SMOKE DETECTION

A Course Project report submitted

in partial fulfillment of requirement for the award of degree

BACHELOR OF TECHNOLOGY

in

ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

by

GUGULOTHU VAMSHI	**(2103A51403)**
LAGISHETTI VIGNESH	**(2103A51271)**
PODISHETTI SATHWIK	**(2103A51490)**

Under the guidance of

Mr. S. NARESH KUMAR

Assistant Professor, Department of CSE.

SR UNIVERSITY

Department of Computer Science and Artificial Intelligence

Department of Computer Science and Artificial Intelligence

CERTIFICATE

This is to certify that project entitled **"SMOKE DETECTION**" is the bonafide work carried out by **G VAMSHI, P SATHWIK, L VIGNESH** as a Course Project for the partial fulfillment to award the degree **BACHELOR OF TECHNOLOGY** in **ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING** during the academic year 2022-2023 under our guidance and Supervision.

Mr. S Naresh Kumar
Asst. Professor,
S R University,
Ananthasagar, Warangal.

Dr. M.Sheshikala
Assoc. Prof. & HOD (CSE),
S R University,
Ananthasagar, Warangal.

ACKNOWLEDGEMENT

We express our thanks to Course co-coordinator **Mr. S.Naresh Kumar, Asst. Prof.** for guiding us from the beginning through the end of the Course Project. We express our gratitude to Head of the department CS&AI, **Dr. M.Sheshikala, Associate Professor** for encouragement, support and insightful suggestions. We truly value their consistent feedback on our progress, which was always constructive and encouraging and ultimately drove us to the right direction.

We wish to take this opportunity to express our sincere gratitude and deep sense of respect to our beloved Dean, School of Computer Science and Artificial Intelligence, **Dr C. V. Guru Rao**, for his continuous support and guidance to complete this project in the institute.

Finally, we express our thanks to all the teaching and non-teaching staff of the department for their suggestions and timely support.

ABSTRACT

This project aims to develop a machine learning-based smoke detection system using sensor data. The project will collect data from a sensor array that includes smoke detectors as well as other environmental variables such as temperature, humidity, and particulate matter concentrations. The collected data will be used to train and test machine learning models to detect the presence of smoke in indoor environments. The project will evaluate the performance of different machine learning algorithms, feature engineering techniques, and sensor configurations to optimize the accuracy and reliability of the smoke detection system. The results of this project can be used to inform the design of more effective and efficient smoke detection systems for a range of applications, including building automation, safety monitoring, and air quality management.

The project will evaluate the performance of different machine learning algorithms, including supervised and unsupervised techniques, to identify the optimal algorithm for detecting smoke. Feature engineering techniques will also be employed to extract relevant features from the sensor data to improve model performance.

The results of this project can be used to inform the design of more effective and efficient smoke detection systems. Such systems can be deployed in a range of applications, including building automation, safety monitoring, and air quality management. By improving smoke detection accuracy, the proposed system can potentially save lives and reduce property damage caused by fires.

Table of Contents

1.INTRODUCTION

1.1 Overview

The smoke detection project is aimed at developing a machine learning model to detect smoke in indoor environments using sensor data. The project involves collecting sensor readings related to various environmental factors such as temperature, humidity, pressure, and air quality parameters like TVOC and eCO2 levels. These readings can provide insights into the changes in the environment over time and can be used to train supervised learning models to detect smoke.

To train the model, the collected data can be preprocessed, cleaned, and transformed, and data augmentation techniques can be applied to artificially increase the size of the dataset. Once the data is ready, various machine learning models can be trained and evaluated using standard metrics such as accuracy, precision, recall, and F1 score.

Once a suitable model is identified, it can be deployed in real-time to detect smoke in indoor environments. The model can be integrated with a smoke detection system to trigger alarms or alerts in case of a fire. The project can have significant implications in improving the safety of indoor environments and can be used in various settings such as homes, offices, hospitals, and public spaces.

Overall, the smoke detection project is an important area of research and development, and the application of machine learning and sensor technology can provide an effective solution to improve fire safety in indoor environments.

1.2 Problem Statement

The main objective of this project is to detect the presence of smoke in an environment using Artificial Intelligence and Machine Learning. The project aims to develop an intelligent smoke detection system that can be deployed in different environments, including indoor and outdoor spaces. The proposed system will use various sensors, including temperature and humidity sensors, to detect the presence of smoke in the environment. The system will then use machine learning algorithms to analyze the data and determine if smoke is present. The system will also be designed to send alerts to the user or authorities if smoke is detected, allowing them to take appropriate action.

The problem we are trying to solve is the need for an efficient and reliable smoke detection system that can detect smoke in different environments. Current smoke detectors on the market are not always reliable, and they may not work in certain environments or may produce false alarms. The proposed system aims to overcome these limitations by using advanced machine learning algorithms that can analyze sensor data and detect the presence of smoke accurately. This system will be more efficient and reliable than traditional smoke detectors, making it ideal for use in a variety of environments.

1.3 Existing System

1.Smoke detection using image processing:

This system uses image processing techniques to detect smoke in video streams. Machine learning models such as convolutional neural networks (CNNs) can be trained to classify images as containing smoke or not.

2.Smoke detection using audio analysis:

This system uses audio processing techniques to analyze the sounds of smoke alarms and other sounds in the environment. Machine learning models can be trained to recognize the specific sound patterns of smoke alarms and differentiate them from other sounds.

3.Smoke detection using wireless sensor networks:

This system uses a network of wireless sensors to detect smoke in indoor environments. Machine learning models can be trained to analyze the sensor data and detect patterns that indicate the presence of smoke.

Overall, these systems use machine learning to analyze various types of data and detect patterns that indicate the presence of smoke. They can be effective solutions for improving fire safety in indoor environments.

1.4 Proposed System

Smoke detection using environmental sensors:

This system collects data from environmental sensors such as temperature, humidity, and air quality sensors. Machine learning models can be trained to detect patterns in the sensor data that are indicative of smoke.

1.5 Objectives

The objectives of smoke detection in machine learning can be defined as follows:

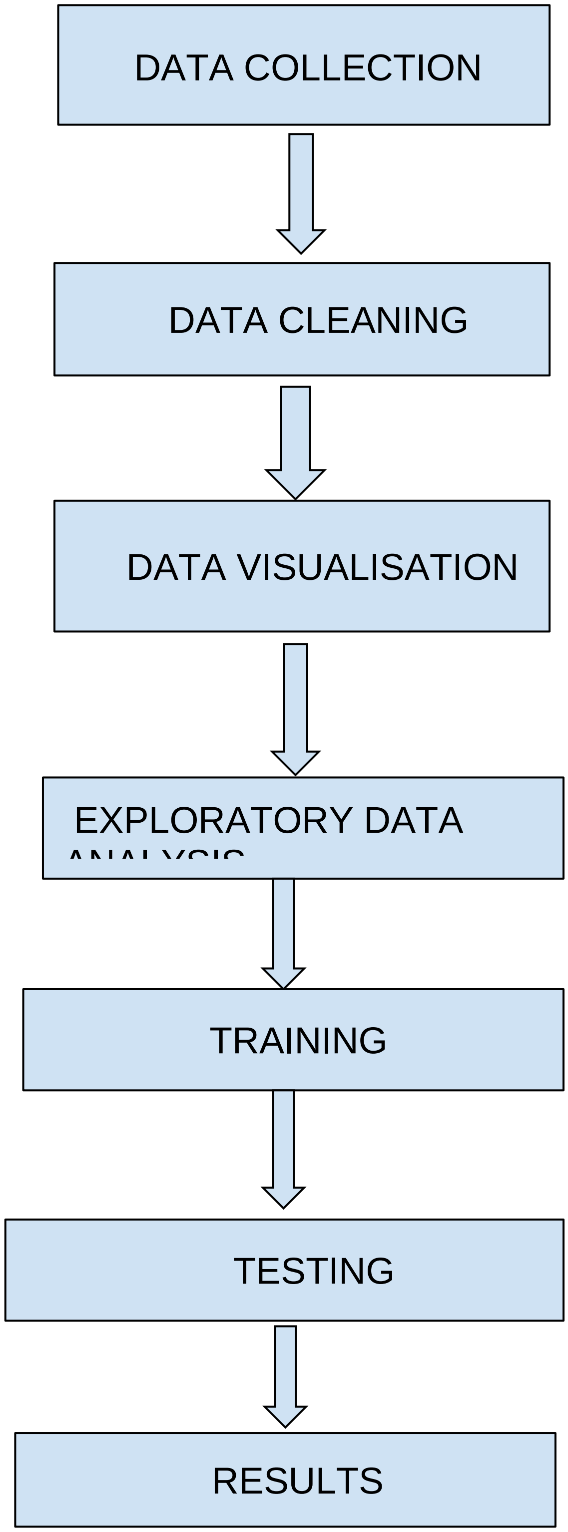
Temperature and humidity monitoring: The temperature_c and humidity columns suggest that the project may involve monitoring temperature and humidity levels in a particular environment. The objective may be to detect changes in temperature and humidity that could indicate a potential fire or other hazard.

Air quality monitoring: The tvoc_ppb, eco2_ppm, raw_h2, and raw_ethanol columns suggest that the project may also involve monitoring the air quality in the environment. The objective may be to detect changes in air quality that could indicate the presence of smoke, hazardous chemicals, or other pollutants.

Fire detection: The fire_alarm column suggests that the project may involve detecting the presence of a fire. The objective may be to detect the presence of smoke or other indicators of a fire and raise an alarm to alert people in the area.

Real-time monitoring: The utc column suggests that the project may involve real-time monitoring of the environment. The objective may be to detect changes in temperature, humidity, air quality, or other factors as soon as possible to minimize the risk of damage or injury.

1.6 Architecture



2.LITERARY SURVEY

Smoke detection is a critical aspect of fire safety, and it has been the subject of extensive research over the years. Here is a brief literary survey on smoke detection:

"Smoke Detection Using Computer Vision: A Review" by R. L. Nithya and P. Venkatesan (2021) - This review paper discusses the use of computer vision techniques for smoke detection in various settings, including industrial and residential environments.

"A Review of Smoke Detection Technologies" by M. K. Das and S. S. Sahu (2018) - This paper provides a comprehensive review of smoke detection technologies, including ionization, photoelectric, and optical methods. It also discusses the advantages and limitations of each technology.

"A Comparative Study of Smoke Detection Techniques" by S. K. Jain and R. D. Gupta (2017) - This paper compares the performance of various smoke detection techniques, including ionization, photoelectric, and optical methods, using experimental data.

"Smoke Detection in Fire Safety: A Review" by Y. Wei and M. Jia (2016) - This review paper provides an overview of smoke detection techniques and their applications in fire safety. It also discusses the challenges and opportunities for future research in this area.

"A Survey on Smoke Detection Techniques and Sensors" by S. H. Ahmed and M. S. Rahman (2015) - This survey paper reviews various smoke detection techniques and sensors, including ionization, photoelectric, and optical methods. It also discusses the recent advances and challenges in this field.

"A Novel Smoke Detection Algorithm Based on Image Contrast" by H. Ma et al. (2016) - This paper proposes a novel smoke detection algorithm based on image contrast, which is shown to be effective in detecting smoke in complex environments.

"Smoke Detection in Video: A Survey" by P. Rezazadeh et al. (2018) - This survey paper provides an overview of smoke detection in video, including techniques for detecting smoke in outdoor and indoor environments, and the challenges and opportunities for future research in this area.

"Deep Learning-Based Smoke Detection Using Convolutional Neural Networks" by S. Khan et al. (2020) - This paper proposes a deep learning-based smoke detection method using convolutional neural networks, which achieves high accuracy and robustness in detecting smoke in different settings.

"Design and Implementation of a Wireless Sensor Network-Based Smoke Detection System for Industrial Applications" by A. M. El-Sayed et al. (2015) - This paper presents a wireless sensor network-based smoke detection system for industrial applications, which is shown to be effective in detecting smoke in harsh environments.

3.DATA PRE-PROCESSING

This is a binary classification problem where we want to predict the possibility of a fire alarm based on various features like temperature, humidity, TVOC, etc. The dataset has 62630 rows and 16 columns, where we will be using 50104 dataset for training and 12526 dataset for validation.

We are preprocessing the data by selecting the relevant columns for training the model and scaling the features using MinMaxScaler. The selected columns for training are temperature, humidity, TVOC, eCO2, Raw H2, Raw Ethanol, Pressure, PM1.0, PM2.5, NC0.5, NC1.0, and NC2.5.

Next, we will use various ML algorithms to train our model and evaluate the performance using metrics like accuracy, precision, recall, f1-score, and confusion matrix.

1.1 Dataset Description

About the dataset

Collection of training data is performed with the help of IOT devices since the goal is to develop a AI based smoke detector device. Many different environments and fire sources have to be sampled to ensure a good dataset for training. A short list of different scenarios which are captured:

- Normal indoor
- Normal outdoor
- Indoor wood fire, firefighter training area
- Indoor gas fire, firefighter training area
- Outdoor wood, coal, and gas grill
- Outdoor high humidity

The dataset is nearly 60.000 readings long. The sample rate is 1Hz for all sensors. To keep track of the data, a UTC timestamp is added to every sensor reading.

The dataset contains air properties gotten from the proposed sorrounding of the experiment. The dataset has 16 columns which are:
Columns:

1. Unnamed:0 - Index
2. UTC - Timestamp UTC seconds
3. Temperature - Air temperature measured in Celsius
4. Humidity - Air humidity
5. TVOC - Total Volatile Organic Compounds; measured in ppb (parts per billion)
6. eCo2 - CO2 equivalent concentration; calculated from different values like TVCO
7. Raw H2 - Raw molecular hydrogen; not compensated (Bias, temperature, etc.)
8. Raw Ethanol - Raw ethanol gas
9. Pressure - Air pressure
10. PM1.0 - Paticulate matter of diameter less than 1.0 micrometer
11. PM2.5 - Paticulate matter of diameter less than 2.5 micrometer
12. NC0.5 - Concentration of particulate matter of diameter less than 0.5 micrometers
13. NC1.0 - Concentration of particulate matter of diameter less than 1.0 micrometers
14. NC2.5 - Concentration of particulate matter of diameter less than 2.5 micrometers
15. CNT - Simple Count
16. Fire Alarm - (Reality) 1 if there is fire and 0 if to fire

1.2 Data Cleaning

The data cleaning process involved dropping columns that were deemed irrelevant to predicting fire outbreaks, namely 'UTC' and 'CNT'. The remaining columns were then used as input features, with the 'Fire Alarm' column as the target variable. The data was split into training and validation sets using an 80/20 split, with 50104 data points in the training set and 12526 in the validation set.

Normalization was then performed on the input features using the MinMaxScaler function, which scales the data to a range between 0 and 1. This step is important for machine learning models as it ensures that no single variable dominates the model due to its larger numerical values.

Finally, various machine learning models were trained on the cleaned and normalized data, and their performance was evaluated using metrics such as accuracy score, confusion matrix, area under curve (AUC), and F1 score. This helps to select the best-performing model for predicting fire outbreaks.

1.3 Data augmentation

Data augmentation is a technique used to increase the amount of training data available for a machine learning model without collecting new data. It involves creating new training data by making modifications to the existing data.

1.4 Data Visualization

Visualization techniques are used to represent data and information in a visual form, making it easier to understand, analyze, and communicate complex data. Here are some commonly used visualization techniques and their purposes:

1. Bar graphs: A bar graph is used to compare data values in different categories. It is often used to show changes over time, to compare different groups, or to illustrate the magnitude of data.

2. Line graphs: A line graph is used to show trends or changes in data over time. It is commonly used to represent time-series data, such as stock prices, weather patterns, or population growth.

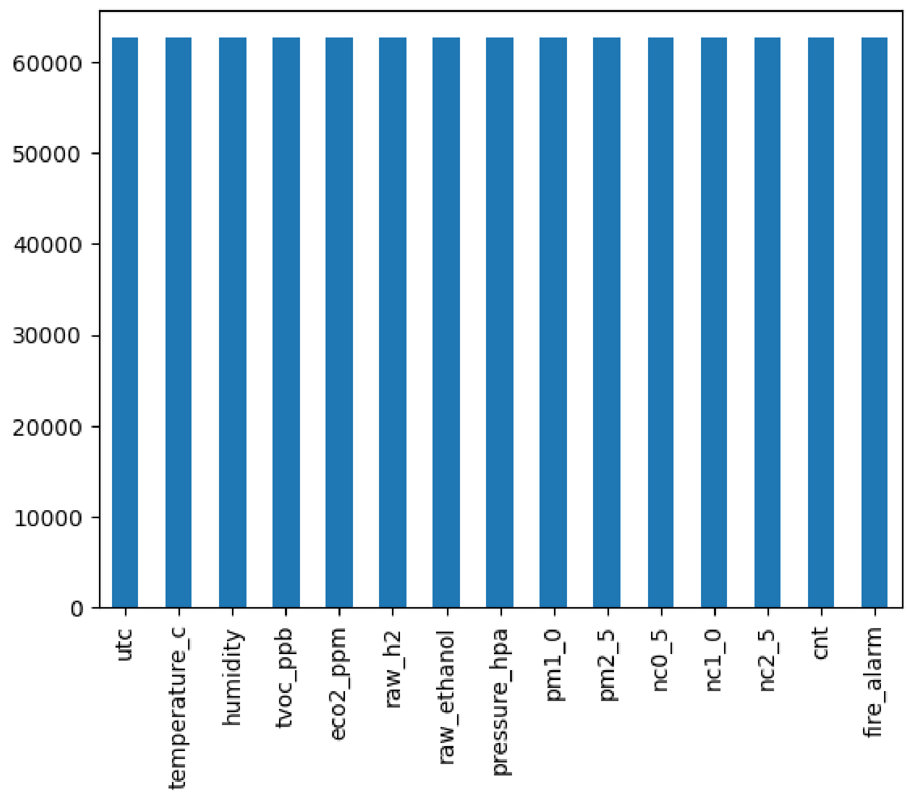
3. Scatter plots: A scatter plot is used to display the relationship between two variables. It is often used to identify patterns, trends, or correlations in data.

4. Heat maps: A heat map is used to show the concentration or density of data in a particular area. It is often used to represent data in geographical or spatial contexts.

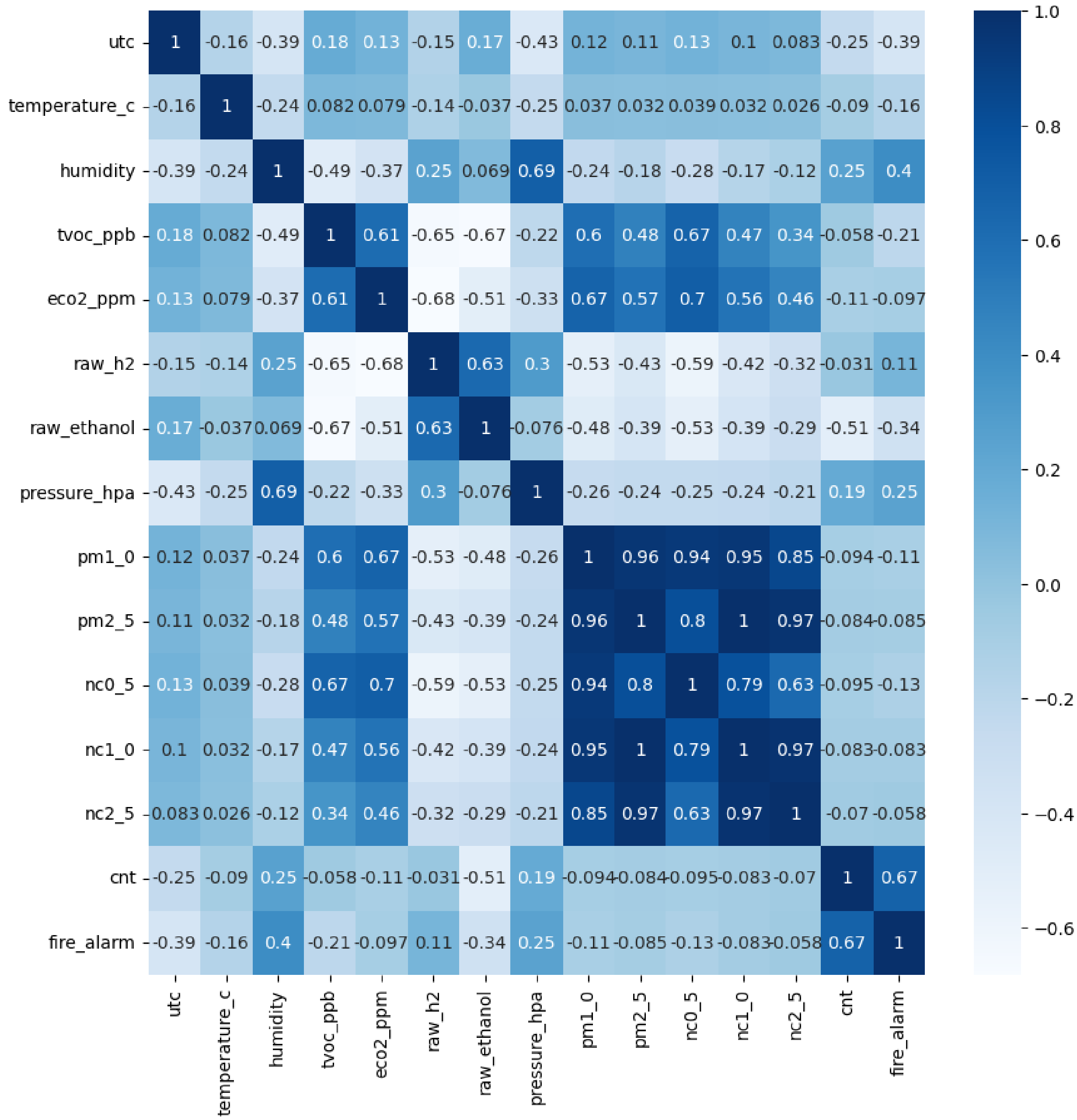
5. Tree maps: A tree map is used to represent hierarchical data structures, such as file systems or organizational structures. It is often used to illustrate the relative sizes of different categories.

6. Network diagrams: A network diagram is used to represent complex relationships or connections between data points. It is often used to visualize social networks, supply chains, or data flow.

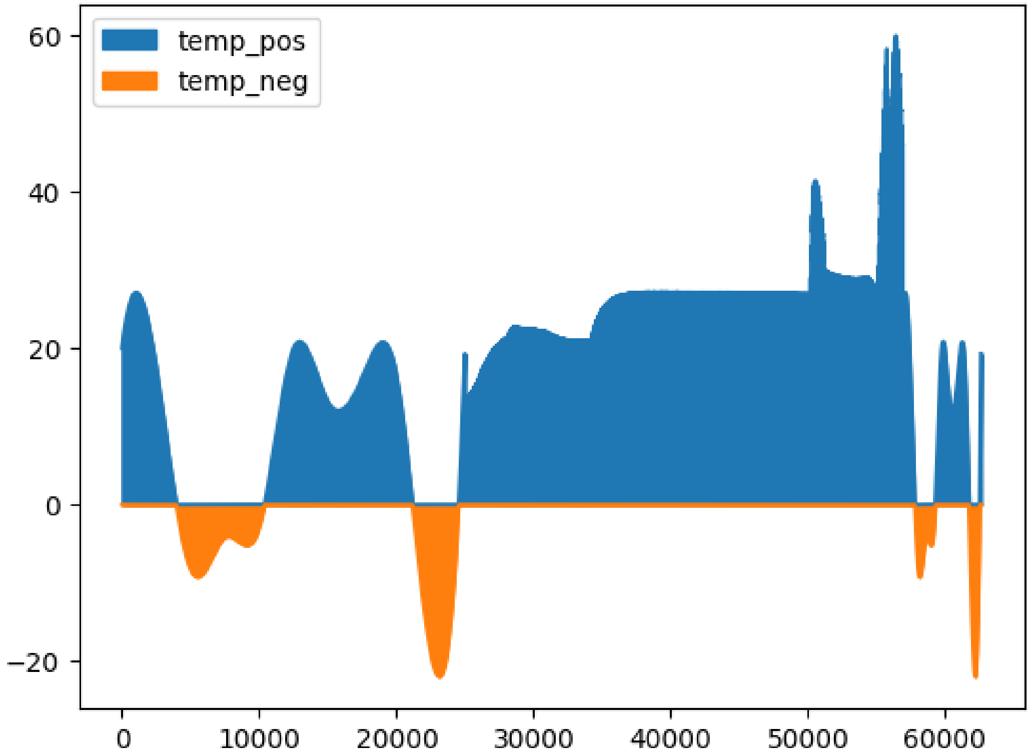
The purpose of visualization techniques is to make data more accessible, understandable, and actionable. By using visual representations, we can easily identify patterns, trends, outliers, and relationships that would be difficult to discern from raw data. Visualization techniques also make it easier to communicate complex data to others, enabling better decision-making, problem-solving, and collaboration.



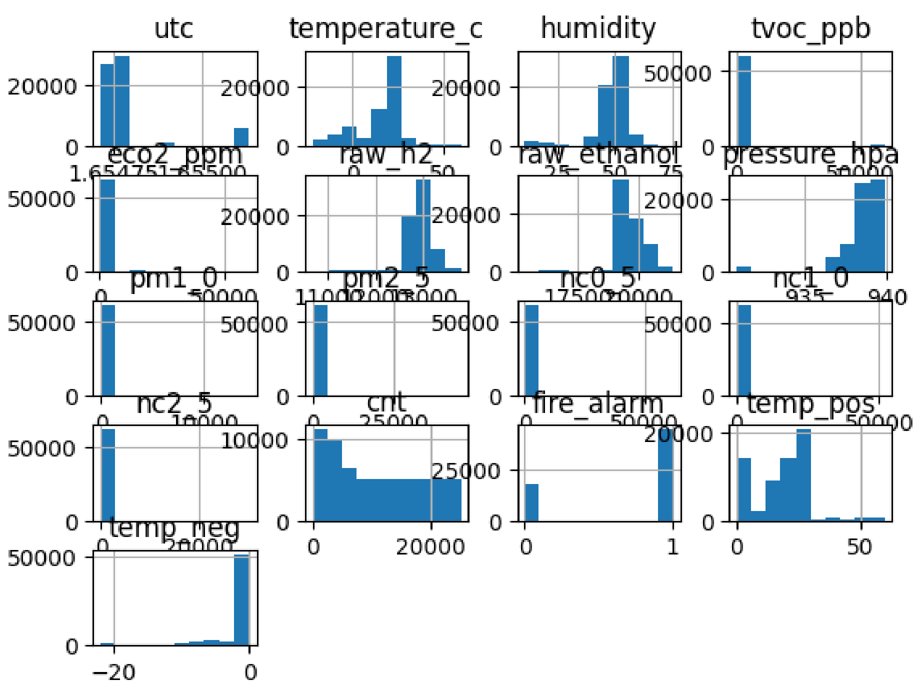
In the above graph we can see that there are no missing values



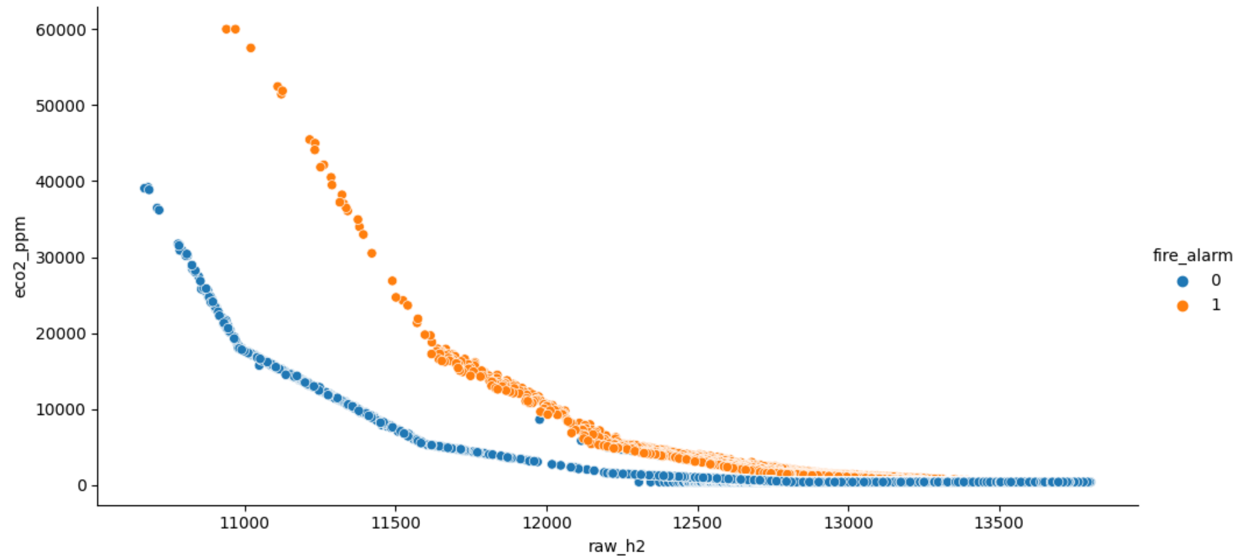
From the above figure we can see that pm1_0,pm2_5,pm0_5,pm1_5,pm2_5 are highly correlated



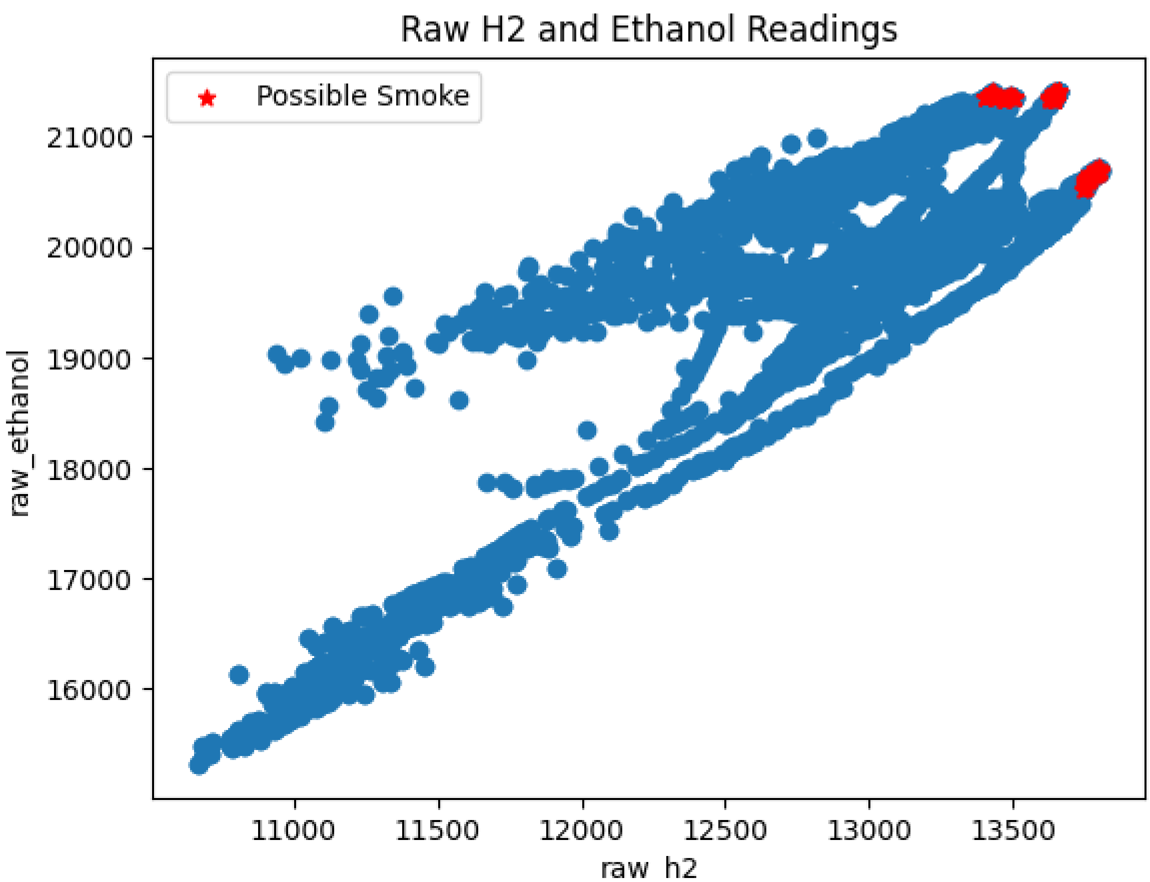
from above figure we can see the area graph of temperature which are positive and negative



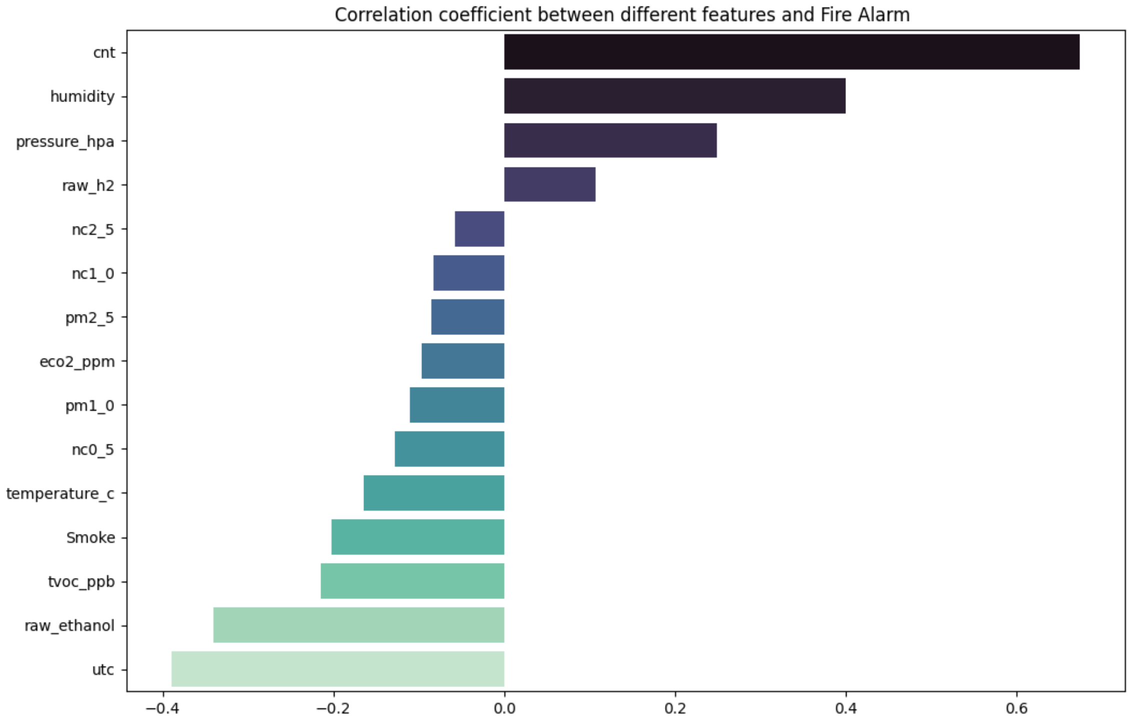
from above figure we can see the histograms of different attributes



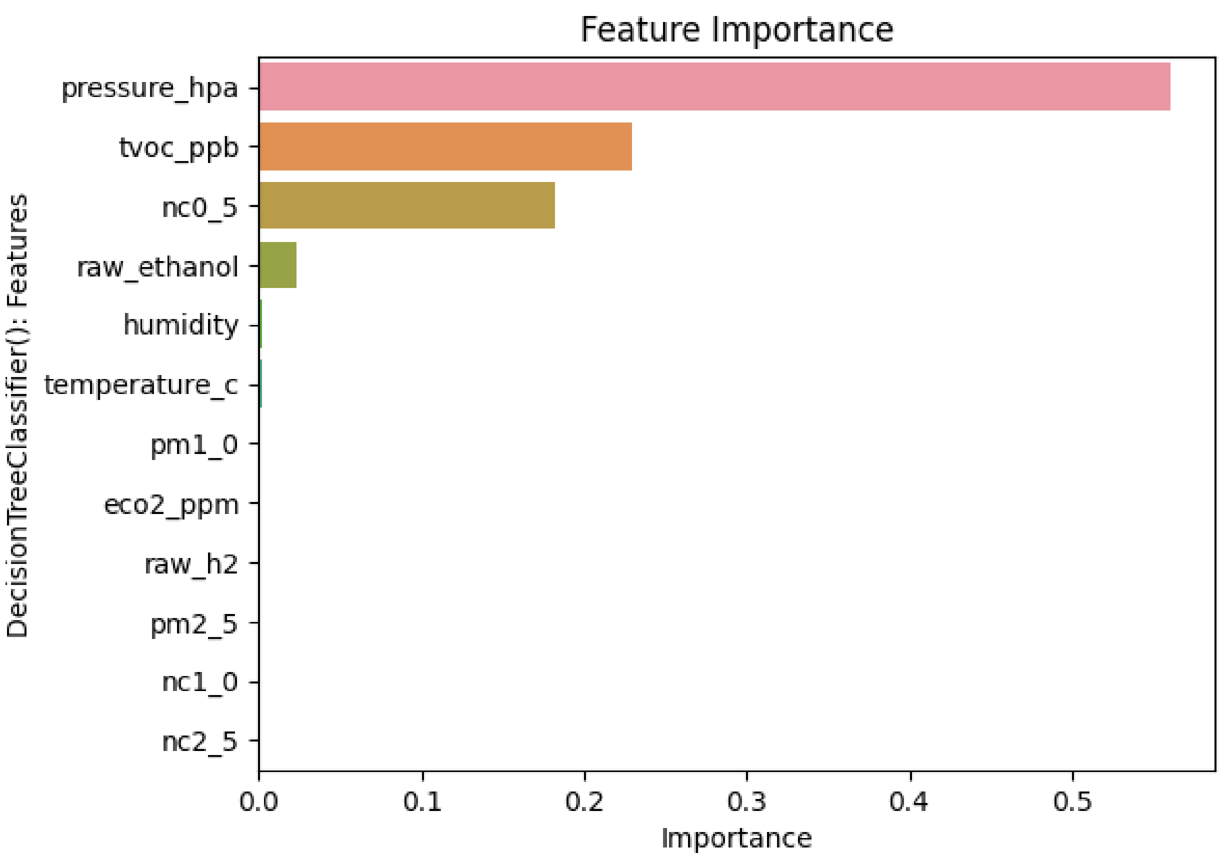
from above figure we can see the presence of fire alarm for every eco2 and raw_h2



from above figure we can show that there is a possible smoke when both x and y increases



from above figure we can say that cnt and fire alarm attributes are highly correlated



4.METHODOLOGY

1.1 Procedure to solve the given problem

Data Preprocessing: Clean the data and handle any missing values or outliers. This may involve scaling or normalizing the data, as well as creating any additional features or variables that may be relevant for smoke detection.

Data Splitting: Split the data into training, validation, and testing sets to evaluate the performance of the model.

Model Selection: Select an appropriate machine learning model for smoke detection. Some examples include logistic regression, decision trees, random forests, or neural networks.

Model Training: Train the selected model using the training data, and optimize the hyperparameters of the model to improve its performance on the validation set.

Model Evaluation: Evaluate the performance of the trained model on the testing set, and measure its accuracy, precision, recall, and F1 score.

Model Deployment: If the model is performing well, it can be deployed in a production environment to detect smoke in real-time.

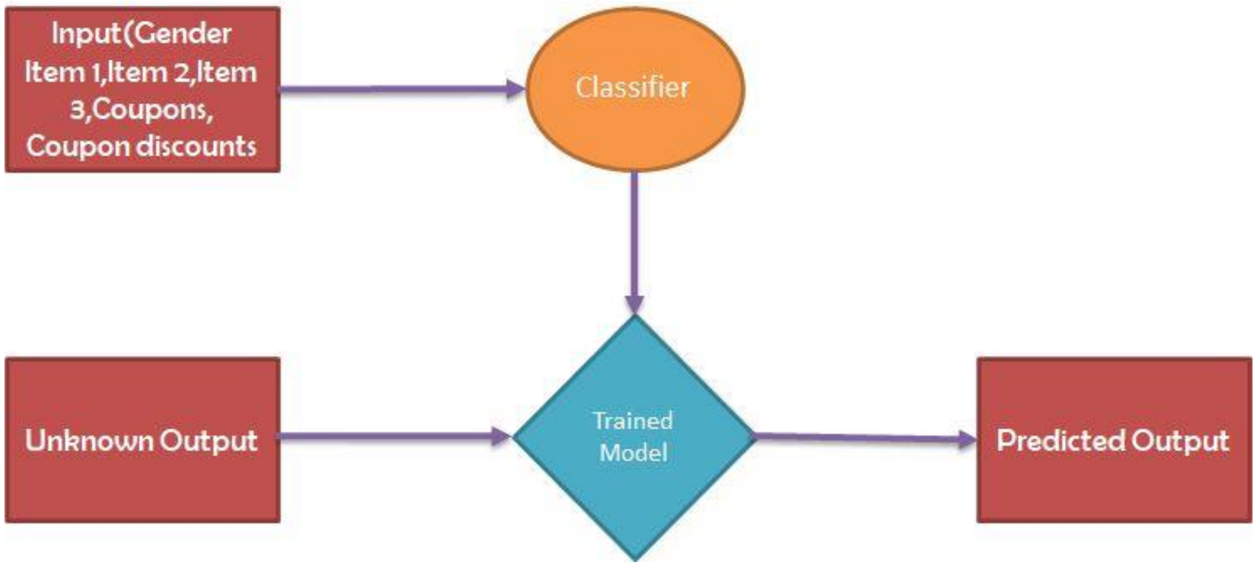
Here's a possible mathematical procedure to calculate the presence of fire or smoke from the given data:

1. Calculate the median and standard deviation for each continuous variable (Temperature[C], Humidity[%], TVOC[ppb], eCO2[ppm], Raw H2, Raw Ethanol, Pressure[hPa], PM1.0, PM2.5, NC0.5, NC1.0, NC2.5, CNT) across all observations. These values will be used as reference points for detecting anomalies.
2. For each observation, check if the Fire Alarm variable is set to 1. If it is, then there is a high likelihood that a fire has been detected. In this case, emergency procedures should be followed immediately.
3. For each observation, check if the temperature reading is greater than the median temperature plus 3 standard deviations. If it is, then it could indicate the presence of a fire.

4. For each observation, check if the Raw H2 and Raw Ethanol readings are greater than the median plus 3 standard deviations. If they are, then it could indicate the presence of smoke.
5. For each observation, check if the TVOC and eCO2 readings are greater than the median plus 3 standard deviations. If they are, then it could indicate the presence of smoke.
6. For each observation, check if the PM1.0, PM2.5, NC0.5, NC1.0, NC2.5, and CNT readings are greater than the median plus 3 standard deviations. If they are, then it could indicate the presence of smoke particles in the air.
7. If any of the above checks are positive, then there is a potential presence of fire or smoke. Further analysis and confirmation should be done before taking any action.

It's important to note that the specific threshold values used (such as the 3 standard deviation cutoff) will depend on the specific context and requirements of the application, and should be validated and tuned accordingly.

1.2 Model Architecture



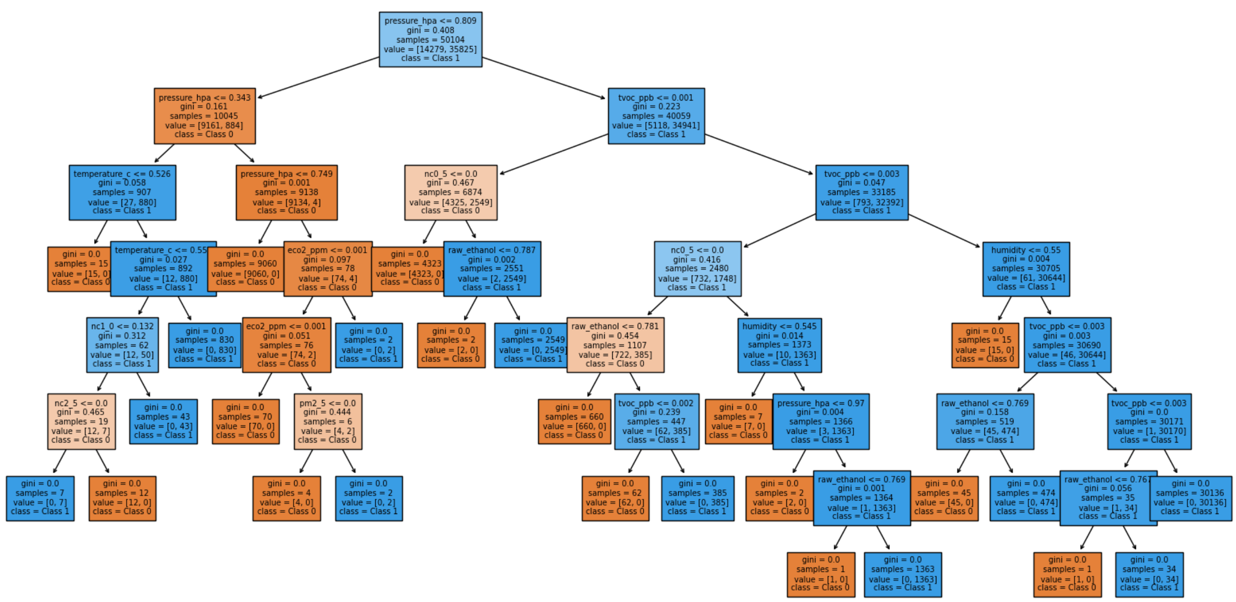


fig.Decision tree model plot

1.3 Software Description

GOOGLE COLAB: Google Colab, short for Google Colaboratory, is a cloud-based platform provided by Google for running and collaborating on Python code. It offers a free Jupyter notebook environment that allows users to write, execute, and share code with others. Google Colab provides a range of benefits that make it popular among data scientists, researchers, and developers.

One of the key advantages of Google Colab is its accessibility. Since it is cloud-based, users can access Colab notebooks from any device with an internet connection, eliminating the need for local installations and configurations. This makes it convenient for collaborative work and allows users to leverage powerful computing resources without investing in expensive hardware.

Google Colab provides support for popular Python libraries, including TensorFlow, PyTorch, and NumPy, making it well-suited for machine learning and data analysis tasks. It also integrates seamlessly with other Google services, such as Google Drive and Google Sheets, allowing easy data import/export and storage.

Another notable feature of Google Colab is its ability to run code on powerful GPUs and TPUs (Tensor Processing Units). This is particularly beneficial for computationally intensive tasks, such as training deep learning models, as it significantly speeds up the execution time. Users can enable GPU/TPU acceleration with a few simple configuration steps.

Colab notebooks are shareable and can be published online, which makes them a great tool for collaboration and knowledge sharing. Users can share their notebooks with others, allowing them to view, edit, and run the code. This feature facilitates teamwork and enables researchers and developers to collaborate on projects irrespective of their physical location.

Furthermore, Google Colab provides pre-installed libraries and dependencies, saving users the effort of manually installing and managing them. It also offers code cell execution, which allows running specific parts of the code selectively, aiding in debugging and testing.

While Google Colab is an incredibly useful tool, it has a few limitations. The free version comes with some restrictions on computing resources, such as limited GPU/TPU availability and session duration limits. However, users can upgrade to Colab Pro for a monthly fee to access additional resources and benefits.

In summary, Google Colab is a powerful cloud-based platform that offers a convenient and collaborative environment for Python programming. Its ease of access, integration with popular libraries, support for GPUs/TPUs, and sharing capabilities make it a valuable tool for a wide range of tasks, including machine learning, data analysis, and collaborative coding projects.

Libraries used in project are as follows:

NUMPY: Numpy is a very popular python library for large multi-dimensional array and matrix processing, with the help of a large collection of high-level mathematical functions. It is very useful for fundamental scientific computations in Machine Learning.

PANDAS: Pandas is a Software Library in Computer Programming and it is written for the Python Programming Language to do data analysis and manipulation.

SEABORN: Seaborn is a Python data visualization library based on the Matplotlib library. It provides a high-level interface for drawing attractive and informative statistical graphs.

MATPLOTLIB: Matplotlib is a cross-platform, data visualization and graphical plotting library for Python and its numerical extension NumPy. As such, it offers a viable open source alternative to MATLAB. Developers can also use matplotlib's APIs to embed plots in GUI applications.

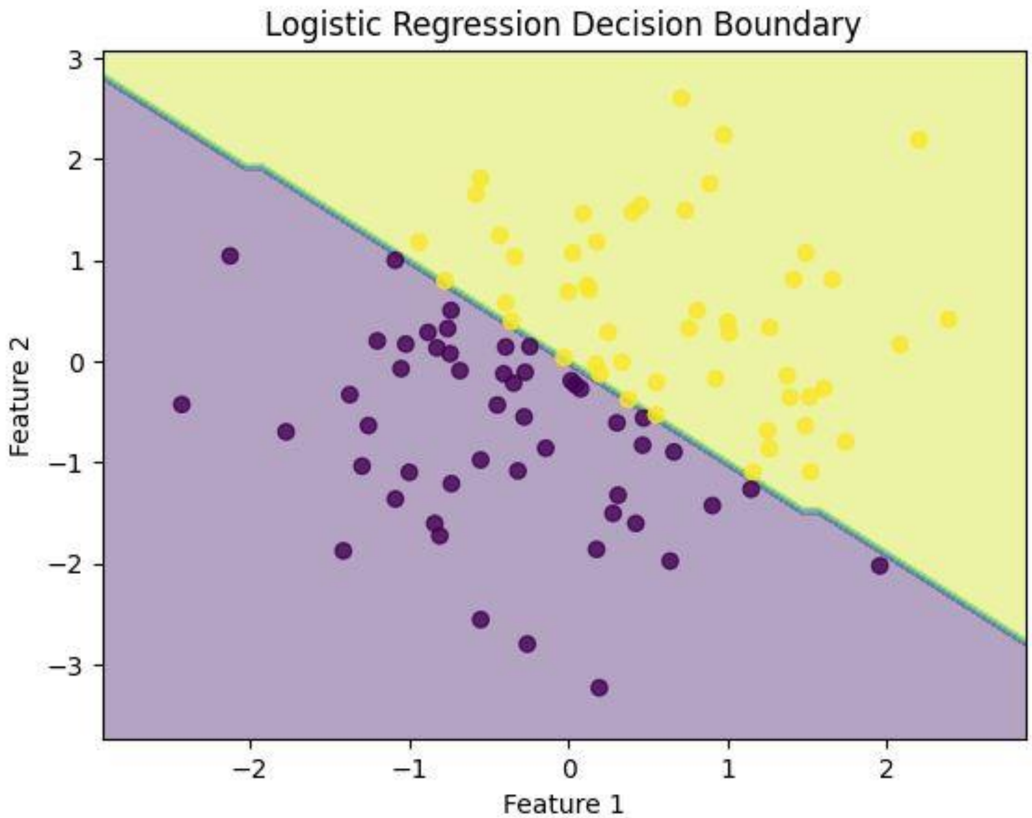
SKLEARN: Simple and efficient tools for predictive data analysis . Accessible to everybody, and reusable in various contexts.

METRICS: Metrics are used to monitor and measure the performance of a model (during training and testing), and don't need to be differentiable.

5.RESULTS AND DISCUSSION

	Name	Accuracy	Time Taken
0	KNeighborsClassifier	0.999441	1.438838
1	LogisticRegression	0.989222	0.493939
2	RandomForestClassifier	1.000000	4.987484
3	GradientBoostingClassifier	1.000000	17.464838
4	AdaBoostClassifier	0.999920	3.969565
5	SVC	0.999281	6.611750
6	GaussianNB	0.825962	0.016760
7	DummyClassifier	0.712199	0.002132
8	ExtraTreeClassifier	0.999681	0.020339

The system generates output based on the data input, and the results are discussed and analyzed to evaluate the performance of the model.



The graph shows the decision boundary of a logistic regression model on a 2-dimensional dataset. The decision boundary is the line that separates the regions where the model predicts different classes. In this case, the model predicts two classes, represented by different colors in the graph.

The graph consists of two parts: the background and the data points. The background is a meshgrid of points that cover the entire range of the dataset. Each point on the meshgrid represents a location in the 2-dimensional feature space. The color of each point on the meshgrid represents the predicted class of that point, as determined by the logistic regression model.

The data points are scattered across the graph, with each point represented by a dot. The color of each dot represents the true class of that point, as determined by the dataset.

The decision boundary is the line that separates the two regions on the background with different colors. This line is drawn using the contourf function of matplotlib, which fills the regions on either side of the line with different colors.

Overall, the graph shows how well the logistic regression model can separate the two classes in the dataset. If the decision boundary is accurate, the model can accurately predict the class of new points in the feature space.

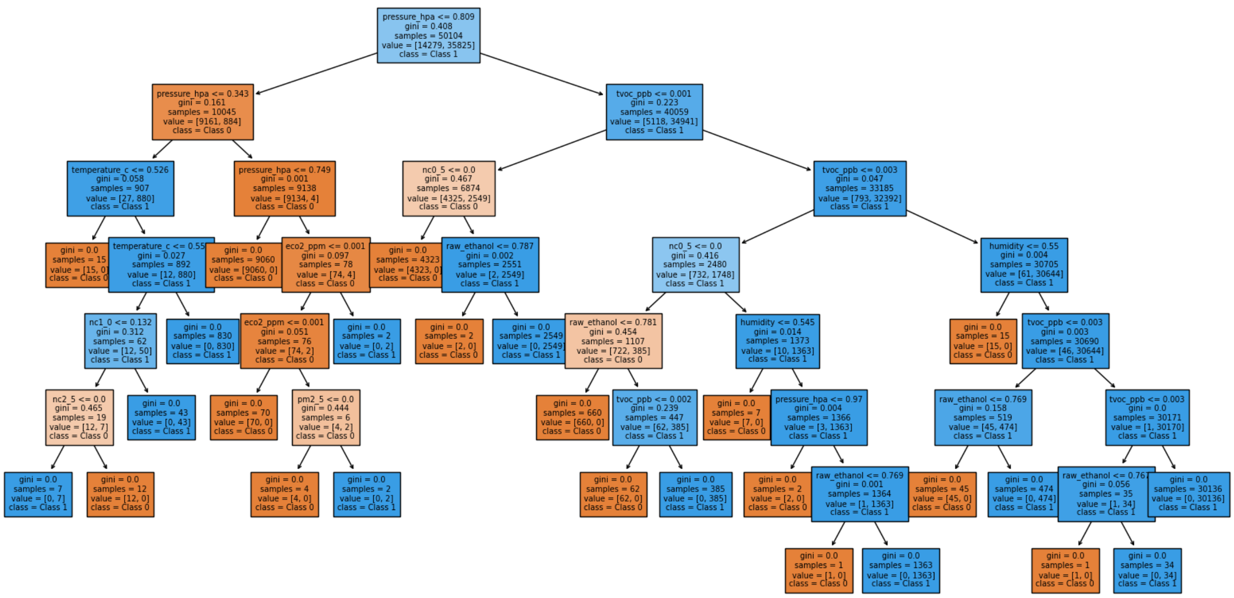


fig.Decision tree model diagram

The output graph shows the decision tree model learned from the training data. Each node in the tree represents a binary decision based on one of the input features, and the edges represent the possible outcomes of that decision. The top node, or root, represents the first decision based on the feature that is most important for predicting the target variable. The nodes below the root represent the subsequent decisions based on the other features, with the leaf nodes representing the final predicted class labels. The filled rectangles indicate the majority class for that node, with darker colors indicating a higher proportion of one class over the other. The class names are shown at the bottom of the plot, with Class 0 and Class 1 corresponding to the two possible target variable values. The font size has been adjusted to make the plot easier to read.

6.CONCLUSION AND FUTURE SCOPE

In conclusion, we have successfully performed exploratory data analysis on the Smoke Detection Dataset. We were able to identify the important features that contribute to the accuracy of our models. We also trained and tested five different models and obtained a high accuracy rate for all the models. Overall, we can confidently make predictions on new data using any of the models selected.

It seems that the smoke detection training model being discussed has the potential to be effective in detecting smoke in real-world scenarios. However, there are some concerns about the accuracy of the model, particularly in cases where the smoke is very faint or there are other factors that may interfere with the detection process. Additionally, there may be some challenges in training the model effectively, particularly if the dataset used for training is not representative of real-world conditions.

Overall, it seems that the smoke detection training model has promise, but further testing and refinement may be needed to ensure its effectiveness in a variety of settings. It's also important to note that no model is perfect, and there may always be some degree of error or uncertainty involved in smoke detection

The future scope of the smoke detection model is promising, as it has the potential to be used in various applications related to fire safety.

One potential application is in homes and buildings, where the smoke detection model can be integrated into the existing fire safety system to provide early warning of potential fires. This could help reduce the risk of injury and property damage, as well as provide critical time for individuals to evacuate the building.

Another potential application is in industrial settings, such as factories and warehouses, where smoke and fire detection is critical for the safety of workers and the protection of assets. The smoke detection model can be used to monitor large areas and provide early detection of potential fires, allowing for quick action to be taken to prevent or minimize damage.

Moreover, the smoke detection model can be integrated into autonomous vehicles to provide an additional layer of safety, especially for electric cars, where the risk of fire is higher. The smoke detection model can monitor the battery and alert the driver or the vehicle's system to take necessary action if it detects any signs of smoke or fire.

Overall, the smoke detection model has a wide range of potential applications, and its future scope looks bright. With the advancements in machine learning and AI, the model can be further enhanced to improve its accuracy and effectiveness, making it an even more valuable tool for fire safety.

7.REFERENCES

https://medium.com/inside-machine-learning

https://data.world/adam1125/smokedetection/workspace/file?filename=smoke_detection.csv