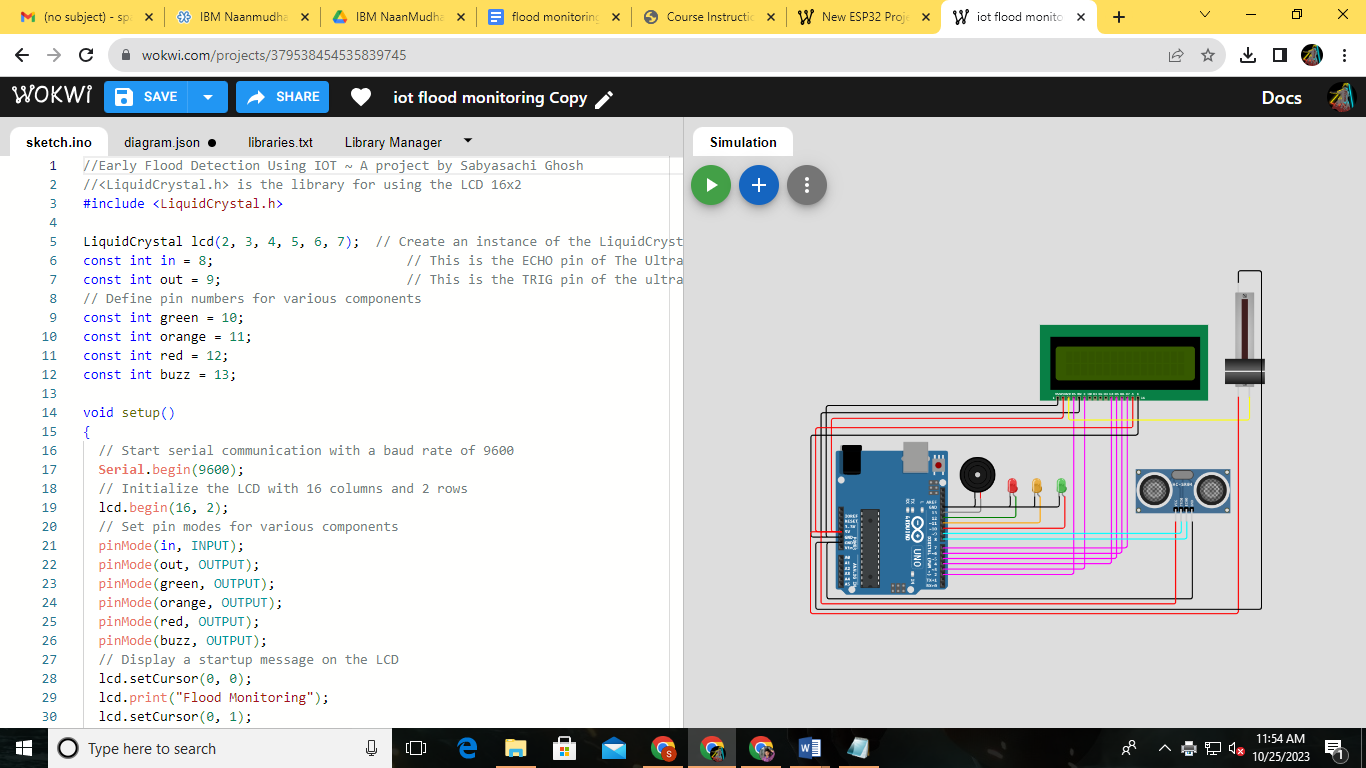
**FLOOD MONITORING AND EARLY WARNING**

Certainly! When working on a machine learning project, there are several key activities that you'll typically go through. Here are some of the fundamental steps, including feature engineering, model training, and evaluation:

1. **Problem Definition:**
   * Understand the problem you're trying to solve. Define your problem statement and objectives.
2. **Data Collection:**
   * Gather relevant data for your problem. This might involve web scraping, using existing datasets, or collecting data through surveys and experiments.
3. **Data Preprocessing:**
   * Clean the data by handling missing values, outliers, and formatting issues.
   * Data normalization and scaling to make the data suitable for modeling.
4. **Feature Engineering:**
   * Create or select appropriate features (input variables) for your machine learning model. This can involve domain-specific knowledge and creativity.
5. **Data Splitting:**
   * Split the dataset into training, validation, and test sets. The training set is used to train the model, the validation set for hyperparameter tuning, and the test set to evaluate the model's performance.
6. **Model Selection:**
   * Choose the appropriate machine learning algorithm(s) based on the nature of your problem (classification, regression, clustering, etc.).
7. **Model Training:**
   * Train the chosen model(s) on the training data using a chosen algorithm. This involves optimizing model parameters to minimize the loss function.
8. **Hyperparameter Tuning:**
   * Optimize hyperparameters to fine-tune the model's performance. This is typically done using techniques like grid search or random search.
9. **Model Evaluation:**
   * Assess the model's performance using various evaluation metrics. Common metrics include accuracy, precision, recall, F1 score, and mean squared error (MSE), among others.
10. **Model Interpretability (optional):**
    * Understand the model's decision-making process. Techniques like feature importance analysis can be used to gain insights.
11. **Model Deployment (if applicable):**
    * Deploy the trained model in a production environment for real-time predictions. This might involve creating APIs or integrating the model into an application.
12. **Monitoring and Maintenance (if deployed):**
    * Continuously monitor the model's performance in production and retrain it as needed.
13. **Documentation and Reporting:**
    * Document all your work, including data sources, preprocessing steps, model details, and evaluation results. This is important for transparency and reproducibility.
14. **Iterative Process:**
    * Machine learning is often an iterative process. You may need to go back to previous steps, such as feature engineering or hyperparameter tuning, to improve your model.

Each of these activities is critical in the machine learning workflow, and they often overlap and interact with one another. Effective feature engineering, for instance, can significantly impact model training and evaluation. Additionally, the choice of evaluation metrics depends on the nature of the problem and the goals of your project



//Early Flood Detection Using IOT ~ A project by Sabyasachi Ghosh

//<LiquidCrystal.h> is the library for using the LCD 16x2

#include <LiquidCrystal.h>

LiquidCrystal lcd(2, 3, 4, 5, 6, 7);  // Create an instance of the LiquidCrystal library

const int in = 8;                         // This is the ECHO pin of The Ultrasonic sensor HC-SR04

const int out = 9;                        // This is the TRIG pin of the ultrasonic Sensor HC-SR04

// Define pin numbers for various components

const int green = 10;

const int orange = 11;

const int red = 12;

const int buzz = 13;

void setup()

{

  // Start serial communication with a baud rate of 9600

**Serial**.begin(9600);

  // Initialize the LCD with 16 columns and 2 rows

  lcd.begin(16, 2);

  // Set pin modes for various components

  pinMode(in, INPUT);

  pinMode(out, OUTPUT);

  pinMode(green, OUTPUT);

  pinMode(orange, OUTPUT);

  pinMode(red, OUTPUT);

  pinMode(buzz, OUTPUT);

  // Display a startup message on the LCD

  lcd.setCursor(0, 0);

  lcd.print("Flood Monitoring");

  lcd.setCursor(0, 1);

  lcd.print("Alerting System");

  // Wait for 5 seconds and then clear the LCD

  delay(5000);

  lcd.clear();

}

void loop()

{

  // Read distance from the ultrasonic sensor (HC-SR04)

  long dur;

  long dist;

  long per;

  digitalWrite(out, LOW);

  delayMicroseconds(2);

  digitalWrite(out, HIGH);

  delayMicroseconds(10);

  digitalWrite(out, LOW);

  dur = pulseIn(in, HIGH);

  dist = (dur \* 0.034) / 2;

  // Map the distance value to a percentage value

  per = map(dist, 10.5, 2, 0, 100);

  // Ensure that the percentage value is within bounds

  if (per < 0)

  {

    per = 0;

  }

  if (per > 100)

  {

    per = 100;

  }

  // Print water level data to serial

**Serial**.print("Water Level:");

**Serial**.println(String(per));

  lcd.setCursor(0, 0);

  lcd.print("Water Level:");

  lcd.print(String(per));

  lcd.print("%  ");

  // Check water level and set alert levels

  if (dist <= 3)

  {

    lcd.setCursor(0, 1);

    lcd.print("Red Alert!   ");

    digitalWrite(red, HIGH);

    digitalWrite(green, LOW);

    digitalWrite(orange, LOW);

    digitalWrite(buzz, HIGH);

    delay(2000);

    digitalWrite(buzz, LOW);

    delay(2000);

    digitalWrite(buzz, HIGH);

    delay(2000);

    digitalWrite(buzz, LOW);

    delay(2000);

  }

  else if (dist <= 10)

  {

    lcd.setCursor(0, 1);

    lcd.print("Orange Alert!  ");

    digitalWrite(orange, HIGH);

    digitalWrite(red, LOW);

    digitalWrite(green, LOW);

    digitalWrite(buzz, HIGH);

    delay(3000);

    digitalWrite(buzz, LOW);

    delay(3000);

  }

  else

  {

    lcd.setCursor(0, 1);

    lcd.print("Green Alert!  ");

    digitalWrite(green, HIGH);

    digitalWrite(orange, LOW);

    digitalWrite(red, LOW);

    digitalWrite(buzz, LOW);

  }

}