relative humidity, wind speed, and fuel moisture to estimate the potential for fire ignition and spread. By tracking the FWI, firefighters can anticipate fire risks and implement appropriate preventive measures to reduce the likelihood and severity of fires.



CamScanner

In [39]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 246 entries, 0 to 245
Data columns (total 14 columns):
 #   Column       Non-Null Count  Dtype 
---  ------       --------------  ----- 
 0   day          246 non-null    object
 1   month        245 non-null    object
 2   year         245 non-null    object
 3   Temperature  245 non-null    object
 4    RH          245 non-null    object
 5    Ws          245 non-null    object
 6   Rain         245 non-null    object
 7   FFMC         245 non-null    object
 8   DMC          245 non-null    object
 9   DC           245 non-null    object
 10  ISI          245 non-null    object
 11  BUI          245 non-null    object
 12  FWI          245 non-null    object
 13  Classes      244 non-null    object
dtypes: object(14)
memory usage: 27.0+ KB
```

All the features are found to be objects and can be converted into numeric.

4.2 Data Cleaning

		day	month	year	Temperature	RH	Ws	Rain	FFMC	DMC	DC	ISI	BUI	FWI	Classes
122	Sidi-Bel Abbes Region Dataset	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
167		14	07	2012	37	37	18	0.2	88.9	12.9	14.6 9	12.5	10.4	fire	NaN

The dataset is converted into two sets based on Region from 122th index

1 : "Bejaia Region Dataset"

2 : "Sidi-Bel Abbes Region Dataset"

Added new coloumn with 'region'. Null values and other unwanted rows are removed using drop() and dropna() . Fixed the coloumn names which were mispaced using the str.strip().Changed the required coloumn as integer data type and float data type.

CamScanner

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 243 entries, 0 to 242
Data columns (total 15 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   day          243 non-null    int32  
 1   month        243 non-null    int32  
 2   year         243 non-null    int32  
 3   Temperature  243 non-null    int32  
 4   RH           243 non-null    int32  
 5   Ws           243 non-null    int32  
 6   Rain         243 non-null    float64
 7   FFMC         243 non-null    float64
 8   DMC          243 non-null    float64
 9   DC           243 non-null    float64
 10  ISI          243 non-null    float64
 11  BUI          243 non-null    float64
 12  FWI          243 non-null    float64
 13  Classes      243 non-null    object 
 14  Region       243 non-null    int32  
dtypes: float64(7), int32(7), object(1)
```

Data description is given below ;

	count	mean	std	min	25%	50%	75%	max
day	243.0	15.761317	8.842552	1.0	8.00	16.0	23.00	31.0
month	243.0	7.502058	1.114793	6.0	7.00	8.0	8.00	9.0
year	243.0	2012.000000	0.000000	2012.0	2012.00	2012.0	2012.00	2012.0
Temperature	243.0	32.152263	3.628039	22.0	30.00	32.0	35.00	42.0
RH	243.0	62.041152	14.828160	21.0	52.50	63.0	73.50	90.0
Ws	243.0	15.493827	2.811385	6.0	14.00	15.0	17.00	29.0
Rain	243.0	0.762963	2.003207	0.0	0.00	0.0	0.50	16.8
FFMC	243.0	77.842387	14.349641	28.6	71.85	83.3	88.30	96.0
DMC	243.0	14.680658	12.393040	0.7	5.80	11.3	20.80	65.9
DC	243.0	49.430864	47.665606	6.9	12.35	33.1	69.10	220.4
ISI	243.0	4.742387	4.154234	0.0	1.40	3.5	7.25	19.0
BUI	243.0	16.690535	14.228421	1.1	6.00	12.4	22.65	68.0
FWI	243.0	7.035391	7.440568	0.0	0.70	4.2	11.45	31.1
Region	243.0	1.497942	0.501028	1.0	1.00	1.0	2.00	2.0

CamScanner

Region 1 (Bejaia Region)

```
df[:122]
```

	day	month	year	Temperature	RH	Ws	Rain	FFMC	DMC	DC	ISI	BUI	FWI	Classes	Region
0	1	6	2012	29	57	18	0.0	65.7	3.4	7.6	1.3	3.4	0.5	not fire	1
1	2	6	2012	29	61	13	1.3	64.4	4.1	7.6	1.0	3.9	0.4	not fire	1
2	3	6	2012	26	82	22	13.1	47.1	2.5	7.1	0.3	2.7	0.1	not fire	1
3	4	6	2012	25	89	13	2.5	28.6	1.3	6.9	0.0	1.7	0.0	not fire	1
4	5	6	2012	27	77	16	0.0	64.8	3.0	14.2	1.2	3.9	0.5	not fire	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
117	26	9	2012	31	54	11	0.0	82.0	6.0	16.3	2.5	6.2	1.7	not fire	1
118	27	9	2012	31	66	11	0.0	85.7	8.3	24.9	4.0	9.0	4.1	fire	1
119	28	9	2012	32	47	14	0.7	77.5	7.1	8.8	1.8	6.8	0.9	not fire	1
120	29	9	2012	26	80	16	1.8	47.4	2.9	7.7	0.3	3.0	0.1	not fire	1
121	30	9	2012	25	78	14	1.4	45.0	1.9	7.5	0.2	2.4	0.1	not fire	1

122 rows × 15 columns

Region 2 (Sidi-Bel Abbes Region)

```
df[122:]
```

	day	month	year	Temperature	RH	Ws	Rain	FFMC	DMC	DC	ISI	BUI	FWI	Classes	Region
122	1	6	2012	32	71	12	0.7	57.1	2.5	8.2	0.6	2.8	0.2	not fire	2
123	2	6	2012	30	73	13	4.0	55.7	2.7	7.8	0.6	2.9	0.2	not fire	2
124	3	6	2012	29	80	14	2.0	48.7	2.2	7.6	0.3	2.6	0.1	not fire	2
125	4	6	2012	30	64	14	0.0	79.4	5.2	15.4	2.2	5.6	1.0	not fire	2
126	5	6	2012	32	60	14	0.2	77.1	6.0	17.6	1.8	6.5	0.9	not fire	2
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
238	26	9	2012	30	65	14	0.0	85.4	16.0	44.5	4.5	16.9	6.5	fire	2
239	27	9	2012	28	87	15	4.4	41.1	6.5	8.0	0.1	6.2	0.0	not fire	2
240	28	9	2012	27	87	29	0.5	45.9	3.5	7.9	0.4	3.4	0.2	not fire	2
241	29	9	2012	24	54	18	0.1	79.7	4.3	15.2	1.7	5.1	0.7	not fire	2
242	30	9	2012	24	64	15	0.2	67.3	3.8	16.5	1.2	4.8	0.5	not fire	2

121 rows × 15 columns

CamScanner

4.3 Exploratory Data Analysis

Encoded the classes 'fire' and 'not fire' as '1' and '0'.

And the Year, month,day values are dropped using drop().

4.3.1 Visualization of distribution

Univariate visualization

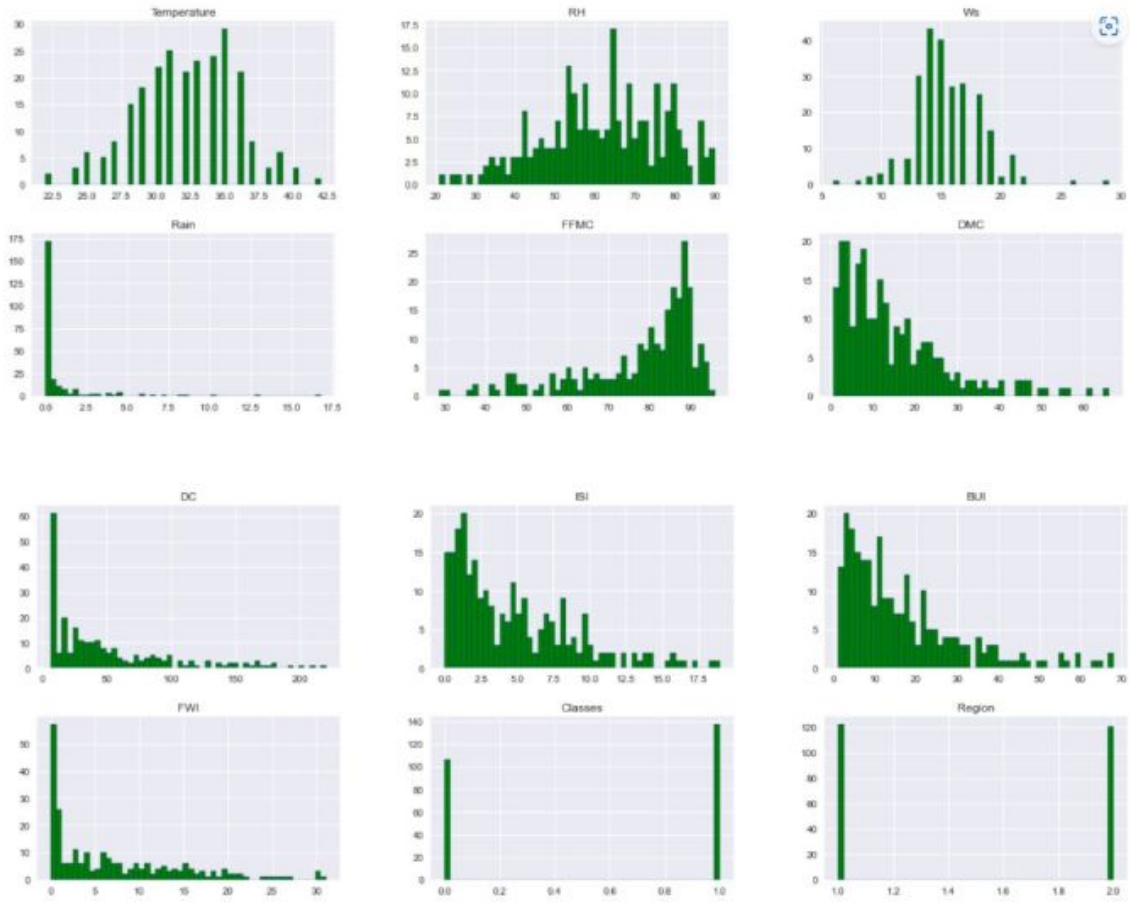


Figure 4.1: Histogram of attributes

CamScanner

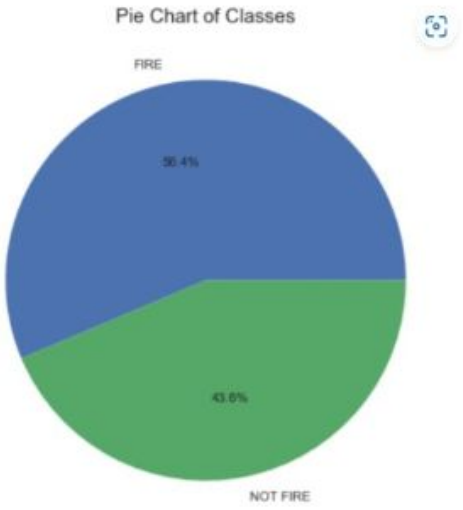


Figure 4.2: Pie chart

Multivariate Visualisations

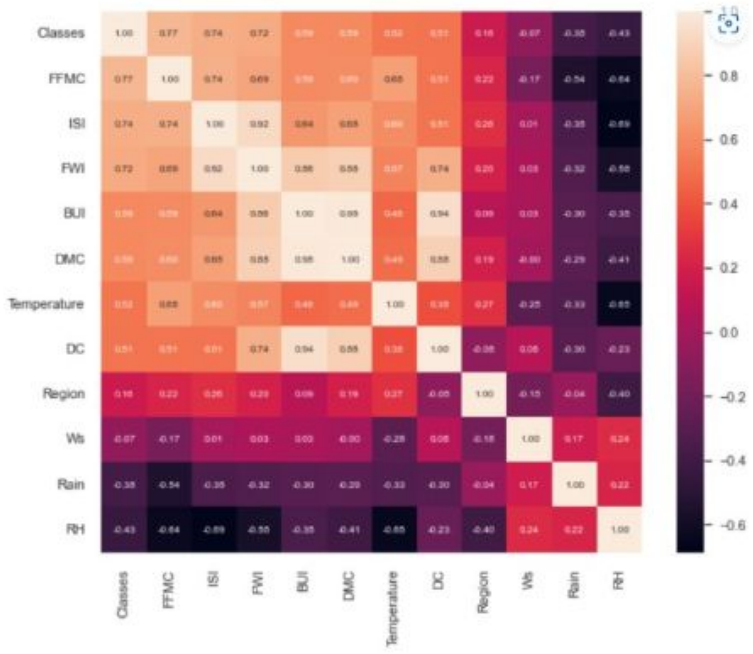


Figure 4.3: Heat map

CamScanner

FWI has a positive correlation of 0.72 with classes, negative correlation of -0.58 with RH , negative correlation of -0.32 with Rain, positive correlation of 0.03 with WS, positive correlation of 0.57 with temperature,positive correlation of 0.88 with DMC and a positive correlation of 0.92 with ISI and 0.69 with FFMC.

Classes has a negative correlation of -0.43 with RH, negative correlation of -0.38 with Rain, negative correlation of -0.07 with WS , positive correlation of 0.52 with temperature, positive correlation of 0.59 with DMC, positive correlation of 0.74 with ISI , positive correlation of 0.77 with FFMC and positive correlation of 0.72 with FWI

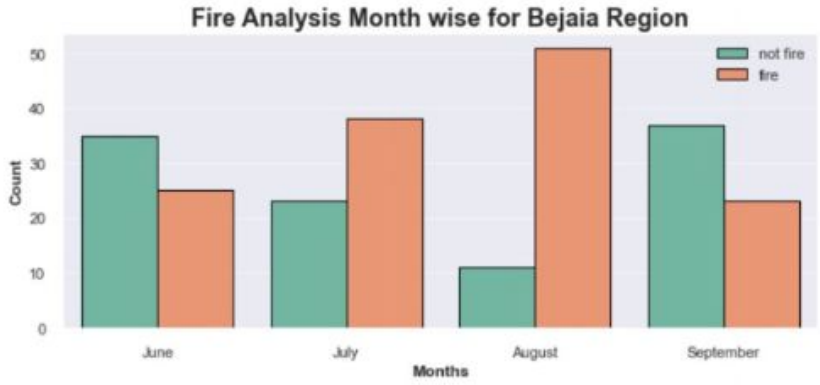


Figure 4.4: Bar graph

Occurence of Fire is more in the month of August and less fire is on september.

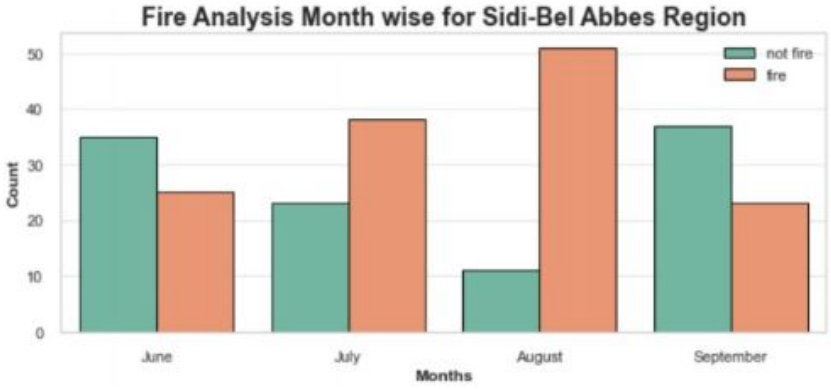


Figure 4.5: Bar graph

CamScanner

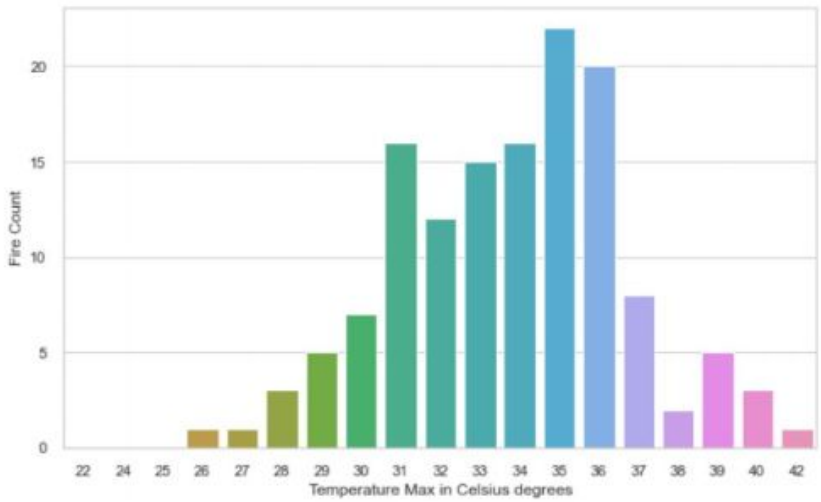


Figure 4.6: Bar graph

Figure 4.7: Bar graph

CamScanner

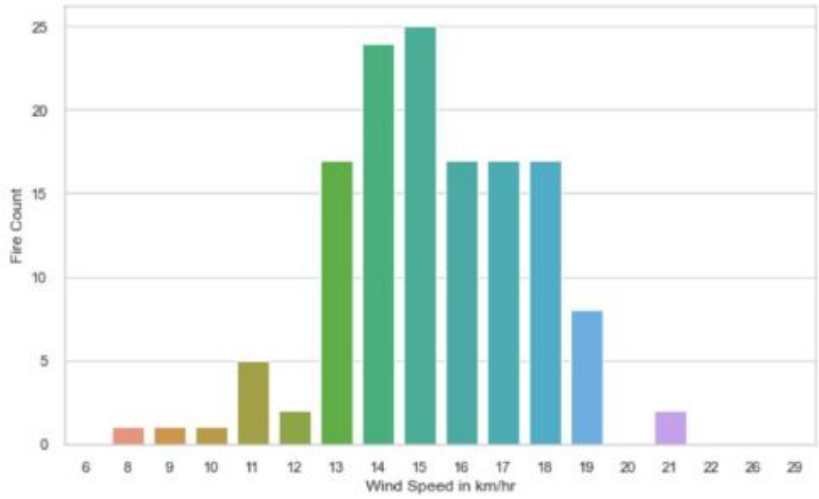


Figure 4.8: Bar graphs

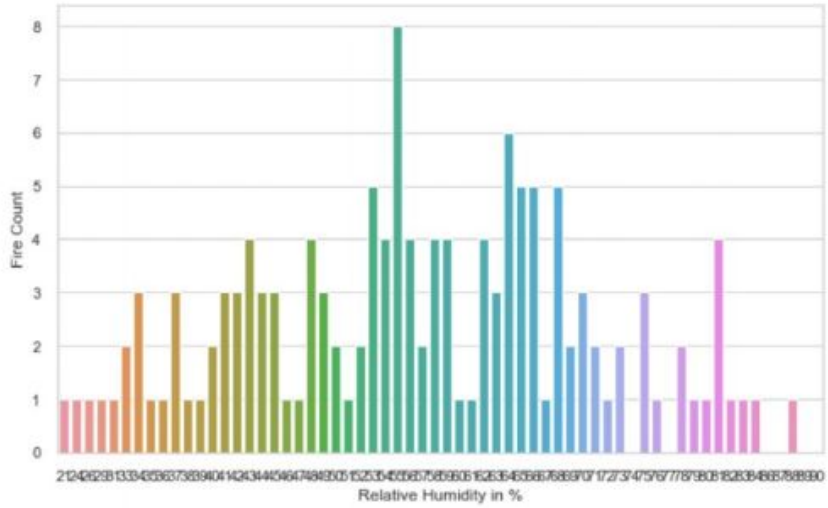


Figure 4.9: Bar graph

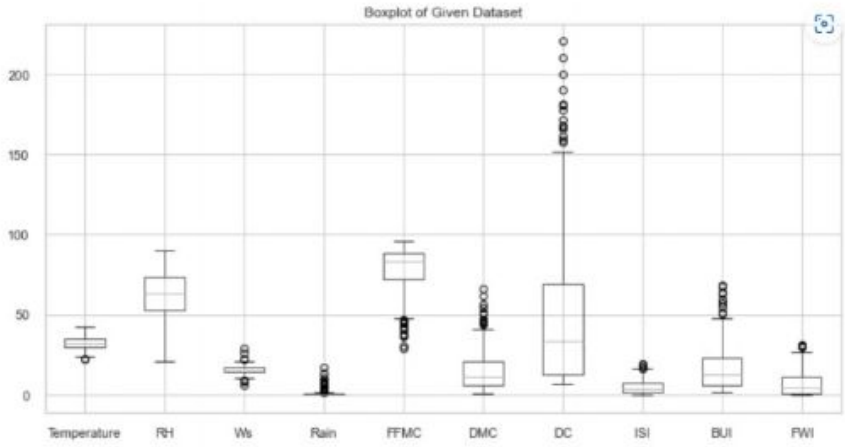


Figure 4.10: Box Plot

CamScanner

Distribution of Temparature in celcius degree (Figure 4.6) , Rain in mm (Figure 4.7), wind speed in km/hr (Figure 4.8) , Relative Humidity in % (Figure 4.9) and Box plot (4.10) of given data set which shows the outliers in each attributes.

4.3.2 Analysis Report

Weather System Report

- **30-37 degree Celsius is the highest Temparature count.**
- **0.0 to 0.3 is the lowest range of rain obtained where highest fire count happened.**
- **13 to 19 Km/hr. is the range of Wind Speed at which highest fire count happened item 50 to 80 percentage is the percentage range of Relative Humidity at which highest fire count happened.**

FWI System Components Report

FWI (Canadian Forest Fire Weather Index)

- **Fine Fuel Moisture Code (FFMC) index : 28.6 to 92.5, higher chance of Forest fires are at above 75.**
- **Duff Moisture Code (DMC) index : 1.1 to 65.9, here 1.1-10 has lower chance of Forest fires whereas above 10-30 DMC has very high evidence of Forest fires in past.**
- **Drought Code (DC) index : 7 to 220.4, here 0-25 is safe and has lower chance of Forest fires whereas range above 25 DC has higher chance of forest fires.**
- **Initial Spread Index (ISI) index : 0 to 18, here 0-3 has lower Forest fires and above 3 ISI has higher chance of Forest fires.**
- **Buildup Index (BUI) index : 1.1 to 68, here 1.1 to 10 has lower Forest fire chance and above 10 BUI has higher chance of forest fires.**

CamScanner

- **Fire Weather Index (FWI) Index : 1 to 31.1, here 0-3 has lower chance of Forest fires and 3-25 FWI has higher chance of forest fires.**

CamScanner

Chapter 5

TESTING

5.1 Regression

	Temperature	RH	Ws	Rain	FFMC	DMC	DC	ISI	BUI	Classes	Region
Temperature	1.000000	-0.657325	-0.357016	-0.365941	0.684556	0.482965	0.349021	0.618172	0.447959	0.512299	0.296033
RH	-0.657325	1.000000	0.262581	0.275592	-0.653649	-0.393893	-0.203883	-0.712353	-0.333027	-0.446906	-0.427696
Ws	-0.357016	0.262581	1.000000	0.204035	-0.226129	-0.010158	0.079699	-0.018845	0.023680	-0.108246	-0.191245
Rain	-0.365941	0.275592	0.204035	1.000000	-0.589465	-0.300364	-0.302591	-0.347660	-0.308258	-0.393221	-0.086938
FFMC	0.684556	-0.653649	-0.226129	-0.589465	1.000000	0.621958	0.528275	0.742079	0.606527	0.773444	0.267099
DMC	0.482965	-0.393893	-0.010158	-0.300364	0.621958	1.000000	0.884417	0.680918	0.984222	0.629505	0.175711
DC	0.349021	-0.203883	0.079699	-0.302591	0.528275	0.884417	1.000000	0.501412	0.951157	0.562431	-0.080660
ISI	0.618172	-0.712353	-0.018845	-0.347660	0.742079	0.680918	0.501412	1.000000	0.632285	0.747764	0.312299
BUI	0.447959	-0.333027	0.023680	-0.308258	0.606527	0.984222	0.951157	0.632285	1.000000	0.624037	0.079373
Classes	0.512299	-0.446906	-0.108246	-0.393221	0.773444	0.629505	0.562431	0.747764	0.624037	1.000000	0.201988
Region	0.296033	-0.427696	-0.191245	-0.086938	0.267099	0.175711	-0.080660	0.312299	0.079373	0.201988	1.000000

Figure 5.1: correlation table

Encoded not fire as 0 and fire as 1. And the dataset splitted to train and test Checking multicollinearity and removed the highly correlated features (figure 5.2) .

And applied feature scaling.The effect of feature scaling standardization (figure 5.3) can be understood from the box plot.

0=Temperature, 1 = RH, 2 = Ws, 3 = Rain, 4 = FFMC,5 = DMC, 6 = ISI, 7 = Region

CamScanner

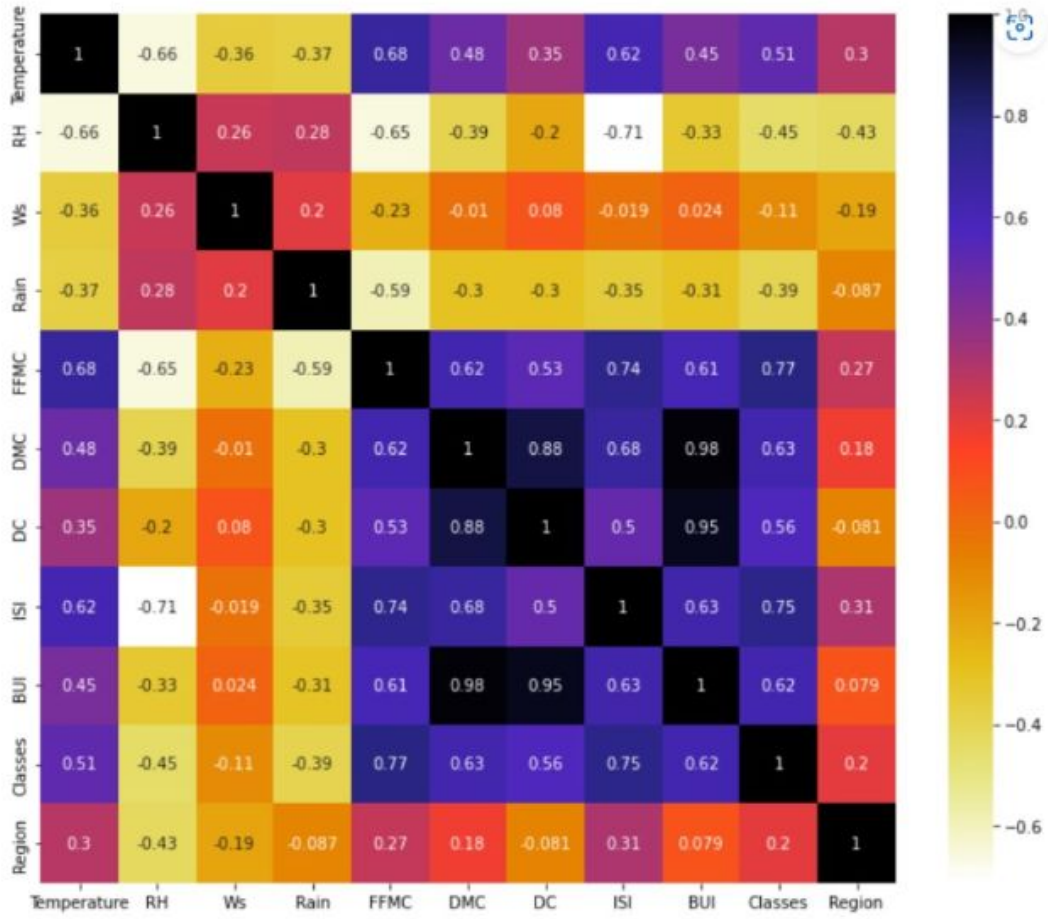


Figure 5.2: heatmap

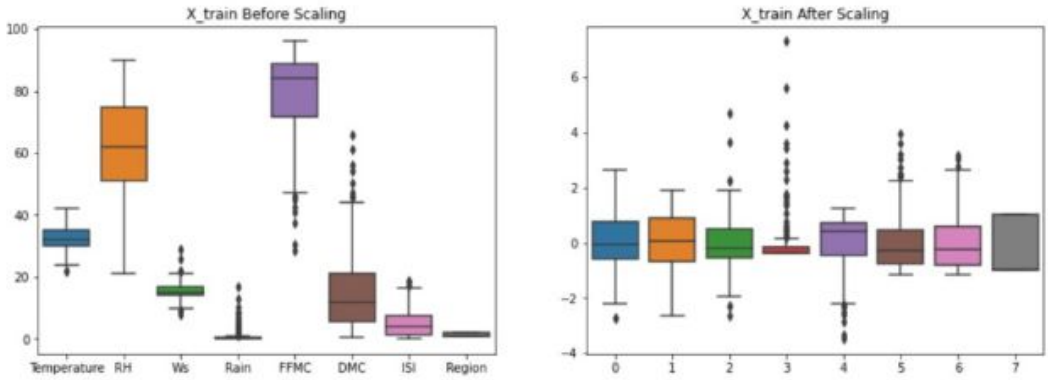


Figure 5.3: boxplot

CamScanner

5.2 Classification

Splitted the dataset into test and train data . Correlation is obtained using the Pearson Correlation and 5 classification models were build. Accuracy and confusion matrix of each model is obtained as below

CamScanner

5.2.1 Logistic Regression

	precision	recall	f1-score	support
0	0.92	0.96	0.94	25
1	0.98	0.96	0.97	48
accuracy			0.96	73
macro avg	0.95	0.96	0.95	73
weighted avg	0.96	0.96	0.96	73

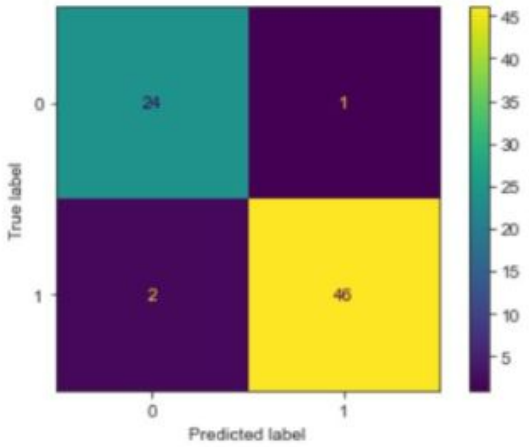


Figure 5.4: Confusion matrix

TP (True Positive)=24, FP(False Positive)=1, TN(True Negative)=46, FN(False Negative)=2

5.2.2 Decision Tree

	precision	recall	f1-score	support
0	0.93	1.00	0.96	25
1	1.00	0.96	0.98	48
accuracy			0.97	73
macro avg	0.96	0.98	0.97	73
weighted avg	0.97	0.97	0.97	73

CamScanner

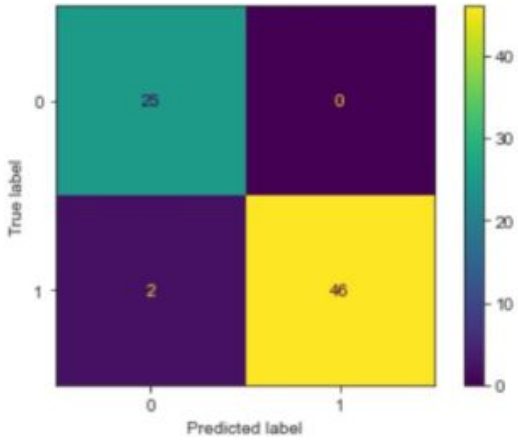


Figure 5.5: Confusion matrix

TP (True Positive)=25, FP(False Positive)=0, TN(True Negative)=46, FN(False Negative)=2

5.2.3 Random Forest

	precision	recall	f1-score	support
0	0.89	1.00	0.94	25
1	1.00	0.94	0.97	48
accuracy			0.96	73
macro avg	0.95	0.97	0.96	73
weighted avg	0.96	0.96	0.96	73

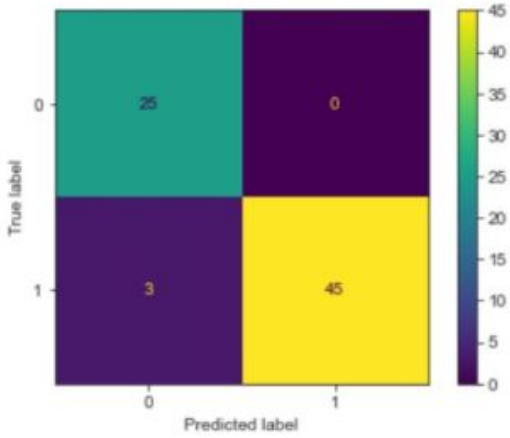


Figure 5.6: Confusion matrix

TP (True Positive)=25, FP(False Positive)=0, TN(True Negative)=45, FN(False Negative)=3

CamScanner

5.2.4 K-Nearest Neighbors Classifier

	precision	recall	f1-score	support
0	0.89	1.00	0.94	25
1	1.00	0.94	0.97	48
accuracy			0.96	73
macro avg	0.95	0.97	0.96	73
weighted avg	0.96	0.96	0.96	73

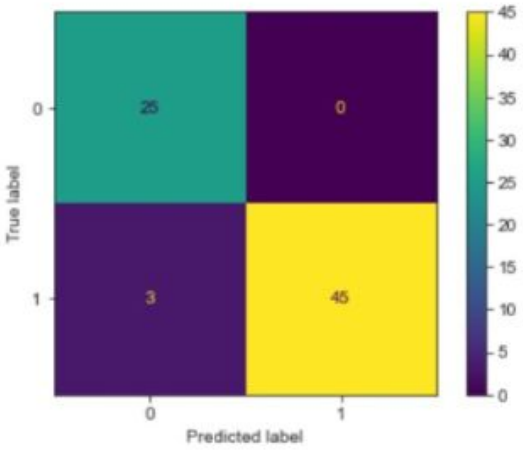


Figure 5.7: Confusion matrix

TP (True Positive)=25, FP(False Positive)=0, TN(True Negative)=45, FN(False Negative)=3

5.2.5 XGboost Model

	precision	recall	f1-score	support
0	0.93	1.00	0.96	25
1	1.00	0.96	0.98	48
accuracy			0.97	73
macro avg	0.96	0.98	0.97	73
weighted avg	0.97	0.97	0.97	73

CamScanner

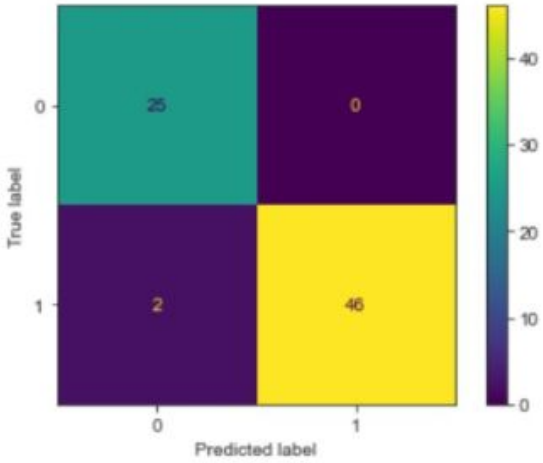


Figure 5.8: Confusion matrix

TP (True Positive)=25, FP(False Positive)=0, TN(True Negative)=46, FN(False Negative)=2

CamScanner

Chapter 6

RESULTS AND DISCUSSION

6.1 Regression

MODELS	R^2 *score*	MEAN ABSOLUTE ERROR
Linear Regression	0.9700	0.6453
Lasso Regression	0.9377	1.1209
Ridge Regression	0.9690	0.6648
Support Vector Regressor	0.9340	0.8676
Random Forest Regressor	0.9789	0.6036
K Nearest Neighbors Regressor	0.9422	0.9816

Figure 6.1: $R^2 and MAE$

Mean Absolute Error of each regression model is calculated and the least MAE value is for Random Forest Regressor and Linear Regression Ridge Regression models. From the R^2 score , accuracy is calculated using R^2 score * 100 and it given below;

Random Forest Model is found to be the best model. Highest accuracy is obtained for Random Forest model and Ridge and Linear Regression model. Hyperparameter tuning are done on Random Forest model and Ridge regression model using Random Search Hyperparameter tuning method. And the accuracy of Random Forest is obtained

CamScanner

REGRESSION	ACCURACY SCORE
Linear Regression	97.00%
Lasso Regression	93.77%
Ridge Regression	96..90%
Support Vector Regressor	93.40%
Random Forest Regressor	97.89%
K Nearest Neighbors Regressor	94.22%

Figure 6.2: Accuracy

as 98.03 %. and Ridge Regression is 96.97 %. ,R^2 Score value is 0.9803 and 0.9697 and MAE value is 0.5770 and 0.6504 The feature importance of the Random Forest classifier is saved inside the model itself, so all I need to do is to extract it and combine it with the raw feature names.

	feature	importance
6	ISI	0.712615
5	DMC	0.144719
4	FFMC	0.132900
2	Ws	0.004020
1	RH	0.003726
7	Region	0.000972
0	Temperature	0.000971
3	Rain	0.000077

Figure 6.3: feature importance

Most important 5 features were obtained (figure 6.2) and (figure 6.3)and created a Graphical User Interface (GUI) (figure 6.4) for predicting the unseen data.

CamScanner

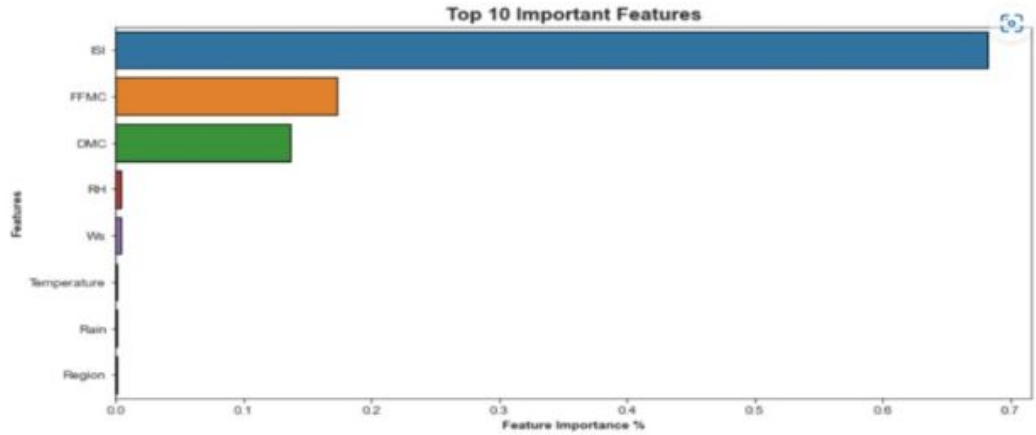


Figure 6.4: feature importance

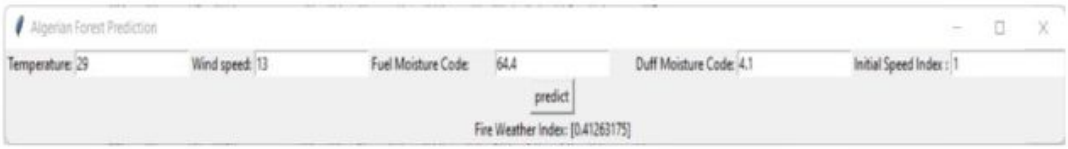


Figure 6.5: Graphical User Interface

6.2 Classification

Models	Accuracy score
XGboost classifier	97.26 %
Decision Tree Classifier	97.26 %
Logistic Regression Accurracy	95.89 %
KNeighbors Classifier	95.89 %
Random Forest Classifier	95.89 %

Figure 6.6: Accuracy

XGboost model has the highest accuracy amomg the 5 models. And hyperparamter tuning is applied on Xgboost model and Random Forest models and Cv score is used to obtain the accuracy of the models.

Final accuracy score is obtained as 0.9726 for XGboost classifier and 0.9726 for Random Forest classifier.

CamScanner

	precision	recall	f1-score	support
0	0.93	1.00	0.96	25
1	1.00	0.96	0.98	48
accuracy			0.97	73
macro avg	0.96	0.98	0.97	73
weighted avg	0.97	0.97	0.97	73

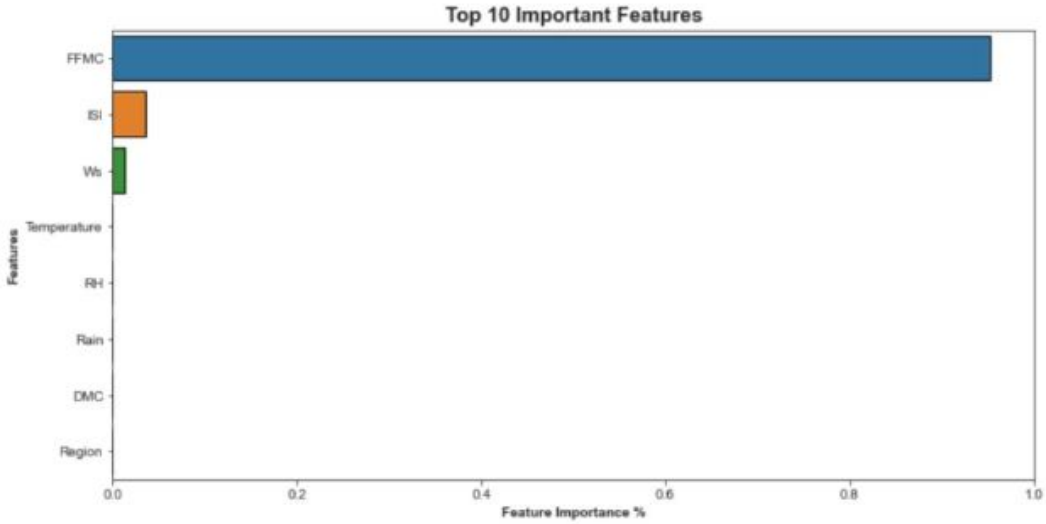
Figure 6.7: XGboost classifier

	precision	recall	f1-score	support
0	0.89	1.00	0.94	25
1	1.00	0.94	0.97	48
accuracy			0.96	73
macro avg	0.95	0.97	0.96	73
weighted avg	0.96	0.96	0.96	73

Figure 6.8: Random Forest Classifier

	feature	importance
4	FFMC	0.952777
6	ISI	0.034850
2	Ws	0.012373
0	Temperature	0.000000
1	RH	0.000000
3	Rain	0.000000
5	DMC	0.000000
7	Region	0.000000

CamScanner



important Features were obtained .Using most important 5 features created a Graphical User Interface to predict the whether its going to 'fire' or 'not fire'.

Figure 6.9: Graphical User Interface

CamScanner

Chapter 7

CONCLUSION

To conclude that all the objectives of the study are achieved. Prediction using the previously mentioned Machine Learning models done after the cross validation score.

Random Forest model is found to be the best among the 6 classification models. Predictions on Fire Weather Index (FWI) are made on the test data using the Random Forest model XGboost model is found to be the best among the 5 classification models. Predictions on whether the area is under 'fire' or 'not fire' are made on the test data using the XGboost model.

5 important features in both Regression and Classification models were obtained using feature removal method. And created a Graphical user interface for predicting ' FWI 'and 'FIRE' or 'NOT FIRE '.

Here , Regression model can be used to predict Fire Weather Index to calculate whether the area is prone to fire or not. And the Classificiation model can be used to give the alert to take the proper precautions to avoid the huge loss to forest fire.

In conclusion, forest fire prediction is an important area of research ,Machine learning algorithms and remote sensing techniques have been used to develop predictive models that can accurately forecast the occurrence, intensity, and spread of forest fires.

CamScanner

Chapter 8

REFERENCES

[1] S. Gayathri1 P.V. Ajay Karthi2 Sourav Sunil3. Prediction and Detection of Forest Fires based on Deep Learning Approach
DOI: 10.47750/pnr.2022.13.S03.071

[2] Chao Gao , Honglei and Haiqing Hu .Forest-Fire-Risk Prediction Based on Random Forest and Backpropagation Neural Network of Heihe Area in Heilongjiang Province, China.2023
https://www.mdpi.com/1999-4907/14/2/170

[3] Paulo Cortez1 and An´ıbal Morais1. A Data Mining Approach to Predict Forest Fires using Meteorological Data.
http://www.dsi.uminho.pt/œpcortez

[4] Madhurima De, 2Linika Labdhi, 3Bindu Garg . Predicting Forest Fires With Different Data Mining Techninques.International Journal of Scientific Development and Research (IJSDR).April 2020 IJSDR — Volume 5, Issue 4
http://www.ijsdr.org

[5] George E. Sakr, Imad H. Elhajj, George Mitri and Uchechukwu C. Wejinya "Artificial Intelligence for Forest Fire Prediction" 2010 IEEE/ASME

CamScanner

[6] **Adithi M. Shrouthy,Syed Matheen Pasha,Yamini S. R. E,Navya Shree S,Lisha U. "Forest fire prediction using ML and AI." International Journal of Advance Research, Ideas and Innovations in Technology.(2021)**

[7] **Faroudja Abid.A Survey of Machine Learning Algorithms Based Forest Fires Prediction and Detection Systems.2020** http://doi.org/10.1007/s10694-020-01056-z

[8] **Pratima Chaubey1 , Nidhi J. Yadav2 , Abhishek Chaurasiya3, Prof. Satish Ranbhise4 .Forest Fire Prediction System using Machine Learning ,2020. International Journal for Research in Applied Science Engineering Technology (IJRASET)** https://doi.org/10.22214/ijraset.2020.32546

[9] **A. Kansal, Y. Singh, N. Kumar and V. Mohindru, "Detection of forest fires using machine learning technique: A perspective" 2015 Third International Conference on Image Information Processing (ICIIP)**

[10] **L. Yu, N. Wang, and X. Meng "Real-time forest fire detection with Wireless Sensor Networks" in Wireless Communications, Networking and Mobile Computing, 2005.** https://www.academia.edu/2694428/Fire_Detection_using_Wireless_Sensor_Networks_An_Approach_Based_on_Statistical_Data_Modeling

[11] **Harita A , K Madhurekaa ,K P Neethu ,Kajol R Singh. "Predicting Forest Fires With Spark Machine Learning.VISVESVARAYA TECHNOLOGICAL UNIVERSITY(2020)** http://203.201.63.46:8080/jspui/bitstream/123456789/6185/1/PR3182%20-%20AC009-%20Predicting%20forest%20fires%20with%20Spark%20Machine%20Learning%20-.pdf

[12] **Mauro Castelli, Leonardo Vanneschi, and Ales Popovic "Predicting burned areas of forest fires: an artificial intelligence approach" Fire Ecology 2015:** https://fireecology.springeropen.com/articles/10.4996/fireecology.1101106

CamScanner