PROJECT ON

SMOKE DETECTION

GUIDED BY: NARESH KUMAR

TEAM MEMBERS

NAME:	HTNO:
G VAMSHI	2103A51403
L VIGNESH	2103A51271
P SATHWIK	2103A51490

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.

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.

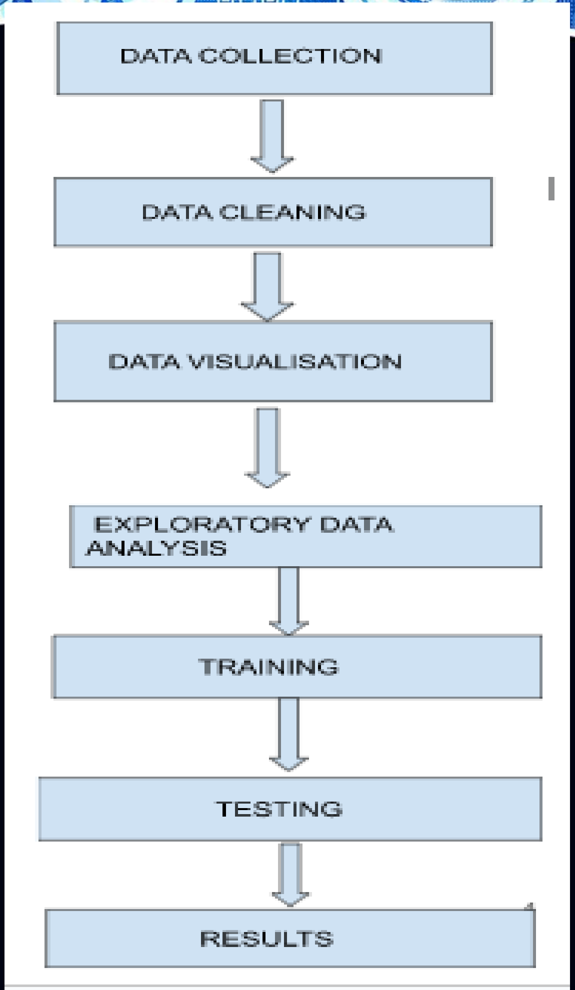
PROPOSED SYSTEM

- **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

OBJECTIVES

- The objectives of smoke detection in machine learning can be defined as follows:
  - Temperature and humidity monitoring:
  - Air quality monitoring:
  - Fire detection:
  - Real-time monitoring:

OVERALL ARCHITECTURE



DATASET

column_a	utc	temperatu	humidity	tvoc_ppb	eco2_ppm	raw_h2	raw_ethar	pressure_	pm1_0	pm2_5	nc0_5	nc1_0	nc2_5	cnt	fire_alarm
0	1.65E+09	20	57.36	0	400	12306	18520	939.735	0	0	0	0	0	0	0
1	1.65E+09	20.015	56.67	0	400	12345	18651	939.744	0	0	0	0	0	1	0
2	1.65E+09	20.029	55.96	0	400	12374	18764	939.738	0	0	0	0	0	2	0
3	1.65E+09	20.044	55.28	0	400	12390	18849	939.736	0	0	0	0	0	3	0
4	1.65E+09	20.059	54.69	0	400	12403	18921	939.744	0	0	0	0	0	4	0
5	1.65E+09	20.073	54.12	0	400	12419	18998	939.725	0	0	0	0	0	5	0
6	1.65E+09	20.088	53.61	0	400	12432	19058	939.738	0	0	0	0	0	6	0
7	1.65E+09	20.103	53.2	0	400	12439	19114	939.758	0	0	0	0	0	7	0
8	1.65E+09	20.117	52.81	0	400	12448	19155	939.758	0	0	0	0	0	8	0
9	1.65E+09	20.132	52.46	0	400	12453	19195	939.756	0.9	3.78	0	4.369	2.78	9	0
10	1.65E+09	20.146	52.15	0	400	12454	19230	939.757	0.89	3.71	0	4.289	2.73	10	0
11	1.65E+09	20.161	51.84	0	400	12467	19264	939.754	0.84	3.51	0	4.053	2.58	11	0
12	1.65E+09	20.175	51.62	0	400	12467	19299	939.755	0.81	3.38	0	3.909	2.488	12	0
13	1.65E+09	20.19	51.39	0	400	12469	19317	939.758	0.74	3.11	0	3.588	2.284	13	0
14	1.65E+09	20.204	51.17	0	403	12468	19338	939.742	0.71	2.96	0	3.419	2.176	14	0
15	1.65E+09	20.219	50.99	0	400	12475	19362	939.741	0.64	2.66	0	3.077	1.959	15	0
16	1.65E+09	20.233	50.86	0	400	12480	19382	939.758	0.6	2.52	0	2.908	1.851	16	0
17	1.65E+09	20.248	50.66	0	400	12477	19400	939.764	0.53	2.23	0	2.58	1.642	17	0
18	1.65E+09	20.262	50.49	0	400	12481	19422	939.761	0.5	2.1	0	2.423	1.542	18	0
19	1.65E+09	20.277	50.27	0	406	12489	19451	939.752	0.41	1.72	0	1.987	1.265	19	0

DESCRIBE DATASET

1. UTC: A variable that contains the time stamp for each observation.
2. Temperature[C]: A continuous variable that contains the temperature readings in degrees Celsius.
3. Humidity[%]: A continuous variable that contains the humidity readings as a percentage.
4. TVOC[ppb]: A continuous variable that contains the total volatile organic compounds readings in parts per billion.
5. eCO2[ppm]: A continuous variable that contains the equivalent CO2 readings in parts per million.
6. Raw H2: A continuous variable that contains the raw hydrogen sensor readings.
7. Raw Ethanol: A continuous variable that contains the raw ethanol sensor readings.
8. Pressure[hPa]: A continuous variable that contains the pressure readings in hectopascals.
9. PM1.0: A continuous variable that contains the readings for particles with a diameter of 1.0 micrometers or smaller.

DESCRIBE DATASET

10. PM2.5: A continuous variable that contains the readings for particles with a diameter of 2.5 micrometers or smaller.
11. NC0.5: A continuous variable that contains the number concentration readings for particles with a diameter of 0.5 micrometers or smaller.
12. NC1.0: A continuous variable that contains the number concentration readings for particles with a diameter of 1.0 micrometers or smaller.
13. NC2.5: A continuous variable that contains the number concentration readings for particles with a diameter of 2.5 micrometers or smaller.
14. CNT: A continuous variable that contains the particle count readings.
15. Fire Alarm: A categorical variable that indicates the presence of a fire alarm (1) or not (0).

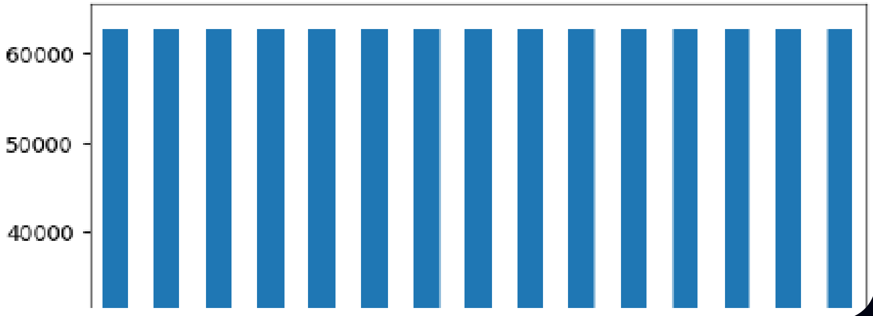
DATA VISUALIZATION

```
data.count().plot.barh()
```

```
<Axes: >
```


```
data.count().plot.bar()
```

```
<Axes: >
```

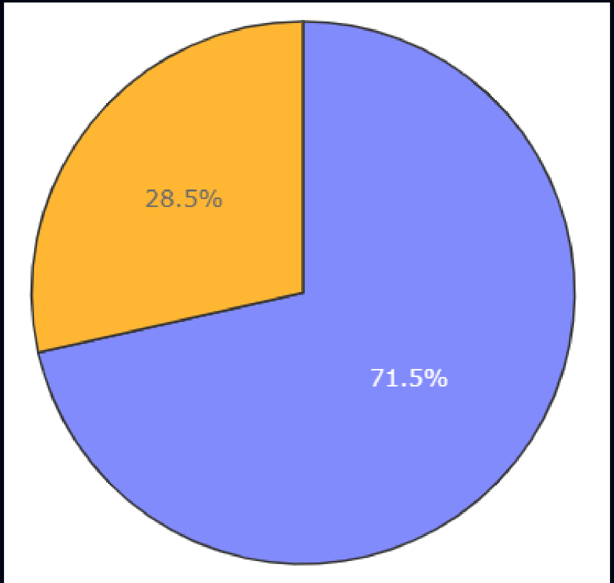



```
sns.heatmap(data.isnull(), cmap='viridis')
```

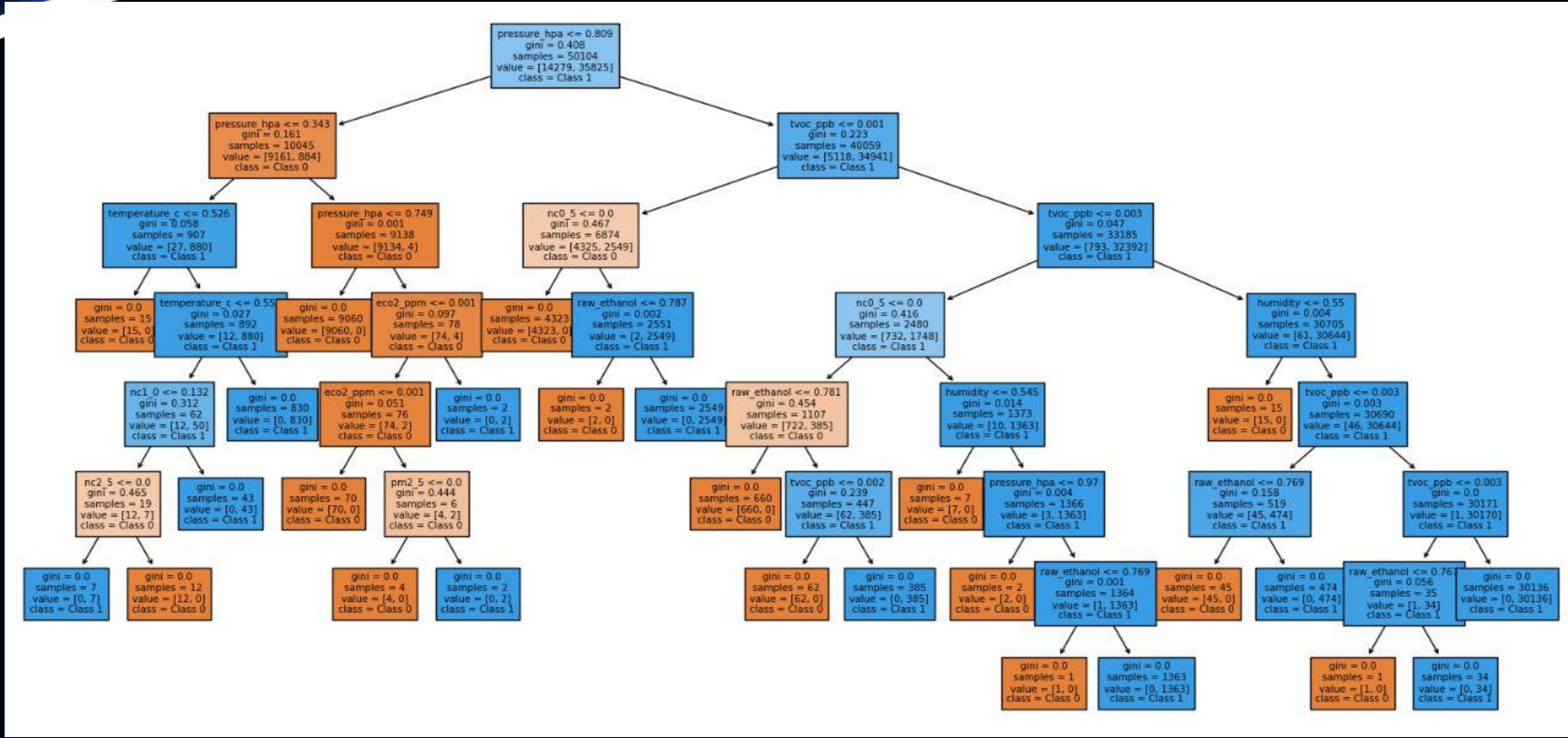
```
<Axes: >
```

DATA VISUALIZATION

The Pie chart represents the distribution of the 'fire_alarm' variable in the 'data' dataframe. The 'labels' parameter in the function contains two values - 'Yes_Fire' and 'No_Fire', which are the labels for the two slices of the Pie chart. The 'values' parameter contains the count of 'fire_alarm' values that are 'Yes_Fire' and 'No_Fire' respectively.



MODEL ARCHITECTURE



DATA NORMALIZATION

```
scaler = MinMaxScaler().fit(data[input_cols])

X_train[input_cols] = scaler.transform(X_train[input_cols])
X_test[input_cols] = scaler.transform(X_test[input_cols])
```

```
X_test.describe().loc[['min', 'max']]
```

	temperature_c	humidity	tvoc_ppb	eco2_ppm	raw_h2	raw_ethanol	pressure_hpa	pm1_0	pm2_5	nc0_5	nc1_0	nc2_5
min	0.0	0.001706	0.0	0.000000	0.004785	0.025111	0.001110	0.000000	0.000000	0.00000	0.000000	0.000000
max	1.0	0.998138	1.0	0.856493	0.991707	1.000000	0.999001	0.991087	0.963304	0.99911	0.961897	0.949159

TRAINING MODEL

```
def train_acc_model(models):
    model_name_list = []
    confusion_metrx_list = []
    execution_time_list = []
    model_score_list = []
    auc_list = []
    f1_score_list = []
    accuracy_score_list = []
    #print('Training dataset Evaluation with different metrix')

    for model in models:

        start = time.time()
        model.fit(X_train, y_train.values.ravel())
        end = time.time()

        model_name_list.append(type(model).__name__)

        #model prediction
        train_preds = model.predict(X_train)
```

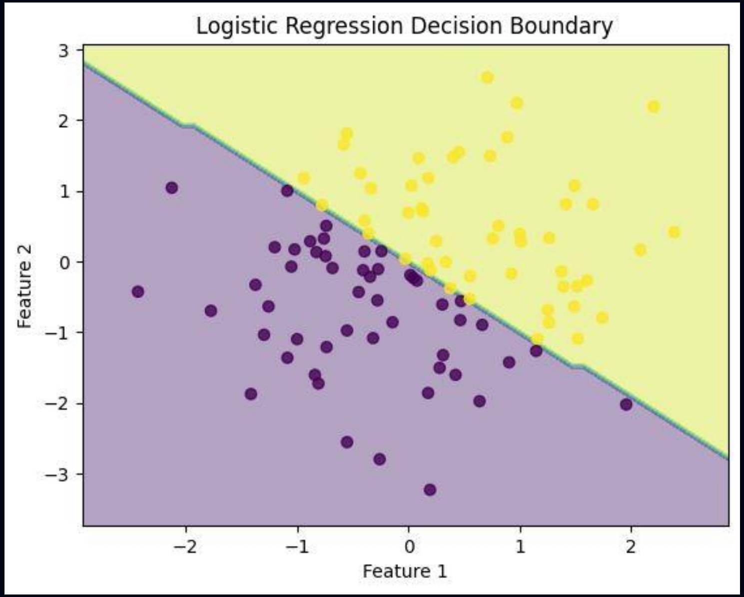
TESTING MODEL

```
def test_acc_model(models):
    model_name_list = []
    confusion_metrx_list = []
    execution_time_list = []
    model_score_list = []
    auc_list = []
    f1_score_list = []
    accuracy_score_list = []
    feature_importance = []
    feature_imp_name = []

    #print('Evaluating Validation dataset with different metrix')

    for model in models:
        start = time.time()
        model.fit(X_train, y_train.values.ravel())
        end = time.time()
        try:
            feature_importance.append(pd.DataFrame({
                (str(model) + ': Features'): X_train.columns,
                'Importance': model.feature_importances_
            }).sort_values('Importance', ascending=False))
        except:
            pass
```

RESULTS



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.

CONCLUSION

we have successfully performed exploratory data analysis on the Smoke 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 for Logistic Regression. Overall, we can confidently make predictions on new data using any of the models selected.

REFERENCES

https://medium.com/inside-machine-learning
https://data.world/adam1125/smokedetection/workspace/file?filename=smoke_detection.csv