NETAJI SUBHAS UNIVERSITY OF TECHNOLOGY

COMPUTER HARDWARE SOFTWARE WORKSHOP

COCSC19

SUBMITTED BY:

YASH CHAUHAN- 2021UCS1628

KHUSHI KHARKE- 2021UCS1617

ADITYA YADAV- 2021UCS1640

TINY ML

PROJECT TITLE: AQI DETECTION

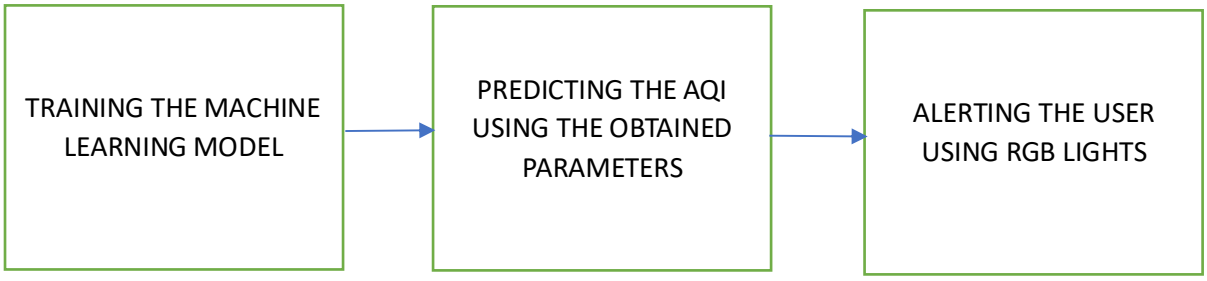
IDEA:

Develop a machine learning model capable of forecasting the Air Quality Index (AQI) for a specific location by leveraging key atmospheric parameters such as PM2.5, PM10, NO2, and SO2 concentrations.

TOOLS USED:

- Arduino uno simulator
- Juptyter notebook

FLOW DIAGRAM:



DATA:

To train the machine learning model we used a dataset consisting amounts of key atmospheric parameters like PM2.5, PM10, NO2, and SO2 concentrations from various cities across India since 2015 to 2020

	PM2.5	PM10	NO2	SO2	AQI
2123	81.40	124.50	20.50	15.24	184.0
2124	78.32	129.06	26.00	26.96	197.0
2125	88.76	135.32	30.85	33.59	198.0
2126	64.18	104.09	28.07	19.00	188.0
2127	72.47	114.84	23.20	10.55	173.0
...	...	...	...	...	...
29525	7.63	32.27	23.27	6.87	47.0
29526	15.02	50.94	25.06	8.55	41.0
29527	24.38	74.09	26.06	12.72	70.0
29528	22.91	65.73	29.53	8.42	68.0
29529	16.64	49.97	29.26	9.84	54.0

PYTHON CODE:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn import linear_model
```

```
combined_data = pd.read_csv('city_day.csv')
```

```
combined_data.head()
```

```
        City        Date  PM2.5  PM10     NO    NO2    NOx  NH3     CO    SO2  \
0  Ahmedabad  2015-01-01    NaN   NaN   0.92  18.22  17.15  NaN   0.92  27.64
1  Ahmedabad  2015-01-02    NaN   NaN   0.97  15.69  16.46  NaN   0.97  24.55
2  Ahmedabad  2015-01-03    NaN   NaN  17.40  19.30  29.70  NaN  17.40  29.07
3  Ahmedabad  2015-01-04    NaN   NaN   1.70  18.48  17.97  NaN   1.70  18.59
4  Ahmedabad  2015-01-05    NaN   NaN  22.10  21.42  37.76  NaN  22.10  39.33

       O3  Benzene  Toluene  Xylene  AQI AQI_Bucket
0  133.36     0.00     0.02    0.00  NaN        NaN
1   34.06     3.68     5.50    3.77  NaN        NaN
2   30.70     6.80    16.40    2.25  NaN        NaN
3   36.08     4.43    10.14    1.00  NaN        NaN
4   39.31     7.01    18.89    2.78  NaN        NaN
```

```
combined_data=combined_data.dropna()
```

```
X= combined_data[['PM2.5','PM10','NO2','SO2']]
Y= combined_data['AQI']
```

```
regr=linear_model.LinearRegression()
regr.fit(X,Y)
```

```
LinearRegression()
```

```
print(regr.coef_)
```

```
[ 0.89991892  0.48236037  0.0356627  -0.04208933]
```

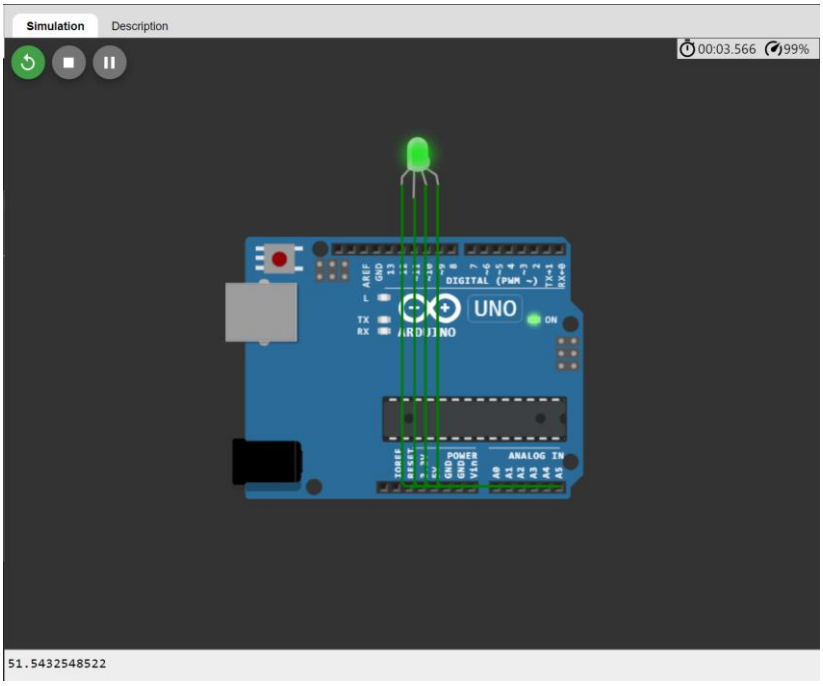
```
print(regr.intercept_)
```

```
25.142174108944573
```

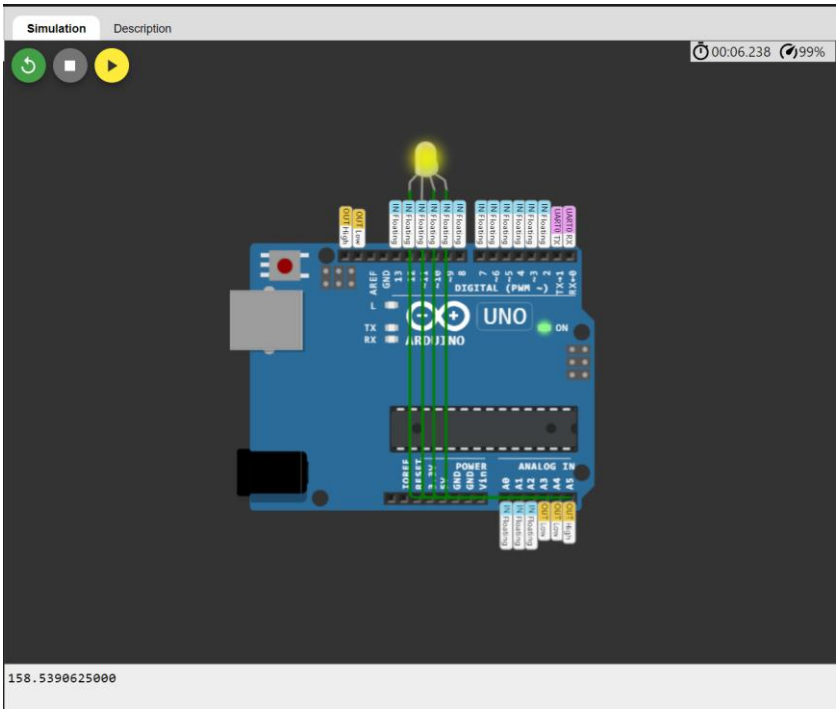
ARDUINO CODE:

```
int redPin = A3;
int greenPin = A4;
int bluePin = A5;
float output=0;
float lrCoef[5] =
{25.142174108944573,  0.89991892,  0.48236037,  0.0356627,  -0.04208933};
void setup() {


pinMode(A0, INPUT);
  pinMode(A1, INPUT);
  pinMode(A2, INPUT);
  pinMode(A3, INPUT);


pinMode(redPin, OUTPUT);
  pinMode(greenPin, OUTPUT);
  pinMode(bluePin, OUTPUT);


Serial.begin(9600);
  // delay(1000);

  float val1 = 81.40;
  float val2 = 124.50;
  float
val3 = 20.50;
  float val4 = 15.24;

  output = multiLinReg(val1, val2, val3, val4);

Serial.print(output, 10);

}

void loop() {
  if(output<=100){
    analogWrite(redPin,
255);
    analogWrite(bluePin, 255);
  }
  else if(output<=200){
    analogWrite(bluePin,
255);
  }
  else{
    analogWrite(greenPin, 255);
    analogWrite(bluePin, 255);
  }

delay(250);
  analogWrite(redPin, 0);
  analogWrite(bluePin, 0);
  analogWrite(greenPin, 0);

delay(250);

}

float multiLinReg(float a, float b, float c, float d) {
  return lrCoef[0] +
a * lrCoef[1] + b * lrCoef[2] + c * lrCoef[3] + d * lrCoef[4];
}
```

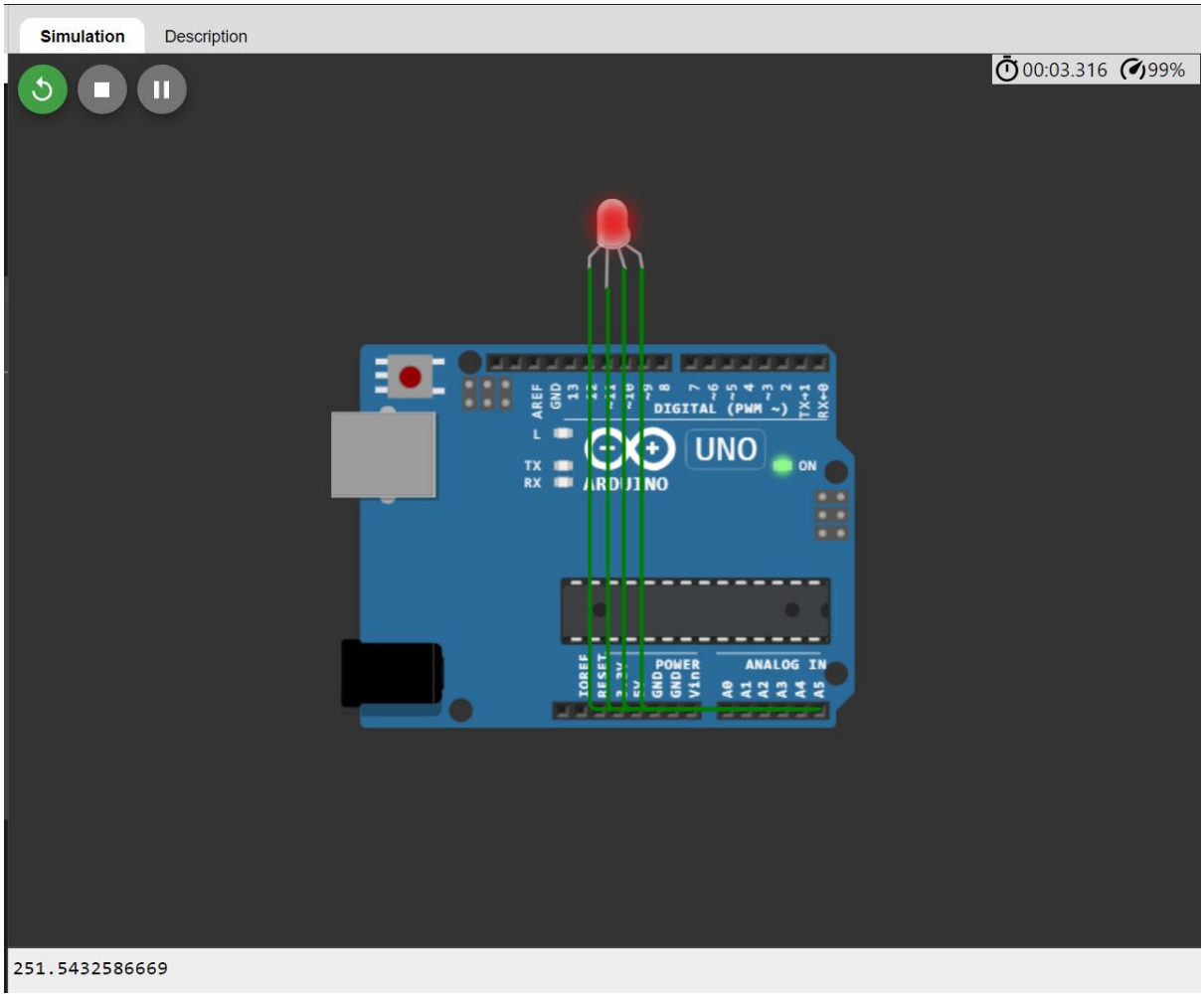
ARDUINO SIMULATION:



GREEN AQI RANGE: 0-100



YELLOW AQI RANGE: 100-200



RED AQI RANGE: >200