**Project Report**
**on**

# FIRE DETECTION SYSTEM USING MACHINE LEARNING ALGORITHMS

**Submitted as partial fulfillment for the award of**

# BACHELOR OF TECHNOLOGY

# DEGREE

**Session 2021-22**

**in**

## Information Technology


**By:**

**Prakash Singh 1803213112**

**Harshit Singh 1803213066**

**Pragati Gupta 1803213111**

**Krishna Vishwakarma 1803213084**

**Under the guidance of:**

**Dr. Amit Sinha**
**Head Of Department, IT**

**DEPARTMENT OF INFORMATION TECHNOLOGY**
**ABES ENGINEERING COLLEGE, GHAZIABAD**


ABES
Estd. 2000

NATIONAL BOARD OF ACCREDITATION

**AFFILIATED TO**
**DR. A.P.J. ABDUL KALAM TECHNICAL UNIVERSITY, U.P., LUCKNOW**
**(Formerly UPTU)**

# Student’s Declaration

We hereby declare that the work being presented in this report entitled **“Fire Detection System using Machine Learning Algorithms”** is an authentic record of our own work carried out under the supervision of **Dr. Amit Sinha, Head of Department, Information Technology.**

The matter embodied in this report has not been submitted by us for the award of any other degree.

**Date: 10th November 2021**

**Signature of student**

**(Name: Prakash Singh)**

**(Roll No. 1803213112)**

**Department: Information Technology**

**Signature of student**

**(Name: Harshit Singh)**

**(Roll No. 1803213066)**

**Department: Information Technology**

**Signature of student**

**(Name: Pragati Gupta)**

**(Roll No. 1803213111)**

**Department: Information Technology**

**Signature of student**

**(Name: Krishna Vishwakarma)**

**(Roll No. 1803213084)**

**Department: Information Technology**

This is to certify that the above statement made by the candidate(s) is correct to the best of my knowledge.

**Signature of HOD**

**(Name: Dr. Amit Sinha)**

**(Information Technology)**

**Date: 10th November 2021**

**Signature of Supervisor**

**(Name: Dr. Amit Sinha)**

**(Information Technology)**

**Date: 10th November 2021**

# Acknowledgement

I would like to express my special thanks of gratitude to my supervisor and HOD (Dr. Amit Sinha) who gave the opportunity to do this wonderful project on the topic (Fire Detection System Using Machine Learning Algorithms) which also helped me in doing alot of research and I came to know about so many new things I am really thankful to them.

Secondly, I would like to thank my parents and friends who helped me a lot in finalizing this project within the limited time frame.

**Signature of student**

**(Name: Prakash Singh)**

**(Roll No. 1803213112)**

**Signature of student**

**(Name: Harshit Singh)**

**(Roll No. 1803213066)**

**Signature of student**

**(Name: Pragati Gupta)**

**(Roll No. 1803213111)**

**Signature of student**

**(Name: Krishna Vishwakarma)**

**(Roll No. 1803213084)**

# Table of Contents

# List of Figures

# List of Tables



# List of Abbreviations

| **Abbreviations** | **Abbreviation Full Form** |
|---|---|
| NDVT | Neighbors Detection Value Table |
| WSN | Wireless Sensor Networks |
| FFNN | Feed Forward Neural Network |
| GPS | Global Positioning System |

# Abstract

There are lots of disasters happening in recent times in the world. One of the major disasters which have drawn the most attention is the large area fires, especially forest areas. Forest fires can occur naturally and also sometimes by humans. Forest Fires just does not harm the plants and trees of the forest, but also the animals living in the forest.

It also affects the nearby places where humans live since smoke, carbon dioxide and carbon mono-oxide concentration increases in the atmosphere which can lead to several health problems such as respiratory problems. We can detect the fire by deploying various wireless sensor nodes in the forest areas by which they sense the fire and can inform us, so that we can control the damage to be done by forest fire.

We can increase the efficiency of detection using machine learning techniques on the data collected by various wireless sensor nodes. In this report we will compare some of the machine learning techniques used by researchers with wireless sensor nodes in predicting forest fire

# Chapter 1
# Introduction

Sensor networks are exceptionally appropriated, lightweight hubs conveyed in huge numbers to screen the climate or framework in which they are available. Sensor networks are distributed in Ad Hoc manner. Sensor nodes are fitted with on-board processors.

## 1.1 What is Sensor?

Sensor senses the environment in which it is present. Sensor is a device which converts data collected from the environment into meaningful information.

Sensor senses the environment in which it is present. Sensor is a device which converts data collected from the environment into meaningful information.

Sensor Node consists of –

**a)** Sensor Subsystem,

**b)** Processing system and

**c)** Communication system.

Sensor senses the environment, collects the data and sends it to their respective base stations from where the base station sends it to the internet for further operations.

## 1.2 Components of Wireless Sensor Network:

In the paper[2], researchers presented the following parts of remote sensor network organization:

**(i). Sensor**: Sensor senses and monitors the environment. They captured the measured variable in a data acquisition network. For further processing sensor signal is converted into electrical signals.

**(ii). Wireless sensor nodes or radio nodes:** They received the sensor data from sensors.

**(iii). WLAN Access point:** Receives sensor data which are transferred by sensor nodes wirelessly.

**(iv) Evaluation software:** For any data analysis, WLAN access point is connected to an evaluation point.

## 1.3 Components of a Wireless Sensor Node

In the paper[2], researchers presented the following parts of remote sensor node:

**(i) Microcontroller:** This is a computer on a chip which is very small in size. It is capable of doing powerful tasks including controlling the function of other devices connecting to it.

Microcontroller consists of:
**(a)** Microprocessor
**(b)** Random Access Memory
**(c)** Associated peripherals

**(ii) Transceiver:** This is a transmitter receiver that is used for communication purposes to send and receive data and commands.

**(iii) External memory**: WSN nodes usually use flash memories. Flash memories are small in size and have reasonable storage capacity which can be used when needed.

**(iv) Power source:** Power is stored in the form of batteries. So, the wireless sensor nodes have limited power capacity. So, the detection method should be power efficient.

The basics of project include parameters such as fire causing factors, methodology which are as follows –

## 1.4 Sources of Fires

There are various factors involved when a forest catches fire. The hot and dry weather works as catalysts for fire. Since when there is fire in any region then the temperature of the region will become high due to fire. So, increase in temperature is one of the factors which can help in fire detection events.

But we cannot just depend on temperature to detect fire as the temperature of a region also may increase due to sunlight. So, we can also check the concentration of CO, $CO_2$ gases in that region to detect the fire.

But CO, CO2 gases can also be generated when someone uses the cigarette in that region so it may give false alarm, i.e., false fire event detection. So, to maintain accuracy in predicting the fire event using WSN, researchers in paper [12] presented an approach of setting thresholds on concentration of values. A value is set as a threshold on concentration of CO, CO2. It means that if concentration is greater than the threshold then it is fire, else not.

## 1.4 Methodology

- In the proposed System monitoring and tracking fires in different areas are done using Wireless Sensor Networks.
- Machine Learning Algorithms are used to detect if there is a fire in a particular area or not.

# Chapter 2
## Problem Statement

In earlier times fires were detected with the help of watching towers or using satellite images. Paper [8] discussed these approaches. Satellites collect images and send it to the monitoring authority which will decide by seeing images that it is a fire or not. But this approach was very slow as the fire may have spread in the large areas and caused so much damage before the rescue team came. In the watching tower method, there was a man always standing on the tower who would monitor the area and inform if there was fire. This method was also slow because before the man got to know about the fire it may have spread in the inner parts of forest, also it always requires a man who must be present there. Since, we know that some areas, especially forest areas are large so it is practically impossible to put a man in every part of forest from where they can monitor the forest area. So, both these approaches of watching towers and satellite images failed to detect fire as early as possible to reduce the damage done by fire Problems in fire detection:

There were mainly two problems in fire detection as discussed:

**(a). Judging criteria for the fire:** Edge is set, on the off chance that the worth is more noteworthy than edge, it is a fire, else not. So, this problem was removed by using machine learning techniques by many researchers.

**(b). Connection of nodes:** Traditional systems used cables to connect alarm with the detectors. Cable was mainly of copper. But copper wire may be costly or it can suffer from fault in the mid-way. So, this problem was removed using wireless sensor networks.

So, with the advancement in technology researchers find an efficient method to detect forest fire with the help of Wireless Sensor Network. Fire can be identified by conveying sensor hubs in timberland regions by which they illuminate about fire. [2] Conveying sensor hubs in the timberland regions means placing sensors in every part of the forest and mostly in the prone areas where risk of 9 catching fire is more. With the use of wireless sensor networks, now it is easy to detect the fire in large areas as soon as possible.

# Chapter 3
## Project Objective

In earlier times fires are detected with the help of watching towers or using satellite images. Satellites collect images and send it to the monitoring authority which will decide by seeing images that it is a fire or not. But this approach was very slow as the fire may have spread in the large areas and caused so much damage before the rescue team came. In the watching tower method, there was a man always standing on the tower who would monitor the area and inform if there was fire. This method was also slow because before the man got to know about the fire it may have spread in the inner parts of the area such as forests. Also, it always requires a man who must be present there. Since, we know that forest areas are large so it is practically impossible to put a man in every part of forest from where they can monitor the forest area. So, both these approaches of watching towers and satellite images failed to detect fire as early as possible to reduce the damage done by fire Problems in fire detection.

# Chapter 4
# Project Methodology

## Methodology

Dataset or informational index is taken from UCI Machine Learning repository; description of dataset is described as below:

- Dataset contains image and video data.
- Image data contains test and train data in image format each having 3 class i.e., default, smoke, fire.
- Test_default has 84 images, test_fire has 57 images, test_smoke has 30 images.
- Train_default has 161 images, train_fire has 274 images, train_smoke has 258 images.
- Video data contains test and train data in video format.
- Test_video contains 3 videos.
- Train_video contains 12 videos consisting of fire with smoke, only fire, only smoke, no fire videos.

## 4.1 Process Flow

- Loading Dataset.
- Customising (making relevant changes) in the dataset.
- Find the factors (or attributes) which are the most responsible for fire spread.

# 4.2 Using ML Algorithms

## (a). Using Logistic Regression

- train the machine using dataset
- determining accuracy, precision, recall of logistic regression.
- testing the dataset (providing custom input)

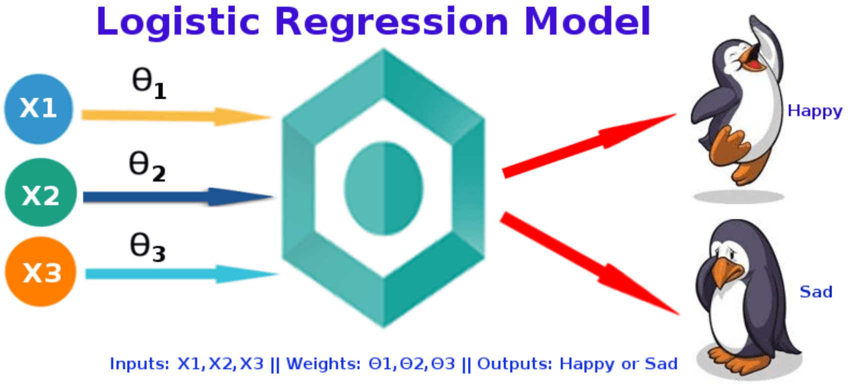


Fig 1: Logistic Regression Model

## (b). Using KNN Classification

- train the machine using dataset
- determining error rate and k value
- determining accuracy, precision, recall
- prediction using the test input.

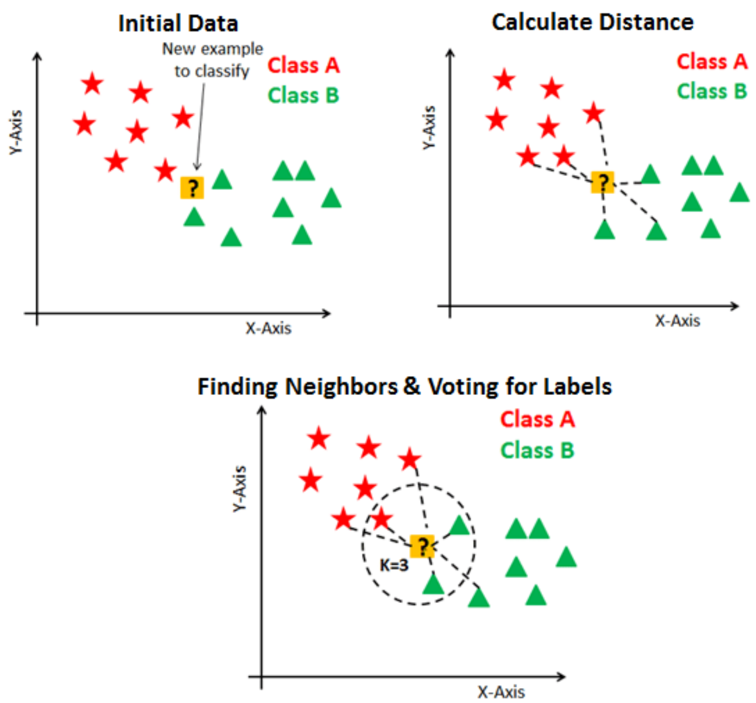


Fig 2: KNN Classification Model

### (c). Using SVM

- fit a SVM model to the dataset
- train the machine using dataset
- determining accuracy, precision, recall
- prediction using the test input.

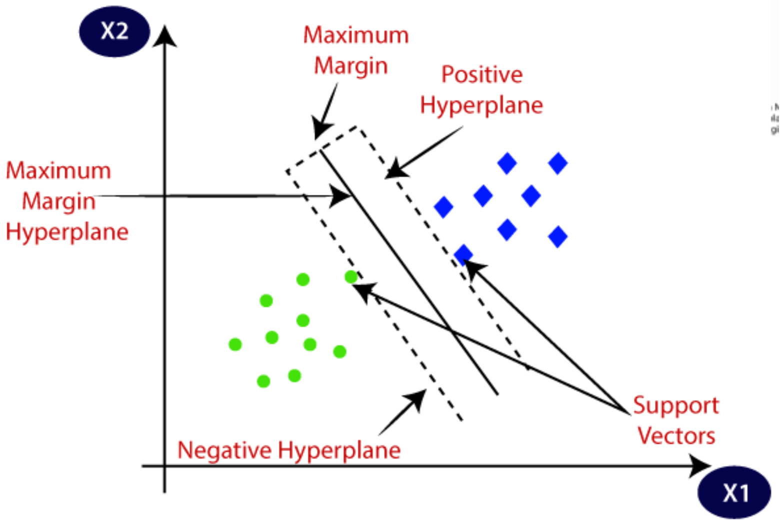


Fig 3: SVM Model

### (d). Using Decision Tree

- train the machine using dataset
- determining accuracy, precision, recall
- prediction using the test input.

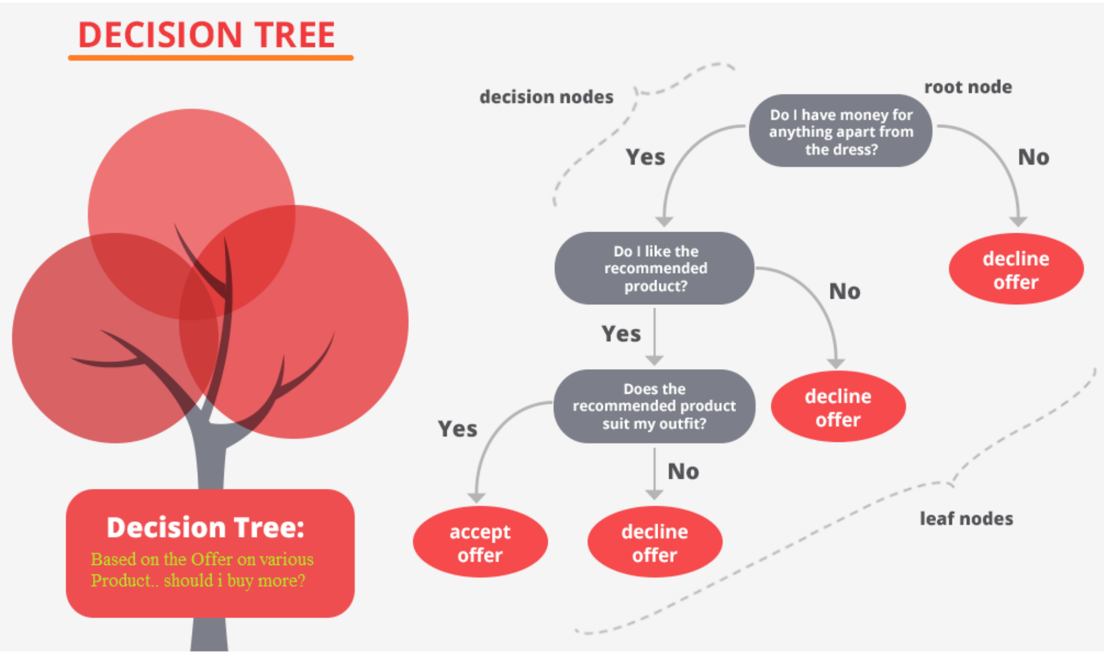


Fig 4: Decision Tree Model

### (e). Using Naïve Bayes

- fit a Naive Bayes model to the dataset
- train the machine using dataset
- determining accuracy, precision, recall
- prediction using the test input.

$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)}$$

using Bayesian probability terminology, the above equation can be written as

$$\text{Posterior} = \frac{\text{prior x likelihood}}{\text{evidence}}$$

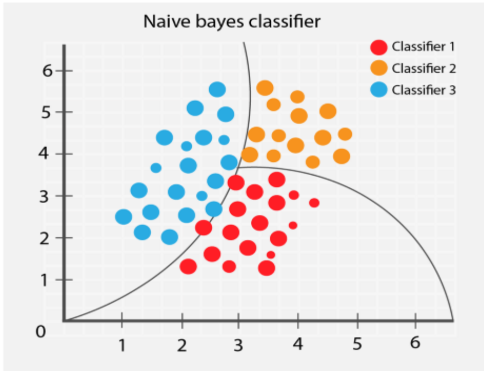


Fig 5: Naïve Bayes Model

### (f). Using Random Forest

- fit a random forest model to the dataset
- train the machine using dataset
- make predictions
- determining accuracy, precision, recall
- prediction using the test input.

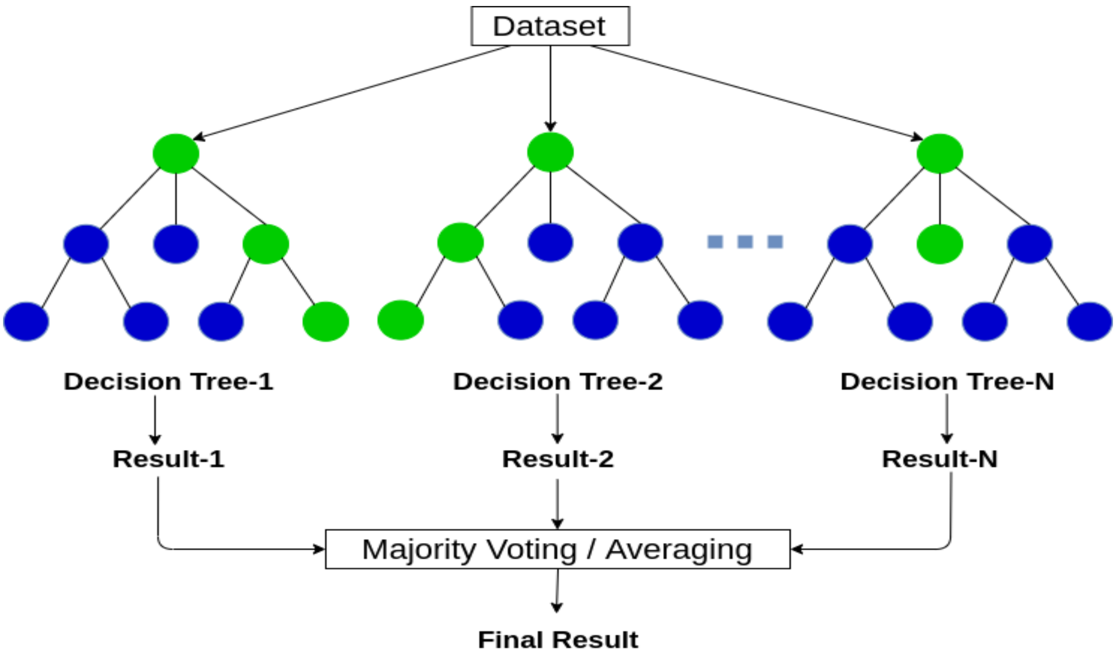


Fig 6: Random Forest Model

# Chapter 5
# Details of Project Work

Machine Learning techniques provide the enhancement of security and efficiency in wireless sensor networks. Machine learning techniques are helpful in analyzing patterns in the data

.

## 5.1 What is Machine Learning?

Machine learning is an application of artificial intelligence (AI).

According to Arthor Samuel, it is a field of study that gives computers the ability to learn without being explicitly programmed.

According to Tom Mitchell, a computer program is said to learn from experience E with respect to some task T and some performance measure P if its performance on task T as measured by P improves with experience E.

## 5.2 Types of Machine Learning

In the paper [8] researchers discussed the types of machine learning techniques. Machine Learning is of 3 types:

1. Supervised Learning (involves Decision Tree, Naive Bayes classifier)
2. Unsupervised Learning (involves clustering)
3. Reinforcement Learning

**1. Supervised Learning:** In this algorithm we have a target or outcome variable which is also known as a dependent variable. We have also a set of predictors also known as independent variables. We try to generate a function which maps independent variables to dependent variables.

**Examples of Supervised Learning Algorithms:** Regression, Classification, Decision Tree, Random Forest, K nearest neighbors, Logistic Regression etc.

**2. Unsupervised Learning:** In unsupervised learning algorithms, we only have independent variables, we do not have dependent variables. Using independent variables, we will try to make a cluster of outcomes.

**Examples of Unsupervised Learning Algorithm:** K-means algorithm, Apriori algorithm etc.

**3. Reinforcement Learning:** Utilizing support learning calculations, the machine is prepared to settle on explicit choices all alone. In this the machine is presented to a climate where it trains itself ceaselessly utilizing experimentation. The machine gains from its previous experience & attempts to further develop its forecast and precision all alone.

**Example of Reinforcement Learning Algorithm:** Markov Decision Process etc.

## 5.3 Machine Learning Techniques used in WSN

Now we will discuss some famous machine learning algorithms advantages and disadvantages when used in Wireless Sensor Network for event/fire detection.

**1. Decision Tree Based Event Detection**

**What is a Decision Tree?**

In the paper[2] researchers used a decision tree approach to detect the event. Decision tree is a kind of supervised learning algorithm in which dependent and independent variables both are given. Decision trees are utilized for arrangement issue. It works for both downright and ceaseless information and result factors. In this algorithm we divide the dataset on the basis of a splitter taken from input variables.

**Types of Decision Trees**

Based on it, dependent or outcome or target variables we can classify decision tree in 2 types:

**(a). Categorical Variable Decision Tree:** In this decision tree the objective variable is of unmitigated sort. Example: If we have to classify that a student is male or not, then its outcome can be yes or no i.e., categorical.

**(b). Continuous Variable Decision Tree:** In this decision tree the objective variable is of unmitigated sort.

**Decision tree construction for classification**: The first thing to be done in construction of a decision tree is training phase. The training phase comprises of bunch of information and a learning calculation to observe a decision tree which of least profundity so that the decision can be taken quicker. The tree should only contain nodes which have impact on output, rest other unimportant nodes are removed to reduce space complexity and reduce time complexity in reaching a decision. Greedy approach will be followed to reach the final decision tree which is of minimum depth.

Since there are various nodes in the remote sensor network, it makes the system robust for event detection. The decision tree algorithm will run on each and every node and then give their result to the voter where the voter runs a reputation-based voting technique or majority voting technique to reach the consensus. Results show that reputation-based voting techniques gave better results. **Reputation based voting:**

Decision tree algorithm runs on each node. After the decision has been made, a consensus has to be reached. We will use voting technique (reputation-based) to reach consensus, i.e., First we have to find the reputation of each sensor node and then choose the decisions made highest reputation by the node.

Reputation based voting:

(i). At first sensor nodes run their local decision tree classifier

(ii). After that every sensor hub sends its recognized occasion which is likewise called detection value (DV) to any remaining hubs/nodes.

(iii). All detection values are received from neighbors which will put away in a table called neighbors detection value table (NDVT).

(iv). Now each sensor node will judge about its neighboring sensor node taking itself as reference.

(v). Judgement criteria: Difference between DV of sensor node itself and value of another sensor node.

(vi). If difference < threshold value, positive vote given to another sensor node i.e., Vnew= $V_{old} + 1$

(vii). Else $V_{new} = V_{old} - 1$ i.e., negative vote given.

(viii). Finally, NDTV tables are sent to voters to reach a common consensus.

Advantage: Simplicity of decision tree approach provides high detection accuracy in detecting an event. It is also helpful in low computational overhead.

**Limitation:** Faulty nodes can lead to false votes.
To remove the limitation in the above approach a method was proposed based on trusted clustering-based event detection in the paper [13]. This method is based on first finding local reputation like above and after that it involves cluster formation of sensor nodes. All sensor nodes send their data after sensing the environment. Then a cluster head is nominated among all using an election algorithm. To check if a node is faulty, trusted value is calculated, after that cluster head maintains a record of trusted value and reputation for each sensor node which will be checked in case of event detection. Fault detection is also discussed in [6] with the help of data correlation.

**2. Naive Bayes Classifier**

In the paper[4] researchers discussed the Bayesian algorithm for event detection. Type of supervised algorithm and used for classification. This algorithm is used for classifying two class variables and multi class variables. Naive bayes classifier is best suited when used with categorical or binary input values.
Bayes' Theorem is given as:

**P(h|d) = (P(d|h) * P(h)) / P(d)**

Where

h : It is hypothesis h .

d : It is data d.

P(h) : It is called prior probability of hypothesis h being true irrespective of data

P(d) : It is called probability of data d irrespective of hypothesis.

P(h|d) : It is called posterior probability and defined as given data d ,the probability of h.

P(d|h) : It is the probability of given data d when hypothesis h is given true.

**Advantage of Naive bayes Classifier**

a: Provides better accuracy,

b: Lower communication overhead (due to centralization).

**Limitation:** Theoretically, we assume that all features are independent of each other but practically it is impossible that all features are independent of each other.

**3. Distributed Bayesian Algorithms**

In the paper [4] researchers used a distributed Bayesian algorithm which has the following advantages and disadvantages.

**Advantage:** Fault tolerant method. It utilizes the information that estimation blunders because of flawed gear are probably going to be uncorrelated, while ecological conditions are spatially corresponded. Distributed Bayesian calculation is utilized to recognize the event with fewer issues.

**Disadvantage:** Delay factor is high. Using of Distributed Event Detection technique in Wireless Sensor Networks for event detection by any machine learning

**Advantage:** Accuracy of detection is high.

**Disadvantage:** Data transmission will be slow. Two types of Distributed Bayesian algorithms are discussed in [4]. Threshold decision scheme and Randomized decision scheme. Research shows that threshold scheme is helpful in minimizing errors as its performance is high.

### 4. Feed Forward Neural Network (FFNN)

Researchers in the paper [10] gave an approach based on feed forward neural networks. Feed forward neural network is a sort of the neural networks, where each layer is taken care by back layer. Neural networks depend on biological neural networks i.e., the human brain. In neural networks, artificial neurons are interconnected to each other and they process information.

It comprises one input layer, at least one hidden layer and one output layer.

Challenge in FFNNs: The most challenging part of FFNN is finding the suitable weight for neurons to have better prediction and accuracy. Weights can be found using gradient descent method.

Advantage of Feedforward neural network in wireless sensor network:
a) Ease to be programmed in sensor node
b) Fast and accurate

### 5. Regression

In the paper[9] researchers discussed how regression can be used in detecting events and how it will be advantageous. Wireless sensor network provides low computational complexity algorithm using two sensors temperature sensor and humidity sensor. It was observed that when there is fire, temperature and humidity maintain an inverse relationship. It means that due to fire, there is an increase in temperature and decrease in humidity. So, in this paper a function was derived that maps the relationship between temperature and humidity in case of fire. The function was named as base function. After this the base function was used to compare the values obtained using temperature and humidity sensors to detect the fire.

It uses interpolation techniques and Dempster Shafer’s theory of evidence to determine whether it is a fire or not.

**Advantage:** The use of base function leads to a very good detection rate.

**Disadvantage:** It was not a robust approach. Since, the environment condition changes very rapidly in a short span of time, using it was not beneficial.

So, to make a robust approach the base function was made using regression analysis. Regression analysis is used for estimation of relationship between two or more quantities. Here in [9], it was used to make a base function or in other words we can say that regression analysis was used to estimate the relationship between temperature and humidity data collected by temperature and humidity sensor respectively. In this method time was made the dependent variable and humidity and temperature was made the independent variable.

**Advantage:** Regression analysis can represent data through adjustment of a mathematical function like straight line, exponential function and polynomial function etc. It can also be used to estimate unknown values.

**Disadvantage:** There is disadvantage in this method that if the sunlight falls directly on the sensor node, it will detect high temperature which may lead to false detection of fire in the environment in which the sensor lies.

### 6. Support vector machine:

In the paper [12] researchers presented an approach based on support vector machine. Support vector machines are a supervised learning model which is helpful in solving small datasets. It is mainly used for performing classification, regression as well as outlier detection. It can be used for nonlinear and linear problems and in this a hyperplane is made to separate datasets into classes. It is also useful in pattern recognition. It is also used for function fitting.

**Advantage:** Its accuracy is high and complexity is low and can detect faults.

### 7. Random Forest:

Random forest uses decision trees. Random forest models make various decision trees and combine their results to reach a more accurate result. It takes advantage of randomness by randomly selecting sub features.

**Advantage:** Improved performance than decision tree model.

## 5.4 Project Workflow

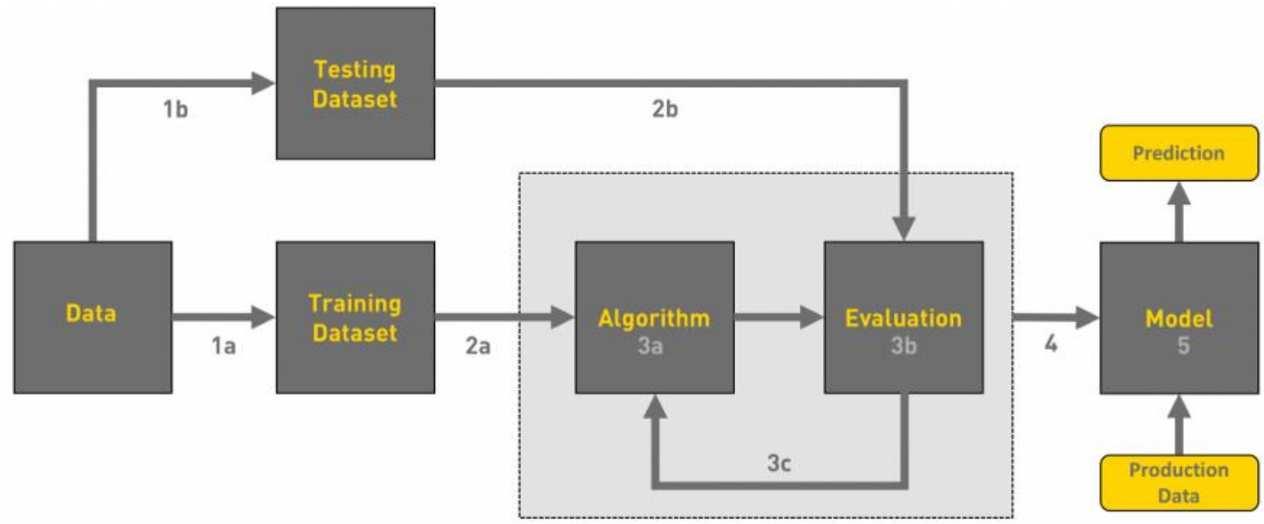


Fig 7: Project Workflow Diagram

- Gathering data
- Data pre-handling
- Investigating the model that will be best for the kind of information.
- Evaluating models by training and testing.

## 5.5 Implementation Diagram of Algorithms

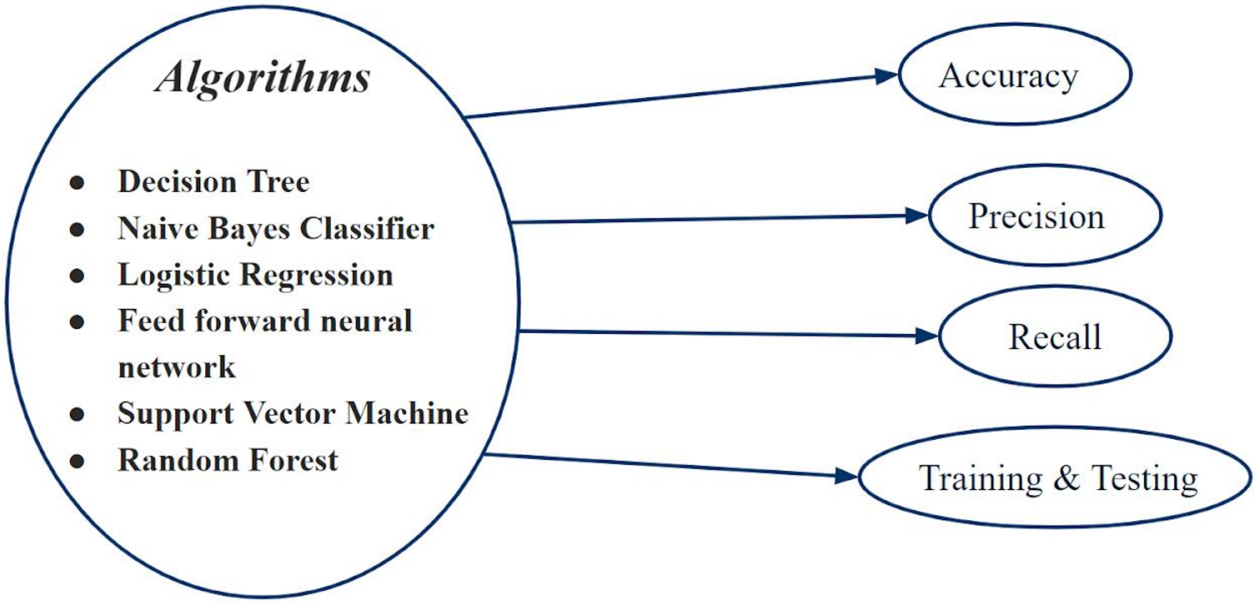


Fig 8: Implementation Diagram

# Chapter 6
# Results and Discussions

For this purpose, **"Forest-fires.csv"** dataset from UCI machine learning repository was taken and machine learning algorithms were applied to find accuracy of detection. The dataset **"Forest-fires.csv"** contains 517 instances and 13 attributes.

Table 1: Accuracy Analysis of Algorithms

| Machine Learning Algorithm | Accuracy |
|---|---|
| Decision Tree | 52.56% |
| Naïve Bayes Classifier | 48.07% |
| Logistic Regression | 55.80% |
| Support Vector Machine | 61.50% |
| K nearest neighbour | 62.82% |

## 6.1 Project Observations

### (a). Logistic Regression

```
Precision: 0.5462184873949579
Recall: 0.7142857142857143
Accuracy: 0.5580110497237569
[[36 54]
 [26 65]]
              precision    recall  f1-score   support

         0.0       0.58      0.40      0.47        90
         1.0       0.55      0.71      0.62        91

    accuracy                           0.56       181
   macro avg       0.56      0.56      0.55       181
weighted avg       0.56      0.56      0.55       181
```

**Prediction using Logistic Regression**

```
class_label={1:'There is Fire',0:'There is no fire'}
x_new=[[1, 4, 9 ,1 ,91.5, 130.1, 807.1, 7.5, 21.3, 35, 2.2, 0]]


y_predict=logistic_model.predict(x_new)
print(class_label[y_predict[0]])

There is no fire
```

## (b). KNN Classification

```
from sklearn import metrics
print("Accuracy:",metrics.accuracy_score(y_test, pred))
print("Precision:",metrics.precision_score(y_test, pred))
print("Recall:",metrics.recall_score(y_test, pred))
```

```
Accuracy: 0.5193370165745856
Precision: 0.5181818181818182
Recall: 0.6263736263736264
```

```
WITH K=17


[[37 53]
 [34 57]]


              precision    recall  f1-score   support

         0.0       0.52      0.41      0.46        90
         1.0       0.52      0.63      0.57        91

    accuracy                           0.52       181
   macro avg       0.52      0.52      0.51       181
weighted avg       0.52      0.52      0.51       181
```

### Prediction using KNN Classification

```
classes={0:'safe',1:'On Fire'}
x_new=[[1, 4, 9 ,1 ,91.5, 130.1, 807.1, 7.5, 21.3, 35, 2.2, 0]]
y_predict=knn.predict(x_new)
print(classes[y_predict[0]])
```

```
On Fire
```

## (c). Support Vector Machine (SVM)

```
              precision    recall  f1-score   support

         0.0       0.65      0.29      0.40        69
         1.0       0.61      0.87      0.72        87

    accuracy                           0.62       156
   macro avg       0.63      0.58      0.56       156
weighted avg       0.62      0.62      0.58       156

[[20 49]
 [11 76]]
Accuracy: 0.6153846153846154
Precision: 0.608
Recall: 0.8735632183908046
```

### Prediction using SVM

```
classes={0:'safe',1:'On Fire'}
x_new=[[1, 4, 9 ,1 ,91.5, 130.1, 807.1, 7.5, 21.3, 35, 2.2, 0]]
y_predict=svc.predict(x_new)
print(classes[y_predict[0]])
```

```
On Fire
```

## (d). Decision Tree

```
              precision    recall  f1-score   support

         0.0       0.44      0.42      0.43        69
         1.0       0.56      0.57      0.56        87

    accuracy                           0.51       156
   macro avg       0.50      0.50      0.50       156
weighted avg       0.50      0.51      0.51       156

[[29 40]
 [37 50]]
Accuracy: 0.5064102564102564
Precision: 0.5555555555555556
Recall: 0.5747126436781609
```

### Prediction using Decision Tree

```
classes={0:'safe',1:'On Fire'}
x_new=[[1, 4, 9 ,1 ,91.5, 130.1, 807.1, 7.5, 21.3, 35, 2.2, 0]]
y_predict=d_tree.predict(x_new)
print(classes[y_predict[0]])
```

```
On Fire
```

## (e). Naïve Bayes

```
GaussianNB()
              precision    recall  f1-score   support

         0.0       0.45      0.80      0.58        69
         1.0       0.59      0.23      0.33        87

    accuracy                           0.48       156
   macro avg       0.52      0.51      0.45       156
weighted avg       0.53      0.48      0.44       156

[[55 14]
 [67 20]]
Accuracy: 0.4807692307692308
Precision: 0.5882352941176471
Recall: 0.22988505747126436
```

### Prediction using Naïve Bayes

```
classes={0:'safe',1:'On Fire'}
x_new=[[1, 4, 9 ,1 ,91.5, 130.1, 807.1, 7.5, 21.3, 35, 2.2, 0]]
y_predict=G_NB.predict(x_new)
print(classes[y_predict[0]])
```

```
On Fire
```

### (f). Random Forest

```
              precision    recall  f1-score   support

         0.0       0.50      0.61      0.55        69
         1.0       0.62      0.52      0.57        87

    accuracy                           0.56       156
   macro avg       0.56      0.56      0.56       156
weighted avg       0.57      0.56      0.56       156

[[42 27]
 [42 45]]
Accuracy: 0.5576923076923077
Precision: 0.625
Recall: 0.5172413793103449
```

**Prediction using Random Forest**

```
classes={0:'safe',1:'On Fire'}
x_new=[[1, 4, 9 ,1 ,91.5, 130.1, 807.1, 7.5, 21.3, 35, 2.2, 0]]
y_predict=random_forest.predict(x_new)
print(classes[y_predict[0]])

On Fire
```

Table 2: Process Advantages / Disadvantages

| **References** | **Disadvantages** | **Advantages** |
|---|---|---|
| Utilization of AI Techniques for private fire identification in remote sensor organizations. | Risk of delay and bottleneck. | Reduces Fake alarm. |
| Dispersed Bayesian calculations for shortcoming open-minded occasion locale identification in remote sensor organizations. | Risk of delay due to slow data transmission | Fault tolerant method and high accuracy. |
| Conveyed occasion identification in remote sensor networks for calamity the board. | False alarm due to faulty nodes. | Low computation overhead due to the decision tree. |
| Wireless sensor network for forest fire detection. | Not a robust approach. | Good detection rate due to regression. |

# Chapter 7
# Conclusion and Future Scope

Wireless sensor networks are helpful in detecting events. In the case of forest fire detection wireless network sensor nodes remove the difficulty faced in traditional methods like man standing on a tower and monitoring the environment. Now with the use of WSN we can put sensor nodes in each and every part of forest and mostly in the region where the risk is high. All the data collected by sensor nodes have to be aggregated to reach the result so it is done by using tree based and cluster-based methods.

The machine learning techniques add enhancement to the security of wireless sensor networks. With the use of machine learning techniques, the problem of faulty nodes is minimized. With the use of regression algorithm network lifetime is enhanced and with the use of decision tree algorithm network lifetime is enhanced as well as accuracy. SVM and neural network give better results.

## Future Scope

We will be finding a method based on machine learning which will be

- Accurate in prediction
- Fault Tolerant
- Robust

and then finding its space and time complexity and will try to optimise it.

# References

[1] M. Bahrepour, N. Meratnia and P. J. M. Havinga, "Use o–f AI Techniques for Residential Fire Detection in Wireless Sensor Networks", AIAI 2009, 2009.

[2] M. Bahrepour, N. Meratnia, M. Poel, Z. Taghikhaki and P. J. Havinga, "Distributed event detection in wireless sensor networks for disaster management", Proc. 2nd Int. Conf. Intell. Netw. Collab. Syst., pp. 507-512, 2010.

[3] G. Wittenburg, N. Dzicngel, C. Wartenburger and J. Schiller, "A system for distributed event detection in wireless sensor networks", Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks, pp. 94-104, 2010.

[4] B. Krishnamachari and S.S. Iyengar, "Distributed Bayesian Algorithms for Fault-Tolerant Event Region Detection in Wireless Sensor Networks", IEEE Trans. Computers, vol. 53, no. 3, Mar. 2004.

[5] M. Bahrepour, N. Meratnia and P. J. M. Havinga, "Automatic Fire Detection: A Survey from Wireless Sensor Network Perspective", 2008.

[6] Karthik, N., & Ananthanarayana, V. S. (2017). Data trust model for event detection in wireless sensor networks using data correlation techniques. In IEEE 4th Conference on Signal Processing, Communications and Networking (In press).

[7] Wu, Q., Cao, J., Zhou, C., Huang, J., Li, Z., Cheng, S. M., ... & Pan, G. (2018). Intelligent Smoke Alarm System with Wireless Sensor Network Using ZigBee. Wireless Communications and Mobile Computing, 2018.

[8] Zaki Ahmad and Abdus Samad, "A Study of Machine Learning in Wireless Sensor Network", International Journal of Computer Networks and Applications (IJCNA), vol. 4, no. 4, August 2017.

[9] Noel Varelaa, Díaz-Martinez, “Wireless sensor network for forest fire detection” The 15th International Conference on Future Networks and Communications (FNC) August 9-12, 2020, Leuven, Belgium

[10] M. O. Oladimeji, M. G. Smiee and S. D. Mieee, A New Approach for Event Detection using k -means Clustering and Neural Networks, pp. 1-5, 2015.

[11] Sumit Jha, Shubham Narayan, “Fire Detection System Using Wireless Sensor Network” International Journal of Research in Engineering, Science and Management Volume-3, Issue-3, March-2020

[12] Divyansh Puri, Bharat Bhushan, “Enhancement of security and energy efficiency in WSNs: Machine Learning to the rescue” 2019 International Conference on Computing, Communication, and Intelligent Systems(ICCCIS)

[13] Apurva Saoji, P.D.Lambhate, “Trusted Clustering Based Event Detection for a Disaster Management in Wireless Sensor Network”, International Journal of Computer Science and Information Technologies, Vol. 5 (4), 2014

[14] https://www.analyticsvidhya.com/

# Appendix-I (Coding)

***#importing required python libraries***

```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
from sklearn import metrics
from sklearn.metrics import classification_report,confusion_matrix

import warnings
warnings.filterwarnings(action="ignore")
%matplotlib inline
pd.set_option("display.max_rows", 1000)
pd.set_option("display.max_columns", 1000)
```

***#reading the dataset***

```
fires = pd.read_csv("forestfires.csv")
```

***#show the first 15 instances of dataset***

```
fires.head(15)
```

***#show the last 10 instances of dataset***

```
fires.tail(10)
```

***#changing days into numeric quantity because machine learning model deals with numbers***

```
fires.day.replace(('mon','tue','wed','thu','fri','sat','sun'),(1,2,3,4,5,6,7), inplace=True)
```

***#changing month into numeric quantity***

```
fires.month.replace(('jan','feb','mar','apr','may','jun','jul','aug','sep','oct','nov','dec'),(1,2,3,4,5
,6,7,8,9,10,11,12), inplace=True)
```

***#showing first 10 instances of dataset after converting days and months into numbers***

```
fires.head(15)
```

***#generate descriptive statistics of each attribute***

```
fires.describe().T
```

***#given area of land burnt, but we have to predict if there is fire or not so changing values of area to 0 and 1 only***
***#here 0 represet there is not fire and 1 represent fire, changing all values of area which are greater than 0 to 1***

```
fires['area'].values[fires['area'].values > 0] = 1
```

***#renaming the area attribute to output for clear understanding***

```
fires = fires.rename(columns={'area': 'output'})
fires.head(5)
```

***#compute pairwise correlation of columns***

```
fires.corr()
```

***#sorting to see which attribute is correlated more to attribut "output"***

```
fires.corr()['output'].sort_values()
from sklearn.preprocessing import StandardScaler
```

```
#standardization of data
#removing the mean and scaling it to unit variance
#score=(x-mean)/std

scaler = StandardScaler()
#fitting forest fire dataset to scaler by removing the attribute output
scaler.fit(fires.drop('output',axis=1))

scaled_features = scaler.transform(fires.drop('output',axis=1))
df_feat = pd.DataFrame(scaled_features,columns=fires.columns[:-1])
df_feat.head()

#importing logistic regression

from sklearn.linear_model import LogisticRegression
logistic_model = LogisticRegression()
logistic_model.fit(X_train,y_train)

predictions = logistic_model.predict(X_test)

#finding precision,recall,accuracy

print("Precision:",metrics.precision_score(y_test, predictions))
print("Recall:",metrics.recall_score(y_test, predictions))
print("Accuracy:",metrics.accuracy_score(y_test, predictions))

print(confusion_matrix(y_test,predictions))
print(classification_report(y_test,predictions))

#prediction using logistic regression

class_label={1:'There is Fire',0:'There is no fire'}
x_new=[[1, 4, 9 ,1 ,91.5, 130.1, 807.1, 7.5, 21.3, 35, 2.2, 0]]
```

```
y_predict=logistic_model.predict(x_new)
print(class_label[y_predict[0]])
```

***#importing k nearest neighbour***

```
from sklearn.neighbors import KNeighborsClassifier
k_nearest_neighbor_model = KNeighborsClassifier(n_neighbors=1)
k_nearest_neighbor_model.fit(X_train,y_train)
pred = k_nearest_neighbor_model.predict(X_test)

error_rate = []
for i in range(1,100):
    k_nearest_neighbor_model = KNeighborsClassifier(n_neighbors=i)
    k_nearest_neighbor_model.fit(X_train,y_train)
    pred_i = k_nearest_neighbor_model.predict(X_test)
    error_rate.append(np.mean(pred_i != y_test))


plt.figure(figsize=(10,6))
plt.plot(range(1,100),error_rate,color='blue', linestyle='dashed', marker='o',
         markerfacecolor='red', markersize=10)
plt.title('Error Rate vs. K Value')
plt.xlabel('K')
plt.ylabel('Error Rate')

knn = KNeighborsClassifier(n_neighbors=7)
knn.fit(X_train,y_train)
```

```
pred = knn.predict(X_test)
print('WITH K=7')
print('\n')
print(confusion_matrix(y_test,pred))
print('\n')
print(classification_report(y_test,pred))

knn = KNeighborsClassifier(n_neighbors=17)
knn.fit(X_train,y_train)
pred = knn.predict(X_test)
print('WITH K=17')
print('\n')
print(confusion_matrix(y_test,pred))
print('\n')
print(classification_report(y_test,pred))

knn.score(X_test, y_test)

from sklearn import metrics
print("Accuracy:",metrics.accuracy_score(y_test, pred))
print("Precision:",metrics.precision_score(y_test, pred))
print("Recall:",metrics.recall_score(y_test, pred))
```

***#prediction using knn***

```
classes={0:'safe',1:'On Fire'}
x_new=[[1, 4, 9 ,1 ,91.5, 130.1, 807.1, 7.5, 21.3, 35, 2.2, 0]]
```

```
y_predict=knn.predict(x_new)

print(classes[y_predict[0]])
```

***# Support Vector Machine***

```
from sklearn.svm import SVC
```

***# fit a SVM model to the data***

```
X = fires.drop('output', axis=1)

y = fires['output']

X_train, X_test, y_train, y_test = train_test_split(X,y, test_size=0.3,random_state=101)

svc = SVC()

svc.fit(X_train, y_train)
```

***# make predictions***

```
prediction = svc.predict(X_test)
```

***# summarize the fit of the model***

```
print(metrics.classification_report(y_test, prediction))

print(metrics.confusion_matrix(y_test, prediction))

print("Accuracy:",metrics.accuracy_score(y_test, prediction))

print("Precision:",metrics.precision_score(y_test, prediction))
```

```
print("Recall:",metrics.recall_score(y_test, prediction))
```

***#prediction using svm***

```
classes={0:'safe',1:'On Fire'}
x_new=[[1, 4, 9 ,1 ,91.5, 130.1, 807.1, 7.5, 21.3, 35, 2.2, 0]]
y_predict=svc.predict(x_new)
print(classes[y_predict[0]])
```

***#import decision trees***

```
from sklearn import metrics
from sklearn.tree import DecisionTreeClassifier

X = fires.drop('output', axis=1)
y = fires['output']

X_train, X_test, y_train, y_test = train_test_split(X,y, test_size=0.3,random_state=101)

d_tree = DecisionTreeClassifier()
d_tree.fit(X_train, y_train)
```

***# make predictions***

```
predicted = d_tree.predict(X_test)
```

***# summarize the fit of the model***

```
print(metrics.classification_report(y_test, predicted))

print(metrics.confusion_matrix(y_test, predicted))


print("Accuracy:",metrics.accuracy_score(y_test, predicted))

print("Precision:",metrics.precision_score(y_test, predicted))

print("Recall:",metrics.recall_score(y_test, predicted))

#prediction using decision tree


classes={0:'safe',1:'On Fire'}

x_new=[[1, 4, 9 ,1 ,91.5, 130.1, 807.1, 7.5, 21.3, 35, 2.2, 0]]

y_predict=d_tree.predict(x_new)

print(classes[y_predict[0]])

#import naive bayes


from sklearn import metrics

from sklearn.naive_bayes import GaussianNB


X = fires.drop('output', axis=1)

y = fires['output']


X_train, X_test, y_train, y_test = train_test_split(X,y, test_size=0.3,random_state=101)

# fit a Naive Bayes model to the data


G_NB = GaussianNB()
```

```
G_NB.fit(X_train,y_train)

print(G_NB)
```

***# make predictions***

```
predict = G_NB.predict(X_test)
```

***# summarize the fit of the model***

```
print(metrics.classification_report(y_test, predict))

print(metrics.confusion_matrix(y_test, predict))

print("Accuracy:",metrics.accuracy_score(y_test, predict))

print("Precision:",metrics.precision_score(y_test, predict))

print("Recall:",metrics.recall_score(y_test, predict))
```

***#prediction using naive bayes***

```
classes={0:'safe',1:'On Fire'}

x_new=[[1, 4, 9 ,1 ,91.5, 130.1, 807.1, 7.5, 21.3, 35, 2.2, 0]]

y_predict=G_NB.predict(x_new)

print(classes[y_predict[0]])
```

***#import random forest***

```
from sklearn.ensemble import RandomForestClassifier

X = fires.drop('output', axis=1)

y = fires['output']
```

```
X_train, X_test, y_train, y_test = train_test_split(X,y, test_size=0.3,random_state=101)

# fit a Naive Bayes model to the data

random_forest = RandomForestClassifier()
random_forest.fit(X_train,y_train)

# print(random_forest)
# make predictions

predict = random_forest.predict(X_test)

# summarize the fit of the model

print(metrics.classification_report(y_test, predict))
print(metrics.confusion_matrix(y_test, predict))

print("Accuracy:",metrics.accuracy_score(y_test, predict))
print("Precision:",metrics.precision_score(y_test, predict))
print("Recall:",metrics.recall_score(y_test, predict))

#prediction using random forest

classes={0:'safe',1:'On Fire'}
x_new=[[1, 4, 9 ,1 ,91.5, 130.1, 807.1, 7.5, 21.3, 35, 2.2, 0]]
y_predict=random_forest.predict(x_new)
print(classes[y_predict[0]])
```

P16

ORIGINALITY REPORT

| 6% | 4% | 5% | % |
| --- | --- | --- | --- |
| SIMILARITY INDEX | INTERNET SOURCES | PUBLICATIONS | STUDENT PAPERS |

PRIMARY SOURCES

| | | |
| --- | --- | --- |
| 1 | Majid Bahrepour, Nirvana Meratnia, Mannes Poel, Zahra Taghikhaki, Paul J.M. Havinga. "Distributed Event Detection in Wireless Sensor Networks for Disaster Management", 2010 International Conference on Intelligent Networking and Collaborative Systems, 2010<br>Publication | 1% |
| 2 | indico.ictp.it<br>Internet Source | 1% |
| 3 | tsapps.nist.gov<br>Internet Source | 1% |
| 4 | royallib.com<br>Internet Source | <1% |
| 5 | dspace.bracu.ac.bd:8080<br>Internet Source | <1% |
| 6 | home.simula.no<br>Internet Source | <1% |
| 7 | vsip.info<br>Internet Source | <1% |

| | | |
|---|---|---|
| 8 | repositorio.ufmg.br<br>Internet Source | <1 % |
| 9 | aisel.aisnet.org<br>Internet Source | <1 % |
| 10 | N. Cercone. "Naive Bayes Modeling with Proper Smoothing for Information Extraction", 2006 IEEE International Conference on Fuzzy Systems, 2006<br>Publication | <1 % |
| 11 | mafiadoc.com<br>Internet Source | <1 % |
| 12 | aibiotechnology.com<br>Internet Source | <1 % |
| 13 | "Inventive Communication and Computational Technologies", Springer Science and Business Media LLC, 2022<br>Publication | <1 % |
| 14 | Innes Murdo Black, Mark Richmond, Athanasios Kolios. "Condition monitoring systems: a systematic literature review on machine-learning methods improving offshore-wind turbine operational management", International Journal of Sustainable Energy, 2021<br>Publication | <1 % |