

# UntitledAssignment 3, “Supervised learning final project”

Alejandro Pachón, Santiago Meza, Alexander Morgan

2023-05-29

## Sensores

### MQ2:

El funcionamiento del sensor MQ2 se basa principalmente en medir la variación en la conductividad de la alterada por la presencia de gases en el aire, estas variaciones producen señales eléctricas que se interpretan de diferentes maneras para llegar al valor de la concentración de gases en el ambiente. Para información más detallada consultar el datasheet.

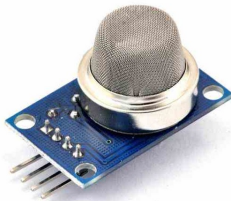


Figure 1: Título de la imagen

Para la recolección de datos usando el sensor MQ2, se simuló el ambiente de una cueva volcánica, la cual constaba con irregularidades en el suelo como si fueran grietas, en estas también se agrego alcohol atomizado, se humedecieron partes del ambiente con gasolina y se agrego un poco de gas de un encendedor para simular los gases tóxicos que dichas cuevas pueden llegar a expulsar.

### DHT11:

El sensor DHT11 es un sensor capacitivo que mide la humedad y temperatura relativa del aire, incluye también un termistor interno el cual mide la temperatura del ambiente, y muestra los datos mediante una señal digital en el pin de datos, Para información más detallada revisar el datasheet.

En cuanto a la recolección de datos, se tuvieron en cuenta dos de los ambientes simulados, la jungla y el desierto, ya que estos ambientes cambian abruptamente las magnitudes de temperatura y calor debido a la relación entre estas, ya que en el ambiente de jungla se buscaba encontrar una alta humedad y mantener una temperatura relativa estable para que le sensor midiera la relación normalizada en este ambiente, en el ambiente de desierto se busco lo contrario, y se evidencio en el dataset como al elevar la temperatura el factor de humedad disminuye considerable y constantemente.

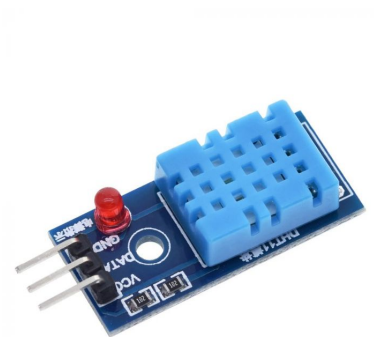


Figure 2: Título de la imagen

### Hc Sr04

El funcionamiento básico del sensor Hc Sr04 (sensor de ultrasonido) es de la emisión y recepción de pulsos ultrasónicos por sus transductores y medir el tiempo que este tarda en llegar al receptor del sensor. Para información más detallada puede consultar el datasheet.

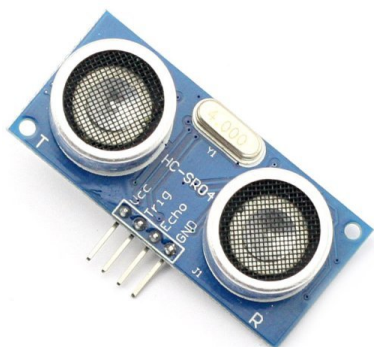


Figure 3: Título de la imagen

El sensor de ultrasonido resultó útil en los 3 ambientes, ya que cada uno de estos simulo una distancia de espacio diferente, en el caso de la cueva, al ser ambiente con menor espacio, comparando sus medidas con los demás ambientes son considerables, en el ambiente de jungla al tener diferentes obstáculos y al estar en diferentes alturas, se buscaba simular un espacio con árboles, hojas y/o arbustos, los cuales dificultaran la medición de un espacio exterior, y finalmente en el espacio de desierto se aplicó lo contrario a los dos anteriores escenarios, ya que als er un desierto el espacio de este es muy amplio y esto se refleja en las medidas del dataset.

### Modelos de Predicción

Para realizar la partición del dataset, se realizan las líneas de código:

las cuales dan a entender que el dataset tomará una partición de 70 / 30, el cual el 70% será dedicado a la función de entrenar nuestro algoritmo para que sea capaz de realizar predicciones, y el 30% será para realizar las pruebas de estas mismas.

Para asignar las funciones de entrenamiento y pruebas con los datos se realizan las líneas de las funciones

respectivas.

```
library(tidyverse)
library(psych)
library(caret)
library(ggplot2)
library(rpart)
library(randomForest)
library(class)
library(gmodels)
predictors <- colnames(DataSet)[-7]
data.samples <- sample(1:nrow(DataSet),
                      nrow(DataSet) * 0.7, replace = FALSE)
sample(DataSet)
training.data <- DataSet[data.samples, c(predictors, "AMBIENTE")]
test.data <- DataSet[-data.samples, c(predictors, "AMBIENTE")]
```

De esta manera ya logramos que en `training.data` quede nuestro nuevo dataset de entrenamiento y en `test.data` nuestros datos de prueba.

## Modelo Lineal

Para nuestro modelo de predicción lineal asignamos las variables y la variable de clase al tipo de modelo lineal que vamos a elaborar, en este caso asignamos nuestras variables de temperatura, humedad, distancia, concentración de gases, y la variable predecir será nuestra variables clase la cual será el ambiente en el que se debería encontrar la plataforma robótica, esto se realiza mediante el siguiente código:

```
#Modelo Lineal
modelo_multilineal <- lm(DataSet1$AMBIENTE ~ DataSet1$TEMP + DataSet1$HUMEDAD + DataSet1$DIS_UP +
                        DataSet1$PPM, data = training.data)
```

Al final de esto, se ve como esto se asigna a nuestro `training.data`. Una vez realizado el modelo lineal, se utiliza la función `summary(modelo_multilineal)` y después se puede realizar la predicción del modelo en base al `test.data`.

```
summary(modelo_multilineal)
prediccionlm <- predict(modelo_multilineal, test.data)
prediccionlm
RMSEmodelo1 = data.frame(prediccion = prediccionlm
                        , actual = DataSet1
                        , RSE = sqrt((prediccionlm - DataSet1$AMBIENTE)^2)
)
View(RMSEmodelo1)
```

## Árbol de decisión

Para el modelo de Árbol de decisión se requiere la función `randomForest`, y como en el caso anterior, se asigna el `training.data` teniendo en cuenta que la variable clase es `AMBIENTE`

```
# Arbol de decision
fit.rf <- randomForest(AMBIENTE ~ TEMP + HUMEDAD + DIS_UP +
                      PPM, data = training.data)
```

Después se realiza la predicción de `fit.rf` junto a nuestro `test.data`, y como salida tenemos un dataframe relacionando nuestra predicción y nuestros datos de prueba.

```

prediction.rf <- predict(fit.rf, test.data)
output <- data.frame(test.data$Mileage, prediction.rf)
RMSE = sqrt(sum((output$test.data.Mileage - output$prediction.rf)^2)/
              nrow(output))

```

Para finalizar, realizamos el modelo de predicción 2 llamado RMSEmodelo2

```

RMSE
RMSEmodelo2 = data.frame(prediccion = prediction.rf
                          ,ahora = test.data$Mileage
                          ,RSE = sqrt((prediction.rf-test.data$mileage)^2)
                          )
View(RMSEmodelo2)

```