

# kNN, Linear regression, and multilinear regression

Eduard Romero / Juan David Cepeda / Carlos Bejarano

2023-04-18

## 1.1 kNN, Linear regression, and multilinear regression

For the acquisition of our data, we will be using the Arduino micro controller with two sensors, an HC-SR04 ultrasonic sensor and the other an analog SHARP 0A41SK sensor.

The HC-SR04 ultrasonic sensor works by sending out high frequency sound waves and then measuring the time it takes for those sound waves to bounce back off of an object and return to the sensor. The sensor consists of two main components: a transmitter and a receiver.

### Code Arduino for sensor Ultrasonic HC-SR04

```
# define pins for HC-SR04
#define trigPin 9
#define echoPin 10

void setup() {
  # initialize serial communication
  Serial.begin(9600);

  # define pin modes
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
}

void loop() {
  # trigger the sensor by setting the trigPin high for 10 microseconds
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  # read the echoPin and calculate the distance
  long duration = pulseIn(echoPin, HIGH);
  float distance_cm = duration * 0.034 / 2;

  # print the distance to the serial monitor
  Serial.print("Distance: ");
  Serial.print(distance_cm);
  Serial.println(" cm");

  #wait for a moment before taking another measurement
  delay(500);
}
```

The Sharp GP2Y0A21 infrared sensor works by emitting an infrared (IR) beam and then measuring the

distance to an object based on the reflection of that beam. The sensor consists of an IR emitter and a receiver.

### Code Arduino for sensor SHARP GP2Y0A21

```
void setup() {
  // Comunicación serial a 9600 baudios
  Serial.begin(9600);
  pinMode(13, OUTPUT);
}

void loop() {

  long tiempo=millis(); //tiempo antes de iniciar la lectura
  int D_cm=distancia(20); //lectura de distancia
  tiempo=millis()-tiempo; //milisegundos que duró la lectura
  Serial.print("Tiempo de lectura: ");
  Serial.print(tiempo);
  Serial.print("ms Distancia: ");
  Serial.print(D_cm);
  Serial.println(" cm");
  delay(100);
}

float distancia(int n)
{
  long suma=0;
  for(int i=0;i<n;i++)
  {
    suma=suma+analogRead(A0);
  }
  float adc=suma/n;
  float distancia_cm = 17569.7 * pow(adc, -1.2062);
  return(distancia_cm);
}
```

## 1.2 Predict Model

- Pre-process your data (if required) and to perform an Exploratory Data Analysis.