# Fire Detection using Machine Learning

Kothawale Chetan(M24IRM003) Priyabrata Saha(M24IRM004)
Vishal Kumar(M24IRM008) Vishal Raathi(M24IRM009)

Indian Institute of Technology Jodhpur

November 10, 2024

**Abstract**

Forests are an important source of many natural resources. Forest fires are thus a major threat to plants, animals and humans. In this project we have tried to detect forest fires by applying Machine Learning models to available sensor data. We also compared various Machine Learning techniques used by researchers.
**Keywords:** Forest fire, Machine Learning

## Contents

**1 Introduction** **1**

**2 Methodology** **2**
2.1 Data Collection . . . 2
2.2 Data Pre-processing . . . 2
2.3 Data Analysis . . . 2
2.4 Splitting data into Training, Validation and Testing sets . . . 2
2.5 Applying different Machine Learning models . . . 2
2.6 Comparing metrics of above models . . . 2

**3 Dataset Explanation** **3**

**4 Implementation** **3**
4.1 Data Pre-processing and Analysis . . . 3
4.2 Applying ML Models and Observations . . . 4
4.2.1 K Nearest Neighbor . . . 4
4.2.2 Decision Tree . . . 5
4.2.3 Support Vector Machine . . . 5
4.2.4 Logistic Regression . . . 5
4.2.5 Naive Bayes . . . 6
4.2.6 Neural Network . . . 6

**5 Comparison** **7**

**6 Conclusion** **7**

**7 References** **7**

# 1 Introduction

Forest Fires are one of the major disasters in today's world. It affects both flora and fauna residing in the forest. Loss of natural resources and emission of harmful gases affects human life also. Thus detecting forest fires at initial stages is an important research problem. Machine Learning techniques can be useful in this purpose. Hence as a part of the course project of Fundamentals of Machine Learning course we are implementing various ML techniques to detect the presence of forest fire using sensor data from field.

# 2 Methodology

## 2.1 Data Collection

We use the data from UCI Machine Learning Repository
https://archive.ics.uci.edu/ml/datasets/Forest+Fires

## 2.2 Data Pre-processing

We mapped the months and days with numerical values and made the output binary.

## 2.3 Data Analysis

We calculated the correlation matrix and figured out that output is mostly correlated to month.We also normalized the input data.

## 2.4 Splitting data into Training, Validation and Testing sets

We split the data set into 80% training data, 10% validation data and 10% testing data.

## 2.5 Applying different Machine Learning models

We applied the following Machine Learning models:

- K Nearest Neighbors
- Decision Tree
- Support Vector Machine
- Logistic Regression
- Naive's Bayes Classifier
- Neural Network

## 2.6 Comparing metrics of above models

We compared the above models using metrics - Accuracy, Precision, Recall, Confusion matrix.

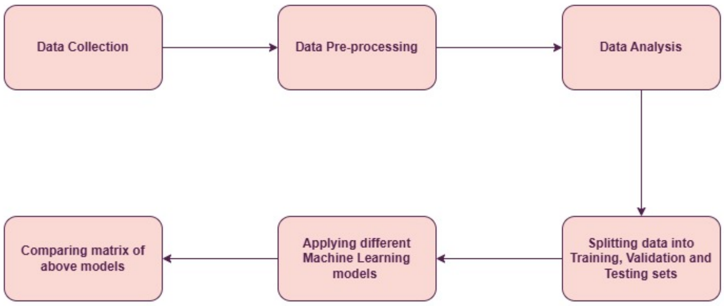



Figure 1: Methodology

# 3 Dataset Explanation

The dataset we used is shown below -

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

Figure 2: Dataset

The dataset has 517 rows and 12 columns. The description of various columns is shown is given below

1. **X** - x-axis spatial coordinate within the Montesinho park map: 1 to 9
2. **Y** - y-axis spatial coordinate within the Montesinho park map: 2 to 9
3. **month** - month of the year: 'jan' to 'dec'
4. **day** - day of the week: 'mon' to 'sun'
5. **FFMC** - FFMC index from the FWI system: 18.7 to 96.20
6. **DMC** - DMC index from the FWI system: 1.1 to 291.3
7. **DC** - DC index from the FWI system: 7.9 to 860.6
8. **ISI** - ISI index from the FWI system: 0.0 to 56.10
9. **temp** - temperature in Celsius degrees: 2.2 to 33.30
10. **RH** - relative humidity in
11. **wind** - wind speed in km/h: 0.40 to 9.40
12. **rain**- outside rain in mm/m2 : 0.0 to 6.4
13. **area** - the burned area of the forest (in ha): 0.00 to 1090.84

Detailed meaning of some FFMC, DMC, DC and ISI index is given below -

1. **FFMC (Fine Fuel Moisture)** helps predict how easily a fire could ignite in fine, easily combustible materials.
2. **DMC (Duff Moisture)** indicates how dry deeper organic material is and can help predict the persistence of fires underground.
3. **DC (Drought Code)** looks at long-term dryness, indicating how severe drought conditions have affected deeper soil and forest material.
4. **ISI (Initial Spread Index)** is an estimate of how fast a fire might spread, based on the combined effect of fuel moisture and wind conditions.

# 4 Implementation

## 4.1 Data Pre-processing and Analysis

As a part of Data Pre-processing and Analysis we performed the following steps -

1. We mapped the **day** column to the therespective day number of the week. eg: mon - 1, sun - 7
2. We mapped the **month** column to the respective month number of the year. eg: jan - 1, dec - 12
3. We converted **area** column to **binary data** i.e. if area burnt is greater than 0 we replaced it by 1 else 0.
4. We calculated the correlation matrix of dataset which is shown in Fig 3. We found that output has highest correlation with month.
5. We selected area as the output data (y) and rest columns as input data (X).
6. We normalized the input data.
7. Then we split the data set into 80% training data, 10% validation data and 10% testing data

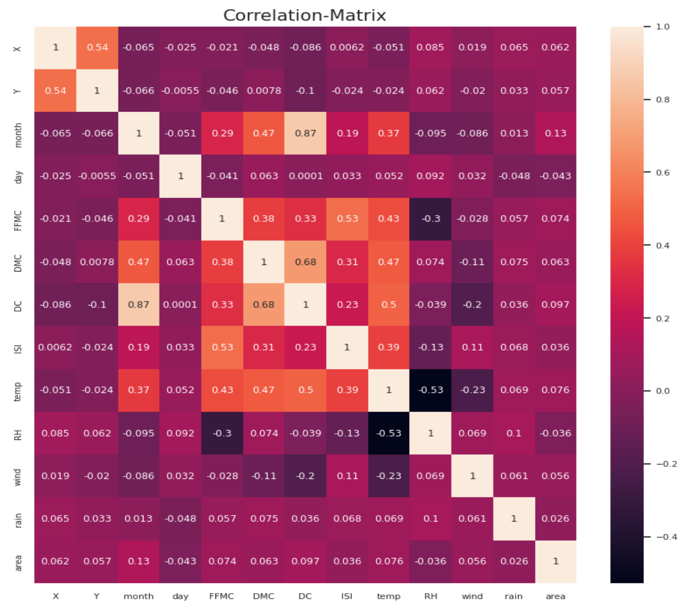



Figure 3: Correlation Matrix

## 4.2 Applying ML Models and Observations

### 4.2.1 K Nearest Neighbor

For various values of K in K nearest neighbor model, we trained and predicted the output among which we found that for value of k as 7 (refer Fig 4), we are getting the minimum error. Hence we used this value in test set and made the classification report and confusion matrix which are shown in Fig 5.

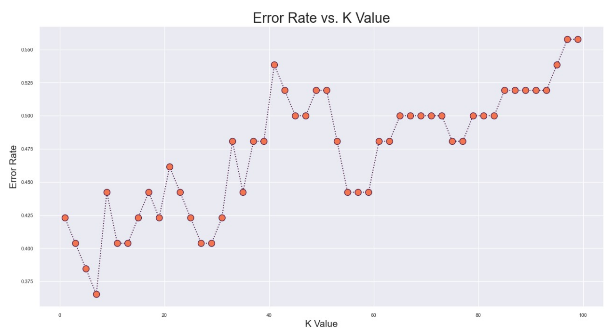
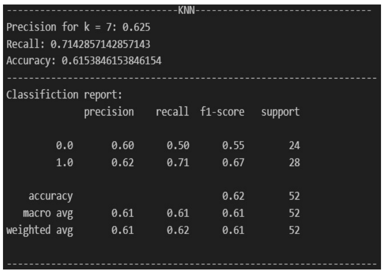
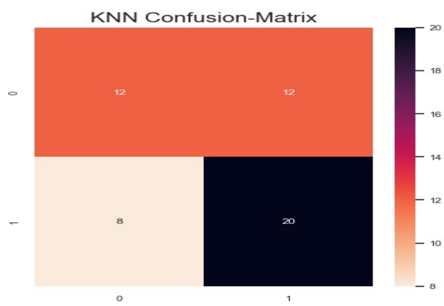



Figure 4: KNN Error vs K values



```
------------------------------KNN------------------------------
Precision for k = 7: 0.625
Recall: 0.7142857142857143
Accuracy: 0.6153846153846154
---------------------------------------------------------------
Classifiction report:
              precision    recall  f1-score   support

         0.0       0.60      0.50      0.55        24
         1.0       0.62      0.71      0.67        28

    accuracy                           0.62        52
   macro avg       0.61      0.61      0.61        52
weighted avg       0.61      0.62      0.61        52

---------------------------------------------------------------
```

(a)




(b)

Figure 5: (a) KNN Classification report (b) KNN Confusion Matrix

### 4.2.2 Decision Tree

We trained the model using in built libraries, plotted the decision tree(refer Fig 6) and predicted the output for the test set.The classification report and confusion matrix are shown in Fig 7.

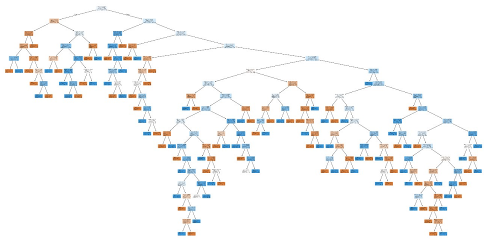
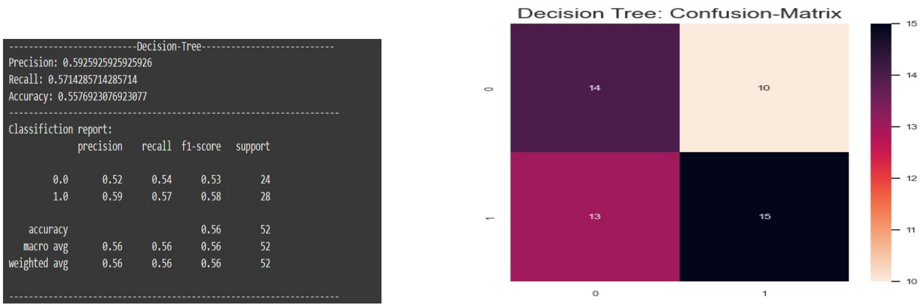


Figure 6: Decision Tree Plot



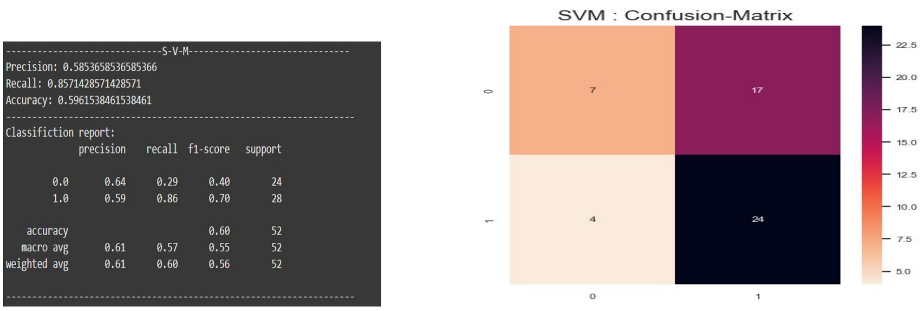

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0.0 | 0.52 | 0.54 | 0.53 | 24 |
| 1.0 | 0.59 | 0.57 | 0.58 | 28 |
| accuracy | | | 0.56 | 52 |
| macro avg | 0.56 | 0.56 | 0.56 | 52 |
| weighted avg | 0.56 | 0.56 | 0.56 | 52 |

(a) (b)

Figure 7: (a) Decision Tree Classification report (b) Decision Tree Confusion Matrix

### 4.2.3 Support Vector Machine

We trained the model using in built libraries, predicted the output for the test set.The classification report and confusion matrix are shown in Fig 8



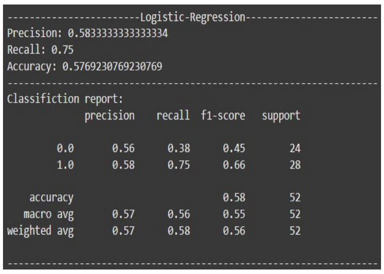

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0.0 | 0.64 | 0.29 | 0.40 | 24 |
| 1.0 | 0.59 | 0.86 | 0.70 | 28 |
| accuracy | | | 0.60 | 52 |
| macro avg | 0.61 | 0.57 | 0.55 | 52 |
| weighted avg | 0.61 | 0.60 | 0.56 | 52 |

(a) (b)

Figure 8: (a) Support Vector Machine Classification report (b) Support Vector Machine Confusion Matrix

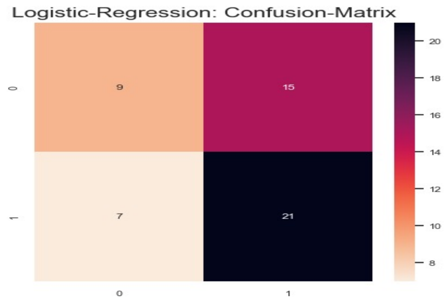
### 4.2.4 Logistic Regression

We trained the model using in built libraries, predicted the output for the test set.The classification report and confusion matrix are shown in Fig 9.



```
-----------------------Logistic-Regression-----------------------
Precision: 0.5833333333333334
Recall: 0.75
Accuracy: 0.5769230769230769
-----------------------------------------------------------------
Classifiction report:
              precision    recall  f1-score   support

         0.0       0.56      0.38      0.45        24
         1.0       0.58      0.75      0.66        28

    accuracy                           0.58        52
   macro avg       0.57      0.56      0.55        52
weighted avg       0.57      0.58      0.56        52

-----------------------------------------------------------------
```

(a)

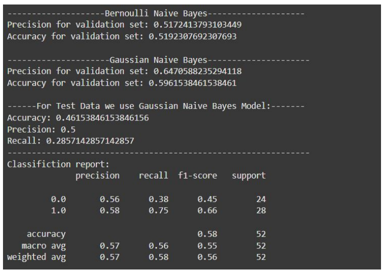



(b)

Figure 9: (a) Logistic Regression Classification report (b) Logistic Regression Confusion Matrix

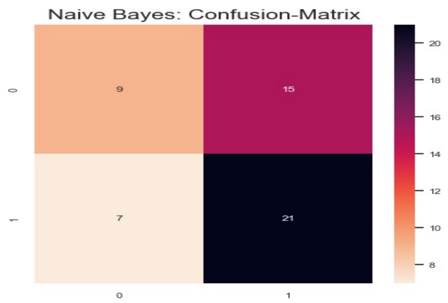
#### 4.2.5 Naive Bayes

We trained Bernoulli Naive Bayes and Gaussian Naive Bayes model For train data set. Using Validation data we found that Gaussian Bayes theorem is giving higher accuracy hence we used this model for prediction. We then predicted the output for the test data set. The classification report and confusion matrix are shown in Fig 10.



```
-------------------Bernoulli Naive Bayes-------------------
Precision for validation set: 0.5172413793103449
Accuracy for validation set: 0.5192307692307693

--------------------Gaussian Naive Bayes--------------------
Precision for validation set: 0.6470588235294118
Accuracy for validation set: 0.5961538461538461

------For Test Data we use Gaussian Naive Bayes Model:-------
Accuracy: 0.46153846153846156
Precision: 0.5
Recall: 0.2857142857142857
------------------------------------------------------------
Classifiction report:
              precision    recall  f1-score   support

         0.0       0.56      0.38      0.45        24
         1.0       0.58      0.75      0.66        28

    accuracy                           0.58        52
   macro avg       0.57      0.56      0.55        52
weighted avg       0.57      0.58      0.56        52
```

(a)




(b)

Figure 10: (a) Naive Bayes Classification report (b) Naive Bayes Confusion Matrix

#### 4.2.6 Neural Network

We trained the model using pytorch library. Our model architecture consists of 1 input layer with 12 nodes, 1 hidden layer with 12 nodes and 1 output layer with 1 node. We used ReLU and Sigmoid activation functions in layer 1 and 2 respectively. Then we used Binary Cross entropy(BCE) loss function, learning rate as 0.001. Number of training epochs and batch size was 300 and 10 respectively. We used this model to predict the test set, the classification report and confusion matrix of which are attached in Fig 12. Also the plot of BCE loss versus number of epochs is shown in Fig 11.

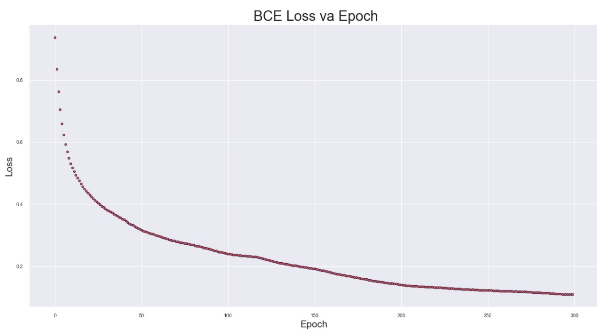
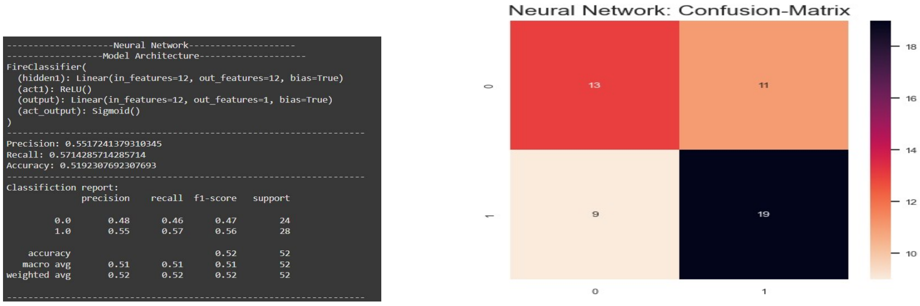



Figure 11: BCE loss vs epochs



```
-------------------Neural Network-------------------
-----------------Model Architecture-------------------
FireClassifier(
  (hidden1): Linear(in_features=12, out_features=12, bias=True)
  (act1): ReLU()
  (output): Linear(in_features=12, out_features=1, bias=True)
  (act_output): Sigmoid()
)
-------------------------------------------------------------------
Precision: 0.5517241379310345
Recall: 0.5714285714285714
Accuracy: 0.5192307692307693
-------------------------------------------------------------------
Classifiction report:
              precision    recall  f1-score   support

         0.0       0.48      0.46      0.47        24
         1.0       0.55      0.57      0.56        28

    accuracy                           0.52        52
   macro avg       0.51      0.51      0.51        52
weighted avg       0.52      0.52      0.52        52

-------------------------------------------------------------------
```



(a) (b)

Figure 12: (a) Neural Network Classification report (b) Neural Network Confusion Matrix

## 5 Comparison

Thus from the observations of the metrics of the implemented models KNN is giving the best accuracy (refer Fig 13).

| Models | Accuracy | Precision | Recall |
|---|---|---|---|
| K-NN | 0.62 | 0.63 | 0.71 |
| Decision Tree | 0.58 | 0.54 | 0.6 |
| SVM | 0.6 | 0.58 | 0.75 |
| Logistic Regression | 0.58 | 0.58 | 0.75 |
| Naive's Bayes | 0.46 | 0.5 | 0.28 |
| Neural Network | 0.6 | 0.62 | 0.64 |

Figure 13: Comparison

## 6 Conclusion

In this project we tried to predict the presence of forest fire using available sensor data. We learned and implemented different Machine Learning models for this task and found the best model for this scenario which came out to be K Nearest Neighbor model (KNN).

## 7 References

1. Sathishkumar, V. E., Cho, J., Subramanian, M., Naren, O. S. (2023). Forest fire and smoke detection using deep learning-based learning without forgetting. Fire Ecology, 19(1).
https://doi.org/10.1186/s42408-022-00165-0
2. Mohri, M., Rostamizadeh, A., Talwalkar, A. (2012). Foundations of machine learning.
http://ci.nii.ac.jp/ncid/BB10313921
3. UCI Machine Learning Repository. (n.d.).
https://archive.ics.uci.edu/ml/datasets/Forest+Fires