HIGH SCHOOL EDITION
Journal of Student Research

# A Machine Learning Approach to Predict the Occurrence of Forest Fires with Meteorological Parameters

Yashnil Mohanty

## ABSTRACT

Forest fires have emerged as a considerable challenge in the United States, posing substantial societal, economic, and environmental risks. As a result, the early and accurate prediction of these fires is imperative for management efforts. In this study, we used two Kaggle datasets: the “Algerian Forest Fires Dataset” with fire readings from 2012 and the “Forest Fires Data Set” with readings from 2007. However, because the second data set was originally intended for a regression task, providing approximate area values representing the predicted burned area of the forest fire, we phased the data set out while developing our final model. Ultimately, we used the Algerian Forest Fires Dataset, containing 13 attributes and 244 instances of forest fires in two regions of Algeria. To streamline the analysis, we reduced the number of features to 5, namely, month, temperature, humidity, wind, and rain. Moreover, we developed a Random Forest Classifier model to predict the occurrence of a forest fire, using the data set for training and testing. Performance was compared against Decision Tree, Logistic Regression, and Artificial Neural Network models, using cross-validation. The experiment showed a slight superiority to the Random Forest Classifier approach, achieving an accuracy score of 86.486% and an F1 score of 88.889%. Our approach provides a decimal value representing the probability for fire likelihood. Overarchingly, this research contributes to the advancement of forest fire prediction technologies by leveraging meteorological data.

## Introduction

Forest fires have become an imminent threat worldwide. Hotter and drier weather caused by climate change and poor land management create conditions favorable for high-intensity forest fires [1]. In the United States, there has been an annual average of 70,025 wildfires burning an annual average of 7 million acres since 2000 [2]. In recent years, this number has skyrocketed to unprecedented heights as the effects of climate change continue to persist. An increase in forest fires directly leads to an increase in global warming, in addition to many additional risks posed by forest fires, including threats to biodiversity, infrastructure, and health [3]. The ability to be able to accurately prevent the occurrence of a forest fire by taking real-time readings from weather stations could assist in helping mitigate this problem. By simply utilizing real-time meteorological readings from weather stations, this classifier model could ultimately lead to park officials taking early action for crowd control in case an area is at imminent risk of a fire. This could ultimately help assist in the early prevention of forest fires by determining whether the weather conditions are conducive to the occurrence of a fire. We decided to test our data on several different classifiers and run cross-validation on the results to ultimately determine a superior method for this approach.

### Literature Review

In this section, we review current approaches that we have encountered in the literature.
In the past, meteorological data has been incorporated into numerical indices, which are used for prevention and management strategies. For example, the Fire Weather Index System, which we further on about in the next section, was developed in the 1970s with simple mathematical calculations using only readings from four meteorological observations: temperature, relative humidity, rain, and wind. The FWI System has become a common method to test the occurrences of forest fires around the world [4].

We created a Random Forest Classifier method that determines the probability of forest fire occurrences through the votes of individual decision trees. Our model was trained on the Algerian Forest Fire Dataset with 244 instances and 13 parameters. However, through pre-processing, we narrowed this down to only 4 parameters. To contextualize our work within the developing field and assist with the creation of a robust machine-learning model, we surveyed several different relevant pieces of literature.

In this literature review, the focus is on exploring the existing knowledge and advancements in machine learning about forest fire management. The review aims to analyze the evolution and current state of machine learning models employed in addressing the occurrences of forest fires.

### *Abdelhamid Zaidi*

Abdelhamid Zaidi's research contributes to fire prediction using an Artificial Neural Network (ANN) architecture. Zaidi's study applies this ANN model to predict the occurrences of forest fires in Algeria. The Artificial Neural Network architecture consists of two hidden layers and his work focuses on the same target scope as us, the general prediction of forest fires. Zaidi utilizes the same dataset that we used, the Algerian Forest Fires Dataset, to train and test the ANN model for fire prediction. The dataset incorporates six variables as reduced by PCA. He achieves a 96.65% accuracy with his model. A key difference between our model and his is that he relied on the FWI Index Parameters, which we intentionally omitted due to the lack of readily available information on how to obtain the respective values. We additionally aimed to learn the relationship between standard meteorological data input and the likelihood of fire rather than using predefined weights, as provided by the FWI System. This makes our model more versatile since we can rely solely on weather parameters. Moreover, our model achieves this by using a Random Forest Classifier, which has a notably lower computational cost and time [5].

### *Mauro Castelli et. Al*

Mauro Castelli’s research contributes to fire prediction using an experimental artificial intelligence system called geometric semantic genetic programming (GSGP). Castelli’s approach applies this system to predict the burned areas of forest fires, suggesting a more regressive approach and task. Castelli and his colleagues used the Forest Fires Dataset, which happened to be the exact dataset that we originally used and phased out. This dataset was created from 517 instances of fires in Montesinho National Park in Portugal between 2000 and 2003 and precisely includes 12 different parameters. GSGP works by improvising a Genetic Programming (GP) model. GP is an evolutionary algorithm that operates using evolving models by modifying old ones through genetic operators. GSGP then introduces further geometric semantic operators that consider the semantics or meaning of the programs in addition to their syntax. Mauro Castelli’s work thus focuses on burned areas within fire prediction. He obtains a testing error of around 10-20 percent, though the exact number remains unspecified. Our model in comparison achieves a similar testing accuracy but serves a different purpose. Whereas Castelli’s original work aims to predict the burned area of forest fires, we aim to predict the occurrence of one through solely meteorological data. This makes our work unique [6].

## The Fire Weather Index System

Both of the datasets we used for this project incorporated the Fire Weather Index (FWI) System into their input readings (as attributes for the dataset). Based on a Canadian empirical model, FWI is one of the most widely used fire weather indices for measuring wildfire risk. The FWI System relies solely on weather readings that can be found in weather stations. It’s recursively calculated using yesterday’s data and current meteorological readings [4].

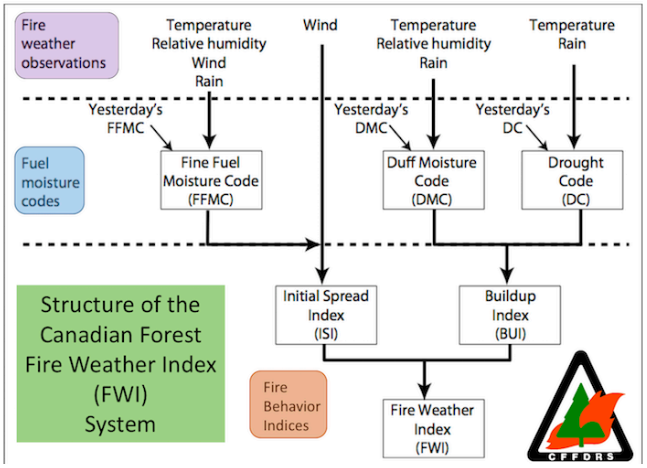


**Figure 1.** Structure of the Canadian Forest FWI System (NWCG 2023).

### Summary of Approach

To proceed with the development of this model, we first extracted a dataset from Kaggle. Initially, we downloaded two datasets with the hopes of creating a merged dataset with 761 instances. However, as we realized later, this was a suboptimal approach as one of the datasets we were using was meant for a regression task instead and the data was not organized in a way suitable to our model. So eventually, we completely phased out that dataset and decided to only proceed with one of our datasets. At the earlier stages of our model, we preprocessed both datasets to create a merged dataset with multiple features. Initially, we did not phase out the use of the FWI parameters, but later, as we tested out different variations and combinations of the datasets, some inclusive and some exclusive of the FWI parameters, we decided to pursue the variation that is presented in this research paper. After we decided on a version of the dataset, we ran PCA to optimize our training data and then used the train_test_split() function to test the data into training and testing data. We then ran the data through four different baseline models and selected the model with the best accuracy. We then tuned the hyperparameters for optimal cross-validation performance. To ultimately fulfill our goal of predicting the probability of a forest fire, We used the predict_proba() function to split the decision trees in the RFC model to give a resulting probability of the occurrence of a forest fire.

## Data Analysis

### Dataset

For this project, We used several different datasets. We downloaded two datasets on Kaggle, namely, Algerian Forest Fires Dataset and Forest Fires Data Set. We ultimately trained and tested the model on several variations of one or both of the datasets combined, to see which one would produce the most optimal and generalizable results.

### *Algerian Forest Fires Dataset*

This dataset includes 244 total instances of forest fires in Algeria in the year 2012. There are 13 input features and 1 output attribute, which is the classification of fire or not fire. This dataset incorporates all six of the features noted above, in addition to all four necessary standard attributes found in weather stations: temperature (Celsius), humidity (%), wind (km/h), and rain (mm). This dataset also included the day of the month, the month of the year, and the year, which was a constant of 2012. The output was a class, namely fire or not fire.

**Table 1**. Attributes of the Algerian Forest Fire Dataset

| No. | Attribute | Description |
|---|---|---|
| 1 | Day | Day of the Month |
| 2 | Month | Month of the Year (as a number) |
| 3 | Year | Year (2012) |
| 4 | Temperature | Temperature at noon (in Celsius) |
| 5 | Humidity | Relative Humidity (in %) |
| 6 | Wind | Wind Speed (in km/h) |
| 7 | Rain | Total day (in mm) |
| 8 | FFMC | Fine Fuel Moisture Code |
| 9 | DMC | Duff Moisture Code |
| 10 | DC | Drought Code |
| 11 | ISI | Initial Spread Index |
| 12 | BUI | Buildup Index |
| 13 | FWI | Fire Weather Index |
| 14 | **Target** | {fire: 1 \| fire: 0} |

### *Forest Fires Data Set*

This dataset includes 517 total instances of fires in Montesinho National Park in Portugal. There are 12 input features, including FFMC, DMC, DC, and ISI, as well as temperature, humidity, wind, and rain. In addition to that, the dataset also includes the x and y coordinates, month, and day of the week as features. The output attribute is the burned area of the respective fire instance. This dataset was meant to be a regression problem, as the output attribute is a burned area given the meteorological parameters, rather than a binary class of fire/not fire.
Note: The Forest Fires Dataset was modeled with ln(x+1) at the beginning since the distribution was so heavily skewed towards 0 (a logarithmic function can effectively compress extremely large data points). They eventually transformed the model's output back to its initial values by running it through the function e^x-1.

**Table 2**. Attributes of the Forest Fires Data Set

| No. | Attribute | Description |
|---|---|---|
| 1 | X | X-axis spatial coordinate in Montesinho National Park |
| 2 | Y | Y-axis spatial coordinate in Montesinho National Park |
| 3 | Month | Month of the Year ('jan' to 'dec') |
| 4 | Day | Day of the Week ('mon' to 'sun') |
| 5 | FFMC | Fine Fuel Moisture Code |
| 6 | DMC | Duff Moisture Code |
| 7 | DC | Drought Code |
| 8 | ISI | Initial Spread Index |
| 9 | Temperature | Temperature (in Celsius) |
| 10 | Humidity | Relative Humidity (in %) |
| 11 | Wind | Wind Speed (in km/h) |
| 12 | Rain | Total day (in mm) |
| 13 | **Target** | Total burned area of the forest fire (in hectares) |

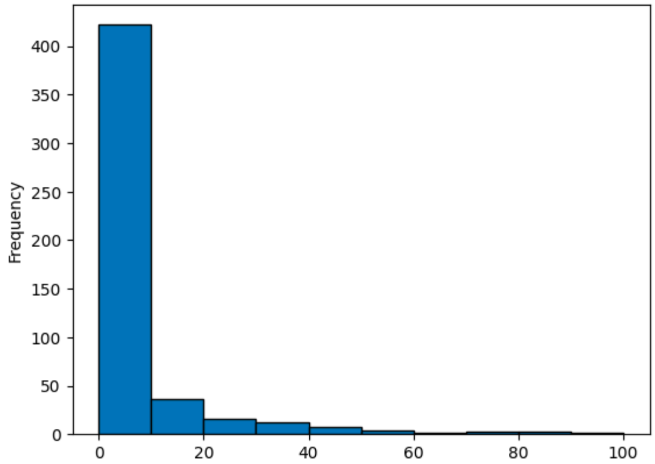


**Figure 2.** Frequency of Areas of Forest Fires in the Forest Fires Data Set

## *Inaccuracies and Notes*

While testing the data through multiple variations of both datasets, the results of the Portuguese dataset seemed to always perform much poorer than the results of the Algerian Data Set. The cause of the inaccuracy in the Portuguese data set seems to most likely be 2 things: (1) According to the EPA, a wildfire is defined as a fire larger than 1000 acres in the Western United States and 500 acres in the Eastern United States [7]. Even if we use the 500-acre threshold, only 4 of the 517 instances in the Portuguese data set meet the sufficient criteria for a 500-acre (202 hectares) fire. The rest of the instances where a fire took place were simply not large enough to be considered a wildfire. (2) Of the instances, already, the majority of them are equal to 0. By changing the bounds for a fire to a max of even 10 hectares (~25 acres), which wouldn't be considered a wildfire by any metric, nearly 450 of the instances are classified as "not fire," suggesting the data is extremely skewed towards 0, as represented in Figure 2. The main hitch with turning a regression data set into a classification one is that the data points/instances aren't exactly the best for wildfire prediction as the dataset was intended to mostly calculate the spread of a fire rather than the occurrence of one. This is why we phased out this dataset completely, as the data points for the Algerian data set more accurately represent that of a forest fire.

## Preprocessing

We ultimately decided to use the Algerian Forest Fires Dataset as the Forest Fires Dataset was more fitting for a regression problem. We omitted all the columns except temperature, humidity, wind, rain, and month. The reason for this is that we found little correlation between day and year with the occurrence of a fire. Furthermore, we intentionally omitted the FWI metrics because we were unable to find any significant information on how to calculate and obtain these metrics, meaning that for further use of this model, these would be largely insignificant. Therefore, we decided to only use the four main meteorological parameters and the month, as they would be easy to obtain and they were effective enough in training and testing this model.

The final data set we used had 5 features (columns) and 244 instances (rows). Figure 3 highlights the data distribution of each of our features.

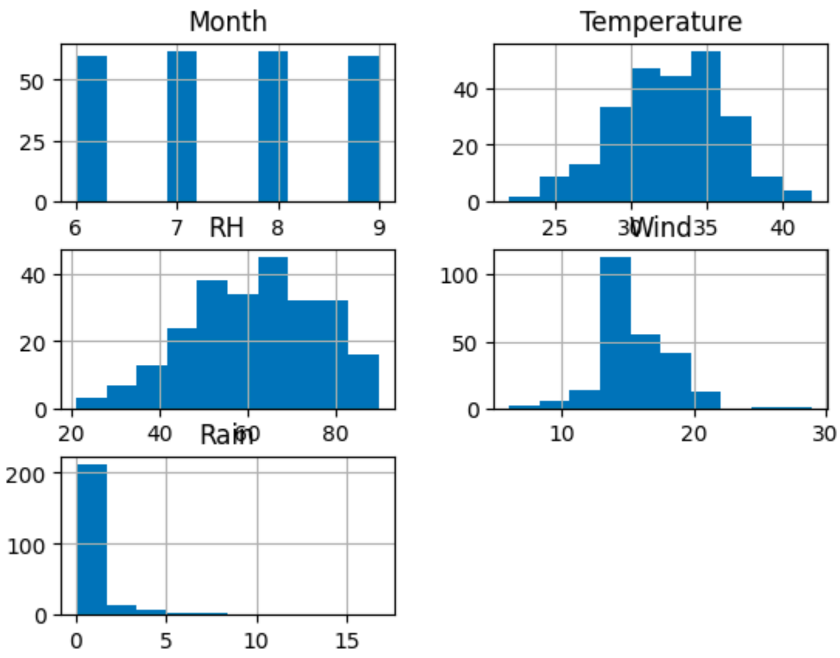


**Figure 3.** Histograms of Attributes

We used the train_test_split() function to break the data up into training and testing datasets. We used a 30% test size sample.

## Correlations and Running PCA

Principal Component Analysis (PCA) is a widely used statistical technique in dimensionality reduction. It is commonly used to simplify datasets with a high number of features while retaining essential information and reducing redundancy in the data. Reducing the number of variables typically comes at the expense of accuracy, but PCA generally effectively trades very minimal accuracy with data simplicity. It is often useful to perform PCA before building a classification model, as PCA can reduce the number of explanatory variables, which in turn reduces the computational demand of a model [8].

On our omitted data, we ran a heat map to test the correlation between different features, shown by Figure 4.

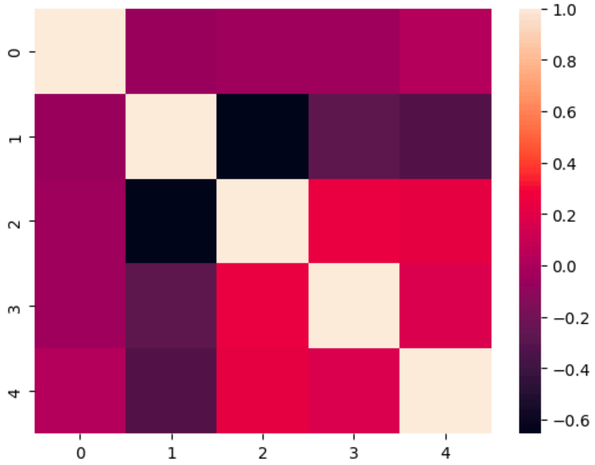


**Figure 4.** Correlation Between Attributes

We then ran the PCA algorithm with 99% variance with the scaled data, and the algorithm didn’t reduce the data frame at all. In other words, the PCA algorithm found a high correlation and usefulness in all features that we used. As a result, we decided to train and test the model on the preprocessed data set without modifying it any further.

# Models

## Decision Trees

Decision Trees are a popular machine-learning model. They are used to make decisions by recursively splitting the dataset into subsets based on the most significant attributes. Decision Trees start with a root question, referred to as the root node, which represents the entire dataset. The tree is then successively built by splitting the dataset into smaller subsets. Each internal node represents a “test” on the attribute (for instance, whether a coin flip comes up as heads or tails). This decision then divides the data into two or more child nodes. Each split allows the data to arrive at a conclusion, denoted by the leaf node. When splitting the data at each node, a splitting criterion is used to determine the feature and split point that best separates the data into subsets. Various metrics, including Gini impurity, information gain, or mean square error, can be used to evaluate the quality of the split [9].

## Random Forest Classifier

Random Forest (RF) is a commonly used machine learning algorithm, proposed by Leo Breiman in 2001. A Random Forest combines the output of multiple independent decision trees to reach a single result. Each tree has a partial vision of the problem due to random sampling:

- A random sampling on the observations with replacement (the rows of the dataset), known as tree bagging
  - This involves constructing n decision trees by taking a random sample of the dataset (with replacement), then training each of the n decision trees independently, and taking the majority of these n predictions as a result
- A random sampling on the variables (the columns of the dataset), known as feature sampling
  - This process involves randomly sampling several features from the given dataset. By default, we randomly sample $\lceil \sqrt{p} \rceil$ variables in a dataset with p total features. For a set of randomly selected

variables, a decision tree is created after tree bagging, which reduces correlation among distinct decision trees, which could alter the results [10].

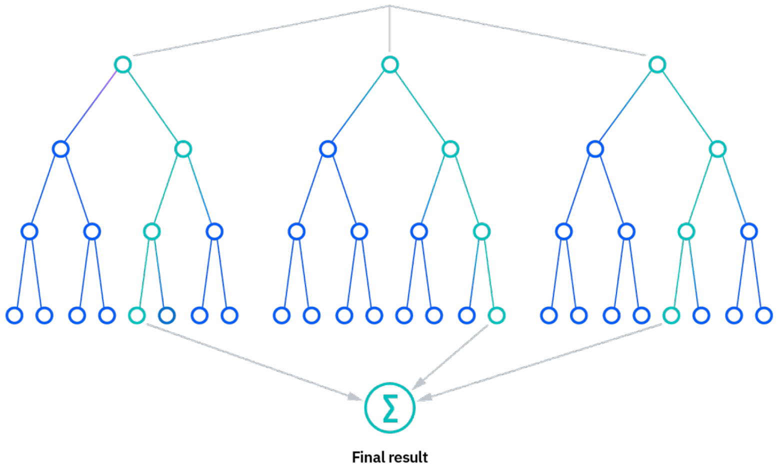


**Figure 5.** Random Forest Algorithm (IBM)

## Logistic Regression

Logistic regression (or a logit model) is a classification model that estimates the probability of an event occurring based on a linear combination of independent variables (features). The dependent variable (output) is a probability bounded between 0 and 1. In a logistic regression, a logit transformation is applied to the odds - which is a ratio of the probability of success and the probability of failure. This is commonly known as the log odds, represented by the following formulas:

$$\log\left(\frac{x}{1-x}\right) = \frac{1}{1+e^{-x}}$$

$$\ln\left(\frac{x}{1-x}\right) = b_0 + b_1x_1 + \ldots + b_kx_k$$

In this logistic regression equation, logit(pi) is the dependent variable and x is the independent variable. The beta parameter, or coefficient, in this model, is commonly estimated via maximum likelihood estimation. A logistic regression seeks to maximize this function to find the best parameters. Once the optimal coefficients are found, the conditional probabilities for each observation can be summed together to yield a final probability. For binary classification, this probability is typically rounded [11].

## Multilayer Perceptron (Artificial Neural Network)

An Artificial Neural Network is an interconnected system of neurons, also known as nodes, that process input to produce the desired output. Each of these neurons is connected by inputs, weights, and activation functions. ANNs are known for their learning capability, trained on known examples to then solve unknown problems. They learn through supervised or unsupervised learning: supervised involves known target values to minimize output errors, while unsupervised involves the network self-learning from data by detecting patterns or clustering similarities. A neural network typically comprises an input layer, hidden layers, and an output layer. Depending on the structure, ANNs are categorized as single-layer or multi-layer networks based on the presence of hidden layers [5].

Regarding some ANN parameters and hyperparameters:

- Weights: Links between neurons each carry a weight that holds input signal information. These weights often help calculate outputs. In a matrix with 'r' nodes and 'c' weights per node, denoted as W, the weight matrix takes the form shown in Figure 6.

$$W = \begin{bmatrix} W_{1,1} & .. & W_{1,c} \\ : & & : \\ W_{r,1} & .. & W_{r,c} \end{bmatrix}$$

**Figure 6.** Weight Matrix of an Artificial Neural Network (Zaidi 2023)

- Bias: The network incorporates bias by adding an extra input element, typically denoted as x0 = 1, into the input vector. This bias corresponds to a weight and helps determine an output. Positively biased values amplify the overall input weight, whereas negative biases diminish the net inputs.

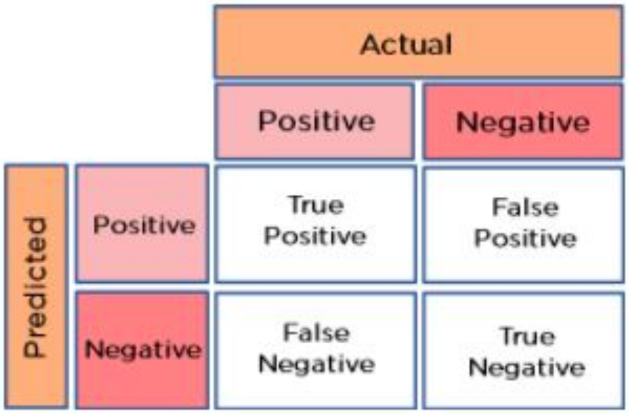
- Threshold: Determines output based on the following comparison:

$$\text{output} = \begin{cases} 1 & \text{if } \sigma(\text{net input}) \geq \text{threshold} \\ 0 & \text{if } \sigma(\text{net input}) < \text{threshold}^{\cdot} \end{cases}$$

## Evaluation of Prediction Models

To evaluate the performance of a model on a classifier data set, we consider the most common performance indices, namely, accuracy, precision, recall, and F-1 score. A confusion matrix is an evaluation grid of the accuracy of the prediction data for the test data. It provides a clear and detailed summary of how well a classifier is performing by comparing its predictions to the actual ground truth values in a dataset. It consists of four main components [12]:

- True Positives (TP): These are the cases where the model correctly predicted the positive class. In other words, these are the instances that are positive and were correctly classified as positive by the model.
- True Negatives (TN): These are the cases where the model correctly predicted the negative class. These are instances that are actually negative and were correctly classified as negative by the model.
- False Positives (FP): These are the cases where the model incorrectly predicted the positive class. These are instances that are negative but were incorrectly classified as positive by the model. False Positives are also known as Type I errors.
- False Negatives (FN): These are the cases where the model incorrectly predicted the negative class. These are instances that are positive but were incorrectly classified as negative by the model. False Negatives are also known as Type II errors.



| | | Actual | |
|---|---|---|---|
| | | Positive | Negative |
| Predicted | Positive | True Positive | False Positive |
| | Negative | False Negative | True Negative |

**Figure 7.** A Confusion Matrix (Simplilearn 2023)

Based on these components, various performance metrics can be calculated.

1. Accuracy: Accuracy measures the overall correctness of a classifier and is calculated by the formula:

$$A = \frac{T_P+T_N}{T_P+T_N+F_P+F_N}$$

2. Precision: Precision quantifies how many predicted positive instances were positive and is calculated by the formula:

$$P = \frac{T_P}{T_P+T_N}$$

3. Recall: Recall measures how many of the actual positive instances were properly predicted and is calculated by the formula:

$$R = \frac{T_P}{T_P+F_N}$$

4. F-1 Score: The F-1 Score is defined as the harmonic mean of the precision and the recall as a measure of the accuracy of a classifier, and is calculated by the formula:

$$F = \frac{2 \cdot P \cdot R}{P+R}$$

## Experimental Results

Our initial objective was to develop a machine-learning model that could effectively predict the occurrence of a forest fire with only meteorological parameters. Even though we decided to ultimately phase out all work done with the Portuguese data set for reasons we mentioned earlier, we will still show our experimental results of all models run involving that data set. The low accuracies shown in Table 3 have been explained in further detail earlier in this paper.

**Table 3**. Accuracy of Different Variations of our Dataset on a Random Forest Classifier (Note: An Advanced Feature is defined as data derived from the FWI System)

| Variation of Model | Accuracy |
|---|---|
| Portugal Dataset with Advanced Features | 0.564102564102564 |
| Portugal Dataset without Advanced Features | 0.544871794871795 |
| Merged Dataset with Advanced Features | 0.689956331877729 |
| Merged Dataset without Advanced Features | 0.712418300653595 |

We ran our models on the cropped Algerian Forest Fire Dataset with only the four meteorological parameters and month of the year (as an integer). A random forest classifier model outperformed our other models for this approach. We achieved an initial baseline accuracy score of 86.486% with the Random Forest Classifier Model. For a Logistic Regression approach, we achieved an accuracy of 72.973%. Meanwhile, for a Decision Tree Classifier approach, we achieved an accuracy of 81.081% and for an Artificial Neural Network without tuning the hyperparameters, we got a 78.378% accuracy.

**Table 4**. Accuracy of Different Models on our Finalized Dataset

| Model | Accuracy |
|---|---|
| Random Forest Classifier | 0.864864864864865 |
| Logistic Regression | 0.72972972972973 |
| Decision Tree Classifier | 0.810810810810811 |

| | |
|---|---|
| Artificial Neural Network | 0.783783783783784 |

Because we achieved our highest initial accuracy with an untuned Random Forest Classifier Model, we decided to pursue an RFC when creating our final model. We used validation curves while hyperparameter tuning to test for the optimal parameters.

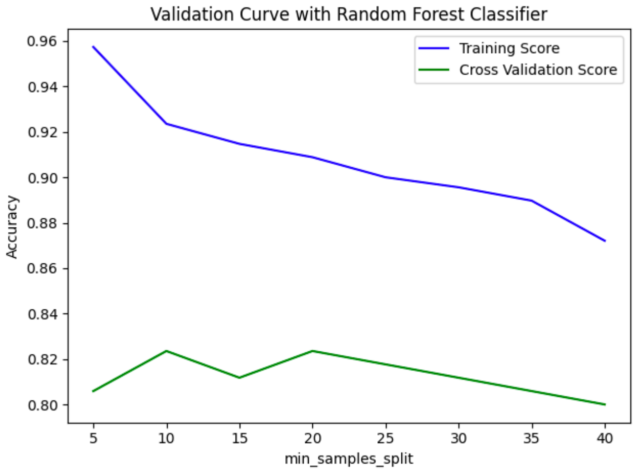


**Figure 8.** Validation Curve with Random Forest Classifier for min_samples_split

Figure 8 shows that 'min_samples_split = 20' would be the ideal value for min_samples_split. As min_samples_split gets larger from here, both the accuracy of the training score and cross-validation score decrease.

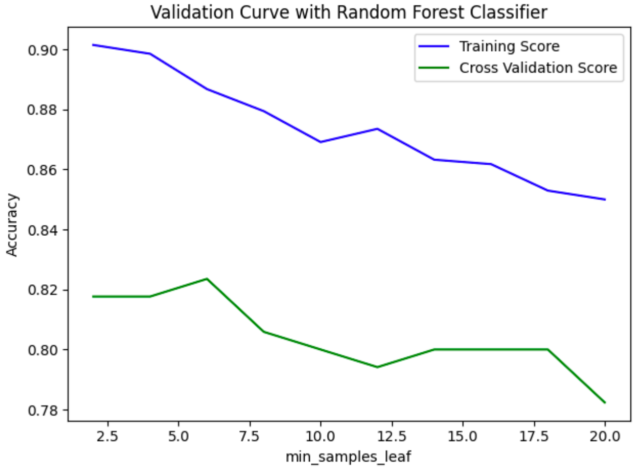


**Figure 9.** Validation Curve with Random Forest Classifier for min_samples_leaf

Moreover, Figure 9 shows us that a 'min_samples_leaf' value of 6 is optimal for this model since we achieve a peak Cross-Validation Score. Increasing the min_samples_leaf parameter from here only decreases the accuracy of the Training Score and the Cross Validation Score. We didn't run a cross-validation curve on any additional hyperparameters and thus used this version of the model as our final. The resulting training accuracy of the model with these tuned hyperparameters is 86.486, which is the same percentage as if it were without parameters. The reason for this is that the hyperparameters we found were chosen through cross-validation, which is a mechanism used to decrease overfitting rather than improve training accuracy. The confusion matrix on the performance of our model is shown below in Figure 10, and the respective ROC curve is shown in Figure 11.

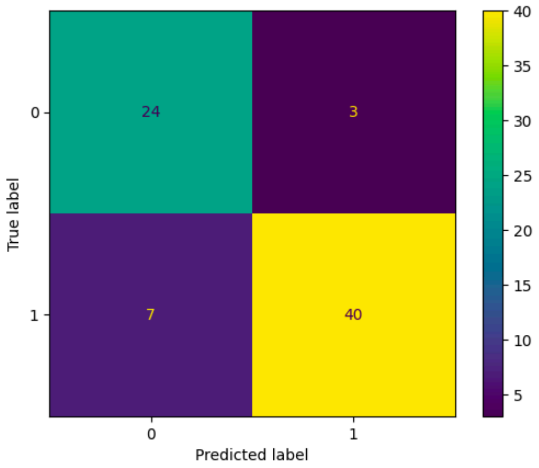


**Figure 10.** Confusion Matrix on Random Forest Classifier

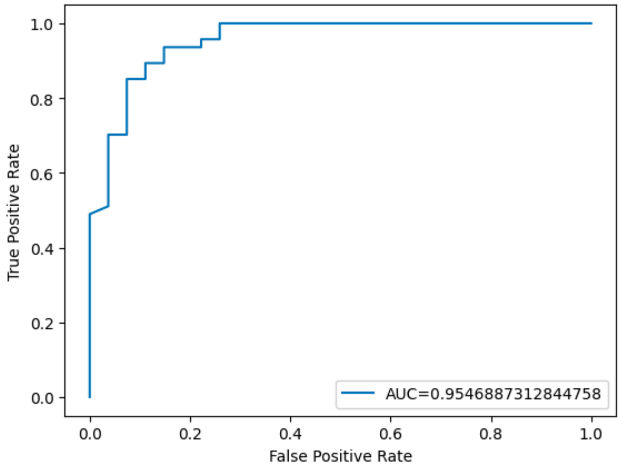


**Figure 11.** ROC Curve on Random Forest Classifier

The graph allows us to confirm that we indeed have a model largely free of overfitting. Table 5 shows the results of our performance metrics on the Random Forest Classifier.

**Table 5**. Random Forest Classifier vs. Different Performance Metrics

| | |
|---|---|
| Accuracy Score | 0.8648648648648865 |
| F1 Score | 0.888888888888889 |
| Precision Score | 0.9302325581395535 |
| Recall Score | 0.851063829787234 |

## Conclusion

In this study, we trained a Random Forest Classifier (RFC) model that can accurately predict the occurrence of a forest fire with only four meteorological parameters. We trained and compared the performance of our Classifier with an ANN, Decision Tree, and Logistic Regression classifier, while also using cross-validation. The experiment shows a slight superiority of the RFC over the other models. We achieve an accuracy of about 86.486% and an F1 score of approximately 88.889%. Moreover, we have shown that the FWI Index is not a viable method for a forest fire prediction model due to its lack of information available. The results of this model could be used to assist forest management institutions by providing a mechanism to accurately prevent the occurrences of forest fires and encourage early action. Since we provide a probability of fire occurrence, fire management services can use this data to rank the severity of wildfires and more effectively manage areas at risk given resources.

## Future Work

Building upon the insights and outcomes derived from the development of our Random Forest Classifier for forest fire prediction, several avenues for future research and enhancement present themselves, offering opportunities to further advance and refine the predictive model. For example, our model notably lacks versatility, as we limited our training data to only 244 instances from Algeria. If we had a larger dataset, we could have possibly achieved a more accurate model, and hopefully, on further research, it would be possible to create a more generalizable model with occurrences from a more diverse sample. There may be several inherent flaws with where we derived our training sample from, as several other factors could play a role in the occurrence of these fires. Therefore, a more comprehensive approach with more instances around the world would help broaden the scope of this model. We pondered this problem when creating this model, but we failed to pursue this or create a new dataset for our likings. Moreover, our model may be overfitting as our training data is very limited. For this to change, more research would have to be done on this topic.

## Acknowledgements

I would like to thank my mentor Juyon Lee for giving me valuable advice and helping me throughout my journey in creating this project and writing this research paper.

## References

1. Staff. “Forest Fires & Climate Change: Effects of Deforestation on Wildfires: GFW.” *Global Forest Watch*, 2024, www.globalforestwatch.org/topics/fires/#intro.
2. Staff. “Wildfire Statistics.” *Congressional Research Service*, 1 June 2023, sgp.fas.org/crs/misc/IF10244.pdf.
3. Keeley, Jon, and Alexandra Syphard. “Climate change and future fire regimes: Examples from California.” *Geosciences*, vol. 6, no. 3, 17 Aug. 2016, p. 37, https://doi.org/10.3390/geosciences6030037.
4. Staff. “Fire Weather Index (FWI) System.” *NWCG*, 28 Aug. 2023, www.nwcg.gov/publications/pms437/cffdrs/fire-weather-index-system.
5. Zaidi, Abdelhamid. “Predicting wildfires in Algerian forests using machine learning models.” *Heliyon*, vol. 9, no. 7, 10 July 2023, https://doi.org/10.1016/j.heliyon.2023.e18064.
6. Castelli, Mauro, et al. “Predicting burned areas of forest fires: An Artificial Intelligence Approach.” *Fire Ecology*, vol. 11, no. 1, 1 Apr. 2015, pp. 106–118, https://doi.org/10.4996/fireecology.1101106.
7. Staff. “Wildfires.” *EPA*, Aug. 2016, www.epa.gov/sites/default/files/2016-08/documents/print_wildfires-2016.pdf.
8. Jaadi, Zakaria. “A Step-by-Step Explanation of Principal Component Analysis (PCA).” *Built-In*, 29 Mar. 2023, builtin.com/data-science/step-step-explanation-principal-component-analysis.
9. Staff. “What Is a Decision Tree.” *IBM*, www.ibm.com/topics/decision-trees#:~:text=A%20decision%20tree%20is%20a,internal%20nodes%20and%20leaf%20nodes. Accessed 17 Feb. 2024.
10. Staff. “What Is Random Forest?” *IBM*, www.ibm.com/topics/random-forest. Accessed 17 Feb. 2024.
11. Staff. “What Is Logistic Regression?” *IBM*, www.ibm.com/topics/logistic-regression. Accessed 17 Feb. 2024.
12. Staff. “What Is a Confusion Matrix in Machine Learning?” *Simplilearn.Com*, Simplilearn, 16 Feb. 2023, www.simplilearn.com/tutorials/machine-learning-tutorial/confusion-matrix-machine-learning.